IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 30, 2018, accepted January 4, 2019, date of publication January 9, 2019, date of current version January 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891575

Data-Driven Adaptive Optimal Control for Linear Systems With Structured Time-Varying Uncertainty

MENG ZHANG AND MING-GANG GAN, (Member, IEEE)
State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Ming-Gang Gan (agan@bit.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61673065.

ABSTRACT In this paper, a data-driven adaptive optimal control strategy is proposed for a class of linear systems with structured time-varying uncertainty, minimizing the upper bound of a pre-defined cost function while maintaining the closed-loop stability. An off-policy data-driven reinforcement learning algorithm is presented, which uses repeatedly the online state signal on some fixed time intervals without knowing system information, yielding a guaranteed cost control (GCC) law with quadratic stability for the system. This law is further optimized through a particle swarm optimization (PSO) method, the parameters of which are adaptively adjusted by a fuzzy logic mechanism, and an optimal GCC law with the minimum upper bound of the cost function is finally obtained. The effectiveness of this strategy is verified on the dynamic model of a two-degree-of-freedom helicopter, showing that both stability and convergence of the closed-loop system are guaranteed and that the cost is minimized with much less iteration than the conventional PSO method with constant parameters.

INDEX TERMS Adaptive optimal control, particle swarm optimization, fuzzy logic, structured uncertainty.

I. INTRODUCTION

The problem of maintaining the stability and performance of linear systems subject to time-invariant or time-varying parameter uncertainties, which is usually caused by internal or external perturbations, has been an active topic of research for quite some time. Particularly, many researchers have focused on the "structured" uncertainty, which allows one to have access to the bounds on the individual elements of the uncertainty and has extensive applications in many engineering disciplines [1]–[3]. Both the admissible uncertainty bounds for stability and stabilization methods for such systems have been studied by means of robust design in various literature, see [4]–[8] for example. Furthermore, many researchers have focused on their performance as well as stability [9]–[12]. In the presence of time-varying parameters, one usually aims at confining a predefined cost function within a boundary by proper controllers, namely "guaranteed cost control (GCC)" [1], [13], [14].

Nevertheless, in practice, it is often not easy to obtain the real-time values of parameters of a dynamic uncertain system. Even the constant nominal model might be unmeasurable. In this case, it is full of challenge to guarantee the performance and stability of the closed-loop system. To this end, we borrow the idea of reinforcement learning, which is called approximate/adaptive dynamic programming (ADP) in control literature [15]–[18], to adaptively derive the optimal control in a data-driven scheme. In other words, the *a priori* knowledge of the system model is no more required, since we are able to iteratively reach the optimal control law by measuring and collecting signals such as state and control input. Plenty results have been brought about on ADP methods for linear and nonlinear systems in the last decade (see [19]–[25]). Most of those work, however, is limited within time-invariant systems. When applying ADP methods on systems with unknown model and time-varying uncertainty, one will be confronted with considerable difficulties, since the stability and performance analysis is much more complicated than that under time-invariant situations [2], [26]. In fact, to the best of the authors' knowledge, few work has been done on the optimality of time-varying systems with unknown parameters.

In this paper, we propose to integrate ADP and particle swarm optimization (PSO) [27]–[30] methods to obtain the optimal control law for a class of systems with structured time-varying uncertainty, which guarantees the quadratically stability of the closed-loop system and minimizes a predefined cost function. Still, the integration is not trivial. First, most existing ADP results are on-policy methods, which means it would take quite a while for each single particle to reach the "fitness" during every iteration in the PSO algorithm, leading to exponential increase of converging time and deteriorated performance. Motivated by the work in [20] and [23], we present an off-policy ADP algorithm, using the state signal on some fixed time intervals repeatedly and implementing all the computation at one time. Furthermore, the selection of parameters for a PSO algorithm has considerable influence on the performance and convergence [31], [32]. In this paper, an adaptive scheme is designed to determine these parameters by means of fuzzy logic [33], [34].

To summarize, this paper proposes an adaptive data-driven strategy to minimize a pre-defined cost function and maintain the stability for linear systems with structured time-varying uncertainty. The main contributions are as follows. First, the optimal control problem for a class of unmodeled systems with time-varying parameters is solved. Second, this is an important step for the extension of RL and ADP methods to general time-varying systems. Third, the gap between ADP and modern intelligent methods such as PSO and fuzzy control is bridged.

This paper is organized follows. The problem and some preliminaries are stated in Section II. Then in Section III, a data-driven algorithm based on off-policy ADP is presented. This law is further optimized by a PSO scheme, and a fuzzy mechanism is developed to determine the PSO parameters. Consequently, an adaptive optimal strategy which yields the optimal GCC law is formulated. The effectiveness of this strategy is verified on the dynamic model of a 2-degree-of-freedom helicopter in Section IV. Finally, conclusion remarks are contained in Section V.

Notation: Throughout this paper, we use $\mathbb{R}$ to denote the set of real numbers. I_n represents the identity matrix of size n. $|\cdot|$ stands for the absolute value of a number, and $\|\cdot\|$ stands for the Euclidean norm of a vector or the induced norm of a matrix. For a symmetric matrix P, we use $P^\dagger$ to denote its pseudoinverse, $P > 0$ to denote its positive definiteness, $\mathrm{tr}(P)$ to denote its trace, and $P^{1/2}$ to denote the unique symmetric positive definite matrix X satisfying $X^2 = P$. The operator $\mathcal{E}(\cdot)$ indicates the expectation of a variable, and $\min(\cdot)$ returns the lowest value among a set. The expression $x \sim \mathcal{U}(a, b)$ is used to denote a variable x that is uniformly distributed on the interval $[a, b]$.

II. PROBLEM FORMULATION AND PRELIMINARIES

Consider the linear system with structured time-varying uncertainty described by

$$\dot{x}(t) = (A_0 + DF(t)E)x(t) + Bu(t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state; $u(t) \in \mathbb{R}^m$ is the control input; $A_0 \in \mathbb{R}^{n\times n}$ is the nominal system matrix which is unknown; the input matrix $B \in \mathbb{R}^{n\times m}$ is available and has full rank; $D \in \mathbb{R}^{n\times p}$ and $E \in \mathbb{R}^{q\times n}$ are known constant matrices, and $F(t) \in \mathbb{R}^{p\times q}$ is a matrix of uncertain time-varying parameters satisfying $F(t)^T F(t) \le I_q$.

Associated with this system is the infinite horizon quadratic cost function

$$\mathcal{J}(x(t_0), u(t)) = \int_{t_0}^{\infty} (x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau))d\tau \tag{2}$$

subject to the optimal control problem

$$u^* = \arg\min_{u(t)} \mathcal{J}(x(t_0), u(t)) \tag{3}$$

where $Q \in \mathbb{R}^{n\times n} > 0$ and $R \in \mathbb{R}^{m\times m} > 0$ are known matrices. For simplicity, the time variable t in $x(t)$, $u(t)$, and $F(t)$ is omitted in the rest of this paper unless necessary.

Here we give the definitions of quadratic stability and guaranteed cost control.

Definition 1 [1], [9], [11]: The closed-loop system (1) is said to be quadratically stable with $u = -Kx$, where $K \in \mathbb{R}^{m\times n}$ is a constant feedback gain, if there exists a matrix $P > 0$ and a positive scalar α, such that for every $x \in \mathbb{R}^n$, the time derivative for the Lyapunov function $V(x, t) = x^T Px$ satisfies

$$\dot{V}(x, t) = 2x^T(P(A_0 + DFE - BK))x \le \alpha\|x\|^2. \tag{4}$$

Remark 1: Note that if $P(A_0+DFE-BK)+(A_0+DFE-BK)^T P < 0$, we can always find a $\alpha > 0$ such that (4) holds for all $x \in \mathbb{R}^n$. Furthermore, by (4) one has $\dot{V}(x, t) < 0$ for all nonzero x. In this sense, the quadratic stability of (1) provides a sufficient condition for its asymptotic stability.

Definition 2 [1], [13], [35]: A control law u is said to define a guaranteed cost control (GCC) for (1) with respect to the cost function described by (2) if there exists a number $\mathcal{J}_0$ such that $\mathcal{J}(x(t_0), u(t)) \le \mathcal{J}_0$ for any finite $x(t_0)$. Specifically, the linear control law $u = -Kx$ is a quadratic GCC for (1) with respect to (2) if there exists a matrix $P > 0$ such that

$$x^T(Q + K^T RK)x + 2x^T P(A_0 + DFE - BK) \le 0. \tag{5}$$

for all $x \in \mathbb{R}^n$ and all matrices $F : F^T F \le I_q$. In this case, the value of the cost function is guaranteed to satisfy

$$\mathcal{J}(x(t_0), u(t)) \le x(t_0)^T Px(t_0). \tag{6}$$

It has been proved in [35] that if $u = -Kx$ is a quadratic GCC for (1), then the closed-loop system is quadratically stable, and vice versa. If we further restrict $x(t_0)$ such that it is zero mean random and satisfies $\mathcal{E}\left(x(t_0)x(t_0)^T\right) = I_n$, then the cost function (2) becomes

$$\mathcal{J}(u) = \mathcal{E}\left(\int_{t_0}^{\infty}\left(x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau)\right)d\tau\right). \tag{7}$$

Then the following preliminaries are recalled.

Lemma 1 [1], [35]: Suppose there exists a constant $\varepsilon > 0$ such that the Riccati equation

$$A_0^T P + PA_0 - PBR^{-1}B^T P + \varepsilon PDD^T P + \frac{1}{\varepsilon}E^T E + Q = 0 \tag{8}$$

for (1) has a solution $P > 0$. *Then the closed-loop system is quadratically stable with the control law being* $u = -R^{-1}B^TPx$. *Moreover, this solution* P*, which is associated with* ε*, serves as an upper bound of the cost function (7), i.e.,* $J(u) \le \mathrm{tr}(P)$.

Lemma 2 [14]: If the Riccati equation (8) has a solution $P(\tilde{\varepsilon}) > 0$ *associated with* $\varepsilon = \tilde{\varepsilon}$*, then it will have a positive definite solution for all* $(0, \tilde{\varepsilon})$*, and* $\mathrm{tr}(P(\varepsilon))$ *is a convex function of* ε *on* $(0, \tilde{\varepsilon})$.

According to Lemma 2, the optimal GCC with respect to (7) which yields the minimum cost upper bound can be obtained by choosing $\varepsilon > 0$ to minimize $\mathrm{tr}(P(\varepsilon))$, where P is the positive definite solution to (8) associated with ε. This work aims at finding this minimum upper bound and the corresponding optimal ε, i.e.,

$$\varepsilon^* = \arg\min_{\varepsilon} \mathrm{tr}(P(\varepsilon)) \tag{9}$$

in the presence of unknown system matrix A_0 and time-varying uncertainty $\Delta A =: DF(t)E$. To this end, some assumptions are required, which are reasonable for many practical systems.

Assumption 1: The system (1) is quadratically stabilizable, i.e., there always exists $\varepsilon > 0$ *such that (8) has a positive definite solution* $P(\varepsilon)$.

Assumption 2: The nominal system matrix A_0*, although unknown, can be separated from the uncertainty* ΔA*. Since the uncertainty is time varying, this assumption is essential for a data-driven algorithm to converge.*

III. MAIN RESULTS

In this section, an adaptive data-driven strategy is developed, which integrates the ideas from ADP, PSO and fuzzy theory to obtain the optimal ε^* and the corresponding cost matrix $P(\varepsilon)$. We start from the case where ε is constant, and then extend the results to that where it is variable.

A. AN OFF-POLICY DATA-DRIVEN ADP ALGORITHM

Suppose $\varepsilon = \varepsilon_0 > 0$ is a constant. Then (8) can be rewritten as

$$A_0^TP + PA_0 - PB\bar{R}^{-1}B^TP + \bar{Q} = 0 \tag{10}$$

where $\bar{R} =: (R^{-1} - \varepsilon_0(B^TB)^{\dagger}B^TDD^TB(B^TB)^{\dagger})^{-1}$, $\bar{Q} =: Q + \frac{1}{\varepsilon_0}E^TE$.

It is not hard to see that $\bar{Q} > 0$, and with Assumption 1 satisfied, for some $\varepsilon_0 > 0$, $\bar{R}$ is well defined and positive definite if R is properly chosen. In the following discussion, we suppose the values of ε_0 and R can guarantee $\bar{R} > 0$. As long as (A_0, B) is controllable, (10) has a unique solution $P_0^* > 0$ [36], and the control law $u = -K_0^*x$ where $K_0^* = \bar{R}^{-1}B^TP_0^*$ is the optimal control law with respect to the following transformed cost function

$$\bar{\mathcal{J}}(x(t_0), u) = \int_{t_0}^{\infty} \left(x^T(\tau)\bar{Q}x(\tau) + u^T(\tau)\bar{R}u(\tau) \right) d\tau \tag{11}$$

for the nominal system of (1) described by

$$\dot{x} = A_0x + Bu. \tag{12}$$

Assumption 2 allows one to design a data-driven scheme on the above system. To this end, we recall the following results.

Lemma 3 [20], [37]: Suppose $K_0 \in \mathbb{R}^{m\times n}$ *is a stabilizing feedback gain for the system (12), and* $P_k(k = 0, 1, \ldots) \in \mathbb{R}^{n\times n}$ *is the symmetric positive definite solution of Lyapunov equation*

$$P_k(A_0 - BK_k) + (A_0 - BK_k)^TP_k + K_k^T\bar{R}K_k + \bar{Q} = 0 \tag{13}$$

where $K_k(k = 1, \ldots)$ *is defined recursively by*

$$K_k = \bar{R}^{-1}B^TP_{k-1}, \tag{14}$$

then $A_0 - BK_k(k = 1, 2, \ldots)$ *is Hurwitz, and* P_k *and* K_k *will converge to* P_0^* *and* K_0^**, respectively, i.e.,* $\lim_{k\to\infty} P_k = P^*$ *and* $\lim_{k\to\infty} K_k = K^*$.

Accordingly, given a stabilizing $K_0 \in \mathbb{R}^{m\times n}$, along the solutions of (12) by (13) and (14) we have

$$x^TP_kx|_t^{t+\delta t} = -\int_t^{t+\delta t} x(\tau)^T(\bar{Q} + K_k^T\bar{R}K_k)x(\tau)d\tau \tag{15}$$

for $k = 0, 1, 2, \ldots$.

Define

$$\hat{P} =: \left[p_{11}, 2p_{12}, \ldots, 2p_{1n}, p_{22}, 2p_{23} \ldots, 2p_{n-1,n}, p_{nn}\right] \tag{16}$$

where p_{ij} is the (i, j)-th element of matrix P , and

$$\hat{x} =: [x_1^2, x_1x_2, \ldots, x_1x_n, x_2x_3, \ldots, x_{n-1}x_n, x_n^2]^T \tag{17}$$

where x_i is the i-th element of vector x. Then (15) could be transformed as

$$\hat{P}_k^T\hat{x}|_t^{t+\delta t} = -\hat{Q}_k^T \int_t^{t+\delta t} \hat{x}d\tau \tag{18}$$

for $k = 0, 1, 2, \ldots$, where $\hat{Q}_k = \mathrm{vec}(\bar{Q} + K_k^T\bar{R}K_k)$.

Define

$$D_{xx} =: \left[\hat{x}|_{t_0}^{t_1}, \hat{x}|_{t_1}^{t_2}, \ldots, \hat{x}|_{t_{r-1}}^{t_r}\right]^T \tag{19}$$

and

$$I_{xx} =: \left[\int_{t_0}^{t_1} \hat{x}(\tau)d\tau, \int_{t_1}^{t_2} \hat{x}(\tau)d\tau, \ldots, \int_{t_{r-1}}^{t_r} \hat{x}(\tau)d\tau\right]^T \tag{20}$$

where $t_1, t_2, \ldots, t_r$ are predefined constants satisfying $t_0 < t_1 < t_2 < \ldots < t_r$ and r is an available positive integer. Obviously, $\hat{x}, \hat{P}_k, \hat{Q}_k \in \mathbb{R}^{\frac{n(n+1)}{2}}$; $D_{xx}, I_{xx} \in \mathbb{R}^{r\times\frac{n(n+1)}{2}}$. Then one can solve $\hat{P}_k(k = 0, 1, 2, \ldots)$ by

$$\hat{P}_k = -(D_{xx}^TD_{xx})^{-1}D_{xx}^TI_{xx}\hat{Q}_k. \tag{21}$$

Thus with the persistent excitation (PE) condition satisfied, the values of $P_k(k = 0, 1, 2, \ldots)$ and $K_k(k = 1, 2, \ldots)$ can be iteratively obtained via (21) and (14), respectively. To ensure the PE condition, the initial stabilizing controller is chosen to be $u_0 = -K_0x + e$, where e is an exploration noise. Practically, the iteration would stop when the condition $\|P_k - P_{k-1}\| \le \kappa$ is satisfied, where $\kappa > 0$ is a predefined threshold that is sufficiently small. For the uniqueness of $\hat{P}_k$, r should be carefully selected such that $\text{rank}(D_{xx}) \geq \dfrac{n(n+1)}{2}$. On the basis of experience, it is a safe choice of fixing $r \geq n(n+1)$.

The above process is concluded as Algorithm 1, and we draw Theorem 1.

Algorithm 1 Off-Policy Data-Driven ADP Algorithm

```
Input: Data of $D_{xx}$ and $I_{xx}$ collected from the nominal system described by (12) on the time interval $[t_0, t_r]$
  repeat
    Compute $(P_k, K_{k+1})$ using $D_{xx}$ and $I_{xx}$ by (21) and (14)
  until $\|P_k - P_{k-1}\| \leq \kappa$
Output: The cost matrix $P(\varepsilon_0) = P_0^* = P_k$
```

Theorem 1: Starting from a stabilizing K_0, with rank $(D_{xx}) \geq \dfrac{n(n+1)}{2}$ *satisfied, the sequences $\{P_k\}_{k=0}^{\infty}$ and $\{K_k\}_{k=0}^{\infty}$ obtained by Algorithm 1 converge to P_0^* and K_0^*, respectively. Furthermore, $u = -K_0^* x$ is a quadratically stabilizing and quadratic GCC law for (1).*

Proof: Since P_k is symmetric, it can be uniquely decided by $\hat{P}_k$, and $\hat{P}_k$ is uniquely decided by (21) if $\text{rank}(D_{xx}) \geq \dfrac{n(n+1)}{2}$. Thus solving P_k by (21) is equivalent to solving it by (13). According to Lemma 3, with a stabilizing K_0, $\{P_k\}_{k=0}^{\infty}$ and $\{K_k\}_{k=0}^{\infty}$ obtained iteratively by (21) and (14) converge to P_0^* and K_0^*, respectively. Further, since P_0^* is the solution to (8), by Lemma 1 we know $u = -\bar{R}^{-1}B^T P_0^* x = -K_0^* x$ is a quadratically stabilizing and quadratic GCC law for (1). The proof is completed. □

Remark 2: The main difference between the Algorithm 1 and the approach introduced by Vrabie et al. [19] is that Algorithm 1 is an off-policy method, while the latter is an on-policy algorithm. For a given ε_0, we can use the information of state x on a fixed time interval $[t_0, t_r]$ in every iteration to approximate P_0^* and K_0^*.

If we replace $\bar{Q}$ and $\bar{R}$ with Q and R, respectively, then (10) is reduced to the nominal Riccati equation

$$A_0^T P + PA_0 - PBR^{-1}B^T P + Q = 0. \tag{22}$$

Given an initial stabilizing law, the unique solution P_{nom}^* to (22) and the corresponding feedback gain $K_{nom}^* = -R^{-1}B^T P_{nom}^*$ could be obtained iteratively in the same way as Algorithm 1, and K_{nom}^* is equal to the feedback gain obtained by the on-policy ADP in [19]. Interestingly, we now show that K_{nom}^* is quadratically stabilizing for (1) if certain condition is satisfied.

Corollary 1: The closed-loop system (1) is quadratically stable with $u = -K_{nom}^* x$ where $K_{nom}^* = R^{-1}B^T P_{nom}^*$ if there exists a symmetric positive definite matrix $\Lambda \in \mathbb{R}^{n \times n}$ such that

$$-P_{nom}^* BR^{-1}B^T P_{nom}^* + P_{nom}^* DD^T P_{nom}^* + E^T E + \Lambda - Q = 0 \tag{23}$$

where P_{nom}^* is the unique solution to (22).

Proof: By (23) it is obvious that

$$-P_{nom}^* BR^{-1}B^T P_{nom}^* + P_{nom}^* DD^T P_{nom}^* + E^T E - Q < 0. \tag{24}$$

Since P_{nom}^* satisfies (22), it follows that

$$A_0^T P_{nom}^* + P_{nom}^* A_0 - 2P_{nom}^* BR^{-1}B^T P_{nom}^* + P_{nom}^* DD^T P_{nom}^* + E^T E < 0. \tag{25}$$

From [8] we know that for any matrices X and Y with appropriate dimensions, the following inequality

$$X^T Y + Y^T X \leq \beta X^T X + \frac{1}{\beta} Y^T Y \tag{26}$$

always holds for any $\beta \in \mathbb{R} > 0$. Considering $F^T F < I$, we have

$$P_{nom}^* DD^T P_{nom}^* + E^T E > P_{nom}^* DD^T P_{nom}^* + E^T F^T FE \geq P_{nom}^* DFE + E^T F^T D^T P_{nom}^*. \tag{27}$$

Thus,

$$A_0^T P_{nom}^* + P_{nom}^* A_0 - 2P_{nom}^* BR^{-1}B^T P_{nom}^* + P_{nom}^* DFE + E^T F^T D^T P_{nom}^* < 0. \tag{28}$$

Therefore,

$$2x^T (P_{nom}^* (A_0 + DFE - BK_{nom}^*))x < 0 \tag{29}$$

holds for any $x \in \mathbb{R}, x \neq 0$. Consequently, there exists a $\alpha > 0$ such that (4) is satisfied for $P = P_{nom}^*$, and the closed-loop system (1) is quadratically stable with $u = -K_{nom}^* x$. The proof is completed. □

Note that even though K_0^* or K_{nom}^* provides GCC for (1), the upper bound of the cost (2) or (7) cannot be guaranteed to be minimum, since the bound is associated with ε, while the derivation of K_0^* or K_{nom}^* is either with a constant ε_0 or without ε. Next, the data-driven algorithm proposed above will be extended to the case where ε is variable.

B. COMPUTATION OF OPTIMAL GCC VIA PSO WITH ADAPTIVE PARAMETERS

According to the above data-driven scheme, given $\varepsilon > 0$, we could iteratively compute the corresponding solution $P(\varepsilon)$ to (8), if it exists. From Lemma 2 it is clear that a local minimum point of $\text{tr}(P(\varepsilon))$ with respect to ε is also the global minimum point. This enables us to design a PSO algorithm to converge to ε^* which yields the minimum $\text{tr}(P)$. In this algorithm, the variable ε serves as the candidate solution, and the fitness function is designed as the reciprocal of $\text{tr}(P)$, i.e.,

$$f(\varepsilon) = \frac{1}{\text{tr}(P(\varepsilon))}. \tag{30}$$

Let S be the swarm size (number of particles in the swarm), each having a position $\varepsilon^i > 0, i = 1, 2, ..., S$ in the search space $[\varepsilon_l, \varepsilon_u]$ where $\varepsilon_u \gg \varepsilon_l > 0$ with ε_l sufficiently small, and a velocity $v^i \in \mathbb{R}$. Let p^i be the best known position of particle i, i.e., with the maximum fitness value $f(\varepsilon^i)$, and g_b be the best known position of the entire swarm. The termination criterion for the algorithm is related to the total number of iterations performed, and the number of consecutive iterations where the global best position g_b remains the same.

The parameters of the PSO, including the inertia factor w, the cognitive factor ϕ_p, the social factor ϕ_g, and the upper threshold of velocity magnitude v_m, have considerable influence on the behavior and efficacy of the algorithm. In this work, we tune these parameters by means of fuzzy logic, and each particle is associated with its own values for these parameters.

Since $\varepsilon_u \gg \varepsilon_l > 0$, the size of search space could be set as ε_u instead of $\varepsilon_u - \varepsilon_l$, without noticeable effect on the performance. Then the normalized distance between two particles $i, j \in 1, 2, \ldots, S$ is defined as

$$d^{ij}(k) =: \frac{|\varepsilon^i(k) - \varepsilon^j(k)|}{\varepsilon_u} \tag{31}$$

where $k = 1, 2, \ldots$ denotes the number of iterations performed so far. Similarly, the normalized distance of the same particle i, which considers the positions of the particle at the current and previous iterations, is defined as

$$d^i(k) =: \frac{|\varepsilon^i(k) - \varepsilon^i(k-1)|}{\varepsilon_u}. \tag{32}$$

Accordingly, the normalized fitness increment for particle i is defined as

$$\Gamma^i(k) =: \frac{f(\varepsilon^i(k)) - f(\varepsilon^i(k-1))}{f(g_b)}. \tag{33}$$

TABLE 1. Fuzzy rules for PSO parameters.

IF	THEN
d is *Near* or Γ is *Worse*	w is *Low*
d is *Medium* or Γ is *Same*	w is *Medium*
d is *Far* or Γ is *Better*	w is *High*
d is *Far* or Γ is *Worse*	ϕ_p is *Low*
d is *Medium* or Γ is *Same*	ϕ_p is *Medium*
d is *Near* or Γ is *Better*	ϕ_p is *High*
d is *Near* or Γ is *Better*	ϕ_g is *Low*
d is *Medium* or Γ is *Same*	ϕ_g is *Medium*
d is *Far* or Γ is *Worse*	ϕ_g is *High*
d is *Near*	v_m is *Low*
d is *Medium* or Γ is *Same* or Γ is *Better*	v_m is *Medium*
d is *Far* or Γ is *Worse*	v_m is *High*

On these bases, a rule system including 12 fuzzy rules is designed, as illustrated in Tab. 1. The rules are based on two linguistic variables: a number d, which is to characterize the fuzzy distance between the particle position and the global best position; and a number Γ, which is a measurement of the fitness improvement for each particle with respect to the previous iteration. The term set of d is composed by three linguistic values: *Near*, *Medium*, and *Far*. The term set of Γ is composed by three linguistic values: *Better*, *Same* and *Worse*. The term set for each of the output variables (w, ϕ_p, ϕ_g, and v_m) is also composed by three linguistic values: *Low*, *Medium*, and *High*. Note that during every iteration, each particle computes its own values of d and Γ independently.

The base variable of d corresponds to the interval [0, 1]. The membership function of d, which is shown in Fig. 1, is described as follows:

$$\mu_N(d) = \begin{cases} 0.5(\cos(2.5\pi d) + 1), & 0 \le d \le 0.4 \\ 0, & 0.4 < d \le 1; \end{cases} \tag{34}$$

$$\mu_M(d) = \begin{cases} 0.5(\sin(2.5\pi(d - 0.2)) + 1), & 0 \le d \le 0.4 \\ 0.5(\cos(2.5\pi(d - 0.4)) + 1), & 0.4 < d \le 0.8 \\ 0, & 0.8 < d \le 1; \end{cases} \tag{35}$$

$$\mu_F(d) = \begin{cases} 0, & 0 \le d \le 0.4 \\ 0.5(\sin(2.5\pi(d - 0.6)) + 1), & 0.4 < d \le 0.8 \\ 1, & 0.8 < d \le 1. \end{cases} \tag{36}$$

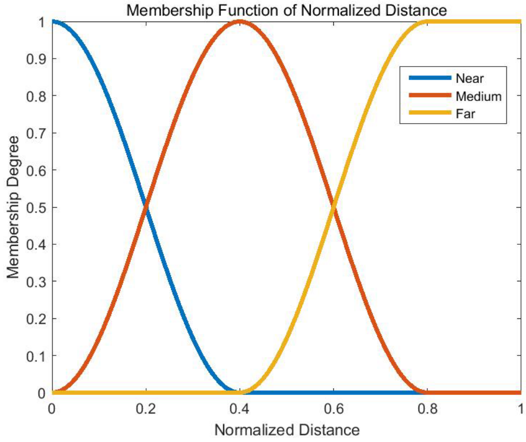

FIGURE 1. Membership function of normalized distance d.

The base variable of Γ corresponds to the interval $[-1, 1]$. The membership function of Γ, which is shown in Fig. 2, is described as follows:

$$\mu_W(\Gamma) = \begin{cases} 1, & -1 \le \Gamma \le -0.5 \\ 0.5(\cos(2\pi(\Gamma + 0.5)) + 1), & -0.5 < \Gamma \le 0 \\ 0, & 0 < \Gamma \le 1; \end{cases} \tag{37}$$

$$\mu_S(\Gamma) = \begin{cases} 0, & -1 \le \Gamma \le -0.5 \\ 0.5(\sin(2\pi(\Gamma + 0.25)) + 1), & -0.5 < \Gamma \le 0 \\ 0.5(\cos(2\pi(\Gamma + 0.25)) + 1), & 0 < \Gamma \le 0.5 \\ 0, & 0.5 < \Gamma \le 1; \end{cases} \tag{38}$$

$$\mu_B(\Gamma) = \begin{cases} 0, & -1 \le \Gamma \le 0 \\ 0.5(\sin(2\pi(\Gamma - 0.25)) + 1), & 0 < \Gamma \le 0.5 \\ 1, & 0.5 < \Gamma \le 1. \end{cases} \tag{39}$$

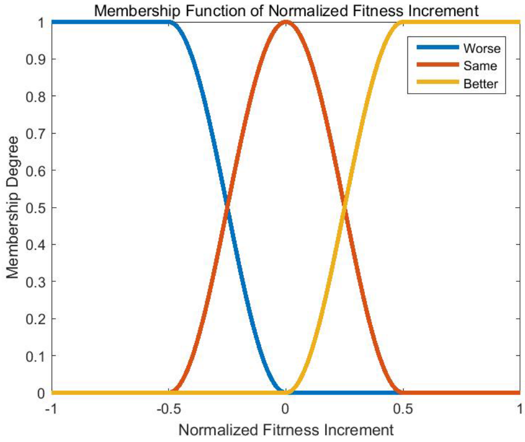

FIGURE 2. Membership function of normalized fitness increment Γ.

The linguistic values of output variables are modeled in Tab. 2.

TABLE 2. Crisp values of output variables.

Output variable	*Low*	*Medium*	*High*
w	0.5	0.8	1
ϕ_p	0.5	1	2
ϕ_g	1	2	3
v_m	$0.05\varepsilon_u$	$0.1\varepsilon_u$	$0.2\varepsilon_u$

For defuzzification, the center-of-gravity method [38] is applied, calculating the final numerical value of each output variable as the weighted average output of rules that has this output variable as their consequent. The formula is described as

$$O_s = \frac{\sum_{t=1}^{l_s} \rho_{st}\varphi_{st}}{\sum_{t=1}^{l_s} \rho_{st}} \tag{40}$$

where l_s is the number of rules that has the s-th output variable ($s = 1, 2, 3, 4$ correspond to w, ϕ_p, ϕ_g, v_m, respectively) as consequent (in this case $l_s = 3$ for $s = 1, 2, 3, 4$), ρ_{st} denotes the maximum membership degree for the input variables in the t-th rule, and φ_{st} denotes the crisp value of the s-th output variable for the t-th rule, as given in Tab. 2.

Take $s = 1$ as an example. Suppose we have $d = 0.2$, $\Gamma = 0.5$ for article i. According to Tab. 2, we know $\varphi_{11} = 0.5$, $\varphi_{12} = 0.8$, $\varphi_{13} = 1$, and

$$\begin{aligned}
\rho_{11} &= \max\{\mu_{Near}(0.2), \mu_{Worse}(0.5)\} \\
&= \max\{0.5, 0\} = 0.5, \\
\rho_{12} &= \max\{\mu_{Medium}(0.2), \mu_{Same}(0.5)\} \\
&= \max\{0.5, 0\} = 0.5, \\
\rho_{13} &= \max\{\mu_{Far}(0.2), \mu_{Better}(0.5)\} \\
&= \max\{0, 1\} = 1.
\end{aligned}$$

Then the output value of inertia factor for article i can be obtained as $w^i = O_1 = (0.5 \times 0.5 + 0.8 \times 0.5 + 1 \times 1)/(0.5 + 0.5 + 1) = 0.825$.

The above proposed PSO framework with adaptive fuzzy-tuned parameters is reported as Algorithm 2. During every iteration, it makes use of Algorithm 1 to compute the fitness for each particle, and update the parameters through the above fuzzy logic.

Algorithm 2 PSO Algorithm With Adaptive Parameters

```
Initialzation:
  $g_b \sim \mathcal{U}(\varepsilon_l, \varepsilon_u)$
  for each particle $i = 1, 2, ..., S$ do
    $w^i \leftarrow 0.8, \phi_p^i \leftarrow 1, \phi_g^i \leftarrow 2, v_m^i \leftarrow 0.1\varepsilon_u; \varepsilon^i \leftarrow \frac{i \times \varepsilon_u}{N}$,
    $v^i \sim \mathcal{U}(-v_m, v_m); p^i \leftarrow \varepsilon^i$
    if $f(p^i) > f(g_b)$ then
      $g_b \leftarrow p_i$
    end if
  end for
Optimization:
  repeat
    for each particle $i = 1, 2, ..., S$ do
      Pick $c_p^i, c_g^i \sim \mathcal{U}(0, 1)$
      $v^i \leftarrow w^i v^i + \phi_p^i c_p^i (p^i - \varepsilon^i) + \phi_g^i c_g^i (g_b - \varepsilon^i)$
      if $v^i > v_m^i$ or $v^i < -v_m^i$ then
        $v^i \leftarrow v_m^i$ or $v^i \leftarrow -v_m^i$
      end if
      $\varepsilon^i \leftarrow \varepsilon^i + v^i$
      if $\varepsilon^i > \varepsilon_u$ or $\varepsilon^i \leq 0$ then
        $\varepsilon^i \leftarrow \varepsilon_u$ or $\varepsilon^i \leftarrow \varepsilon_l$
      end if
      Solve $f(\varepsilon^i)$ via Algorithm 1 using $D_{xx}$ and $I_{xx}$
      if $f(\varepsilon^i) > f(p^i)$ then
        $p^i \leftarrow \varepsilon^i$
        if $f(p^i) > f(g_b)$ then
          $g_b \leftarrow p^i$
        end if
      end if
      Update $(w^i, \phi_p^i, \phi_g^i, v_m^i)$ via fuzzy logic
    end for
  until the termination criterion is met
```

Finally, the flowchart of the data-driven adaptive optimal control strategy for linear systems with time-varying uncertainty is illustrated in Fig. 3.

IV. EXAMPLE

The effectiveness of the above proposed strategy is validated via simulations on the dynamic model of a 2-degree-of-freedom helicopter [39], which is illustrated in Fig. 4.

The state is defined to be $x = [\theta, \psi, \dot{\theta}, \dot{\psi}]^T$, with the initial value $x_0 = [0.5, 0.5, -0.5, -0.5]^T$, and the control input is $u = [F_p, F_Y]^T$. The matrices of the nominal system are

$$A_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\dfrac{B_p}{J_p + ml_{cm}^2} & 0 \\ 0 & 0 & 0 & \dfrac{B_y}{J_y + ml_{cm}^2} \end{bmatrix},$$

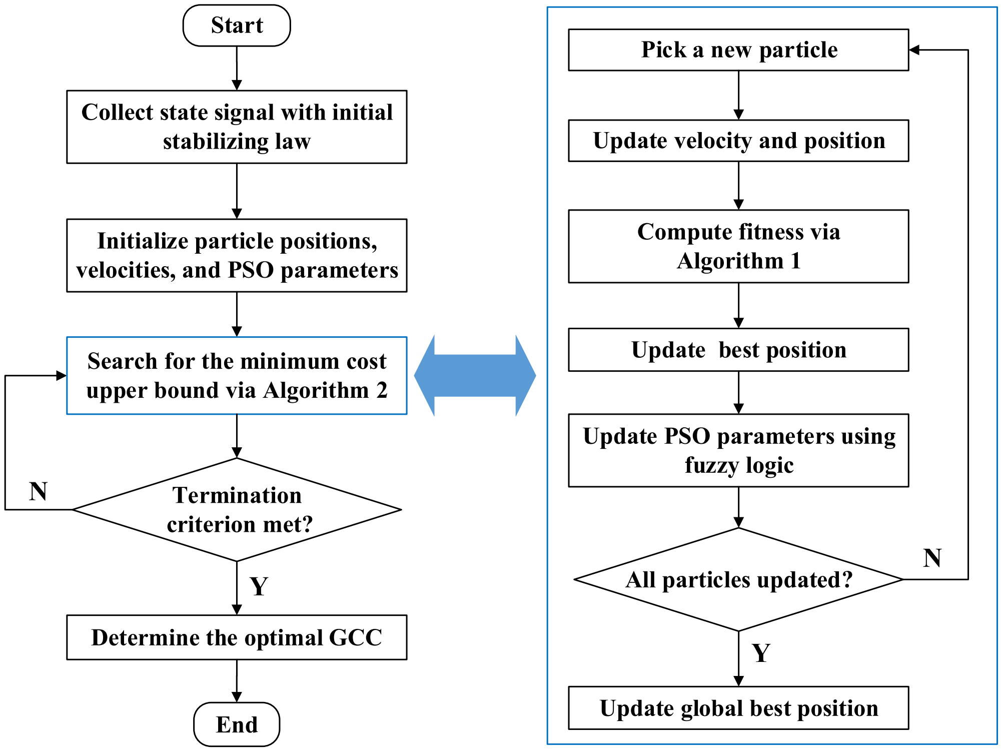
Start
Collect state signal with initial stabilizing law
Initialize particle positions, velocities, and PSO parameters
Search for the minimum cost upper bound via Algorithm 2
Termination criterion met?
N
Y
Determine the optimal GCC
End
Pick a new particle
Update velocity and position
Compute fitness via Algorithm 1
Update best position
Update PSO parameters using fuzzy logic
All particles updated?
N
Y
Update global best position

FIGURE 3. Flowchart of the data-driven adaptive optimal control strategy.

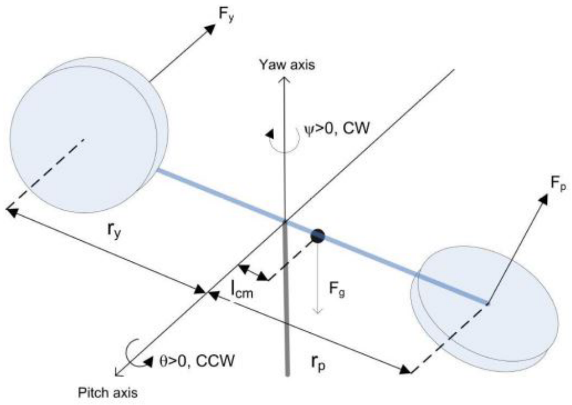

FIGURE 4. Free-body diagram of 2-DoF helicopter.

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \dfrac{K_{pp}}{J_p + ml_{cm}^2} & \dfrac{K_{py}}{J_p + ml_{cm}^2} \\ \dfrac{K_{yp}}{J_y + ml_{cm}^2} & \dfrac{K_y y}{J_y + ml_{cm}^2} \end{bmatrix}. \tag{41}$$

The meanings and values of the physical parameters involved are shown in Tab. 3. Accordingly, we obtain

$$A_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -9.259 & 0 \\ 0 & 0 & 0 & 3.487 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 2.361 & 0.0787 \\ 0.24 & 0.789 \end{bmatrix}.$$

TABLE 3. Meanings and values of physical parameters.

Parameter	Meaning	Value
m	total mass of the helicopter	1.3872 kg
l_{cm}	distance from axes plane to center of gravity	0.186 m
J_p	total moment of inertia about pitch axis	$0.0384\ \mathrm{kg}\cdot\mathrm{m}^2$
J_y	total moment of inertia about yaw axis	$0.0432\ \mathrm{kg}\cdot\mathrm{m}^2$
B_p	equivalent viscous damping about pitch axis	0.8 N/V
B_y	equivalent viscous damping about yaw axis	0.318 N/V
K_{pp}	thrust force constant of yaw propeller	$0.204\ \mathrm{N}\cdot\mathrm{m/V}$
K_{yy}	thrust torque constants acting on yaw axis from yaw controller	$0.072\ \mathrm{N}\cdot\mathrm{m/V}$
K_{py}	thrust torque constants acting on pitch axis from yaw controller	$0.0068\ \mathrm{N}\cdot\mathrm{m/V}$
K_{yp}	thrust torque constants acting on yaw axis from pitch controller	$0.0219\ \mathrm{N}\cdot\mathrm{m/V}$

The uncertainty matrices are $D = \mathrm{diag}(0, 0, 0.1, 0.2)$, $F(t) = \mathrm{diag}(\sin(t), \cos(t), 1, 1)$, and $E = I_4$. The weight matrices are set as $Q = I_4$ and $R = I_2$.

The data collection process of the ADP algorithm is from $t_0 = 0$s to $t_r = 1$s, where $r = 100$, and the state signal is collected every 0.01 seconds. The initial stabilizing feedback gain is chosen to be $K_0 = [2.139, -0.2134, 0, 0; -6.508, 6.402, 0, 0]$, and the exploration noise is $e = [\sum_{i=1}^{5} \sin(\omega_i t), \sum_{j=1}^{5} \sin(\omega_j t)]^T$ where $\omega_i, \omega_j \sim \mathcal{U}(-100, 100)$.

The learning process starts at $t = 1$s, and the control input during this process is $u = -K_0 x$. The threshold of

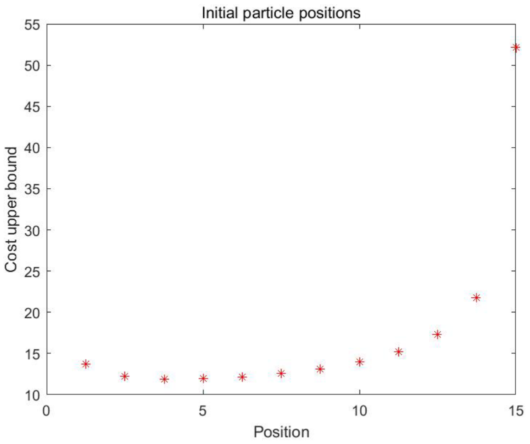

FIGURE 5. **Initial particle positions of PSO with adaptive parameters.**

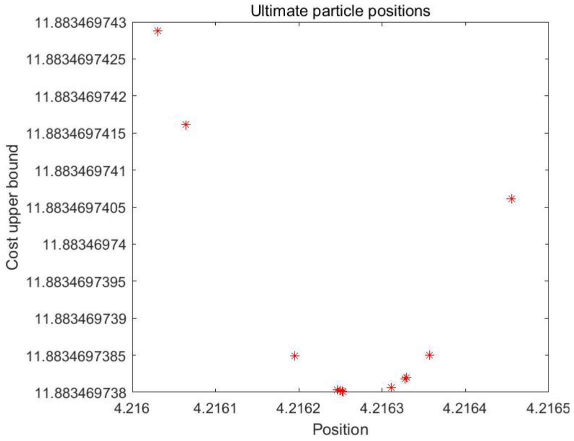

FIGURE 6. **Ultimate particle positions of PSO with adaptive parameters.**

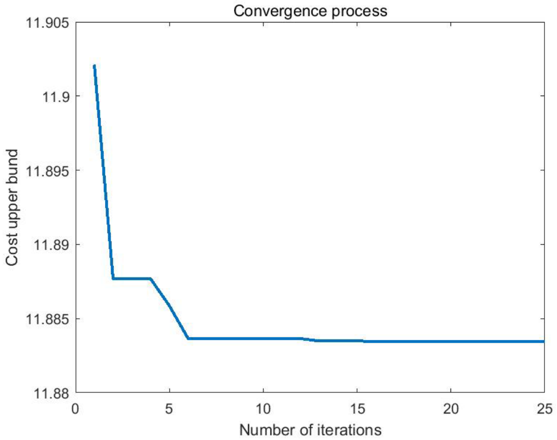

FIGURE 7. **Convergence process of PSO with adaptive parameters.**

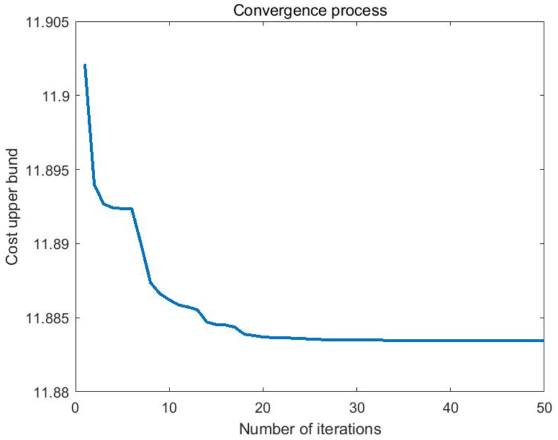

FIGURE 8. **Convergence process of PSO with constant parameters.**

stopping criterion for policy iteration is fixed as $\kappa = 10^{-8}$. The dimension of the PSO search space $M = 1$, so the swarm size is set to be $S = 10 + 2\sqrt{M} = 12$ [40]. The maximum number of iterations is set as $N = 100$. The PSO iteration would also stop if the global best position g_b remains the same for 10 consecutive iterations. Based on experience and experiment results, the bounds for search space are fixed to be $\varepsilon_l = 0.1$, $\varepsilon_u = 15$. The PSO parameters are initialized as in Algorithm 2. The particle positions at start and end of the learning process are shown in Figs. 5 and 6, respectively, together with the corresponding value of cost upper bounds. Note that the 12 particles are evenly distributed on the search space at start (see Fig. 5), and the values of their corresponding cost upper bounds are quite different. When the learning is complete, as can be read from Fig. 6, all particles converge to the optimal position, which is 4.216, and the minimized cost upper bound is 11.88.

Fig. 7 demonstrates The convergence process of the global best solution obtained by the proposed PSO with adaptive parameters. For the sake of comparison, we use the conventional PSO algorithm with constant parameters $w = 0.8$, $\phi_p = 1$, $\phi_g = 2$, $v_m = 1.5$ for all particles to deal with the same problem, and the convergence process is shown in Fig. 8. It is indicated that the PSO with adaptive fuzzy-tuned parameters can converge with about half less number of iterations than the conventional PSO method with constant parameters.

When the learning is over, the control input is switched to the optimal GCC law $u^* = -K^*x$, where $K^* = \bar{R}(\varepsilon^*)^{-1}B^TP(\varepsilon^*)$, with ε^* being the global best position obtained by the proposed strategy and $P(\varepsilon^*)$ being the corresponding minimum upper bound of cost. The state and control input trajectories of the closed-loop system controlled by the proposed strategy are shown in Figs. 9 and 10, respectively. By contrast, the state and input trajectories with the conventional ADP method which did not consider the time-varying uncertainty are also shown in Figs. 11 and 12, respectively.

As we can see, in Figs. 9 and 10 the data collection and learning is completed at about $t = 5$s, and then the optimal GCC law is employed, which is able to accelerate the convergence of state and input trajectories. In Figs. 11 and 12, though it takes less time to implement the learning process, all of the state and input trajectories fail to converge. In fact,

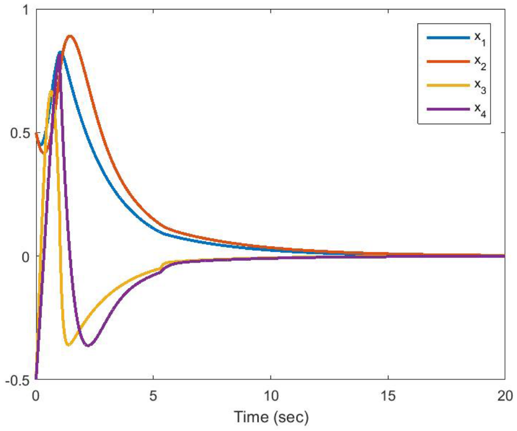

FIGURE 9. State trajectories with proposed adaptive optimal control.

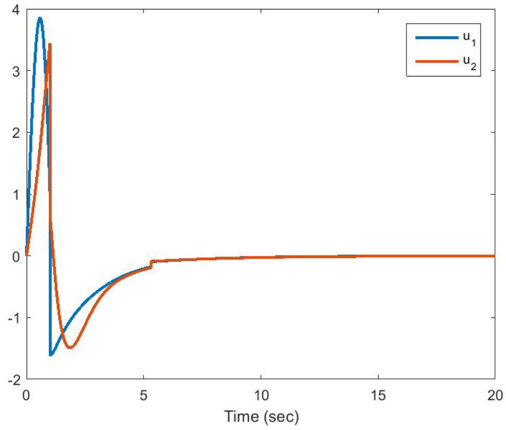

FIGURE 10. Input trajectories with proposed adaptive optimal control.

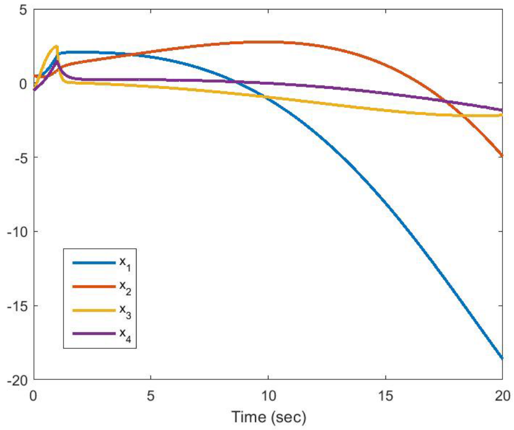

FIGURE 11. State trajectories with the conventional ADP method.

from Theorem 1 and Corollary 1 it is clear that the control law obtained by the proposed adaptive optimal control method is able to guarantee the quadratic stability as well as minimized upper bound of cost for the closed-loop system (1) with time-varying uncertainty, while the conventional ADP method cannot guarantee the stability and convergence.

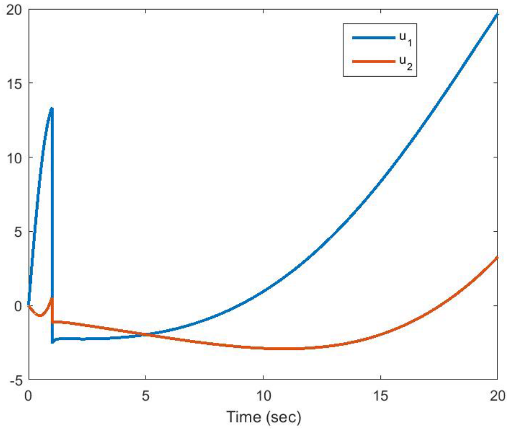

FIGURE 12. Input trajectories with the conventional ADP method.

TABLE 4. Comparisons of performance among different strategies.

Control strategy	State & input trajectories	PSO iteration times
Conventional ADP	Diverge	N/A
PSO-ADP I	Converge	33
PSO-ADP II	Converge	15

The above simulation results are summarized in Tab. 4, which illustrates comparisons of performance among the conventional ADP, the PSO-ADP with constant parameters (PSO-ADP I), and the PSO-ADP with adaptive parameters (PSO-ADP II), showing the superiority of PSO-ADP II, i.e., the proposed control strategy, in terms of stabilization and convergence properties.

V. CONCLUSION

Most previously obtained model-free optimal control results are limited within time-invariant systems. In this paper, a data-driven adaptive optimal strategy for linear systems with structured time-varying uncertainty is proposed, which bridges the gap between ADP methods and modern intelligent control methods such as PSO and fuzzy logic, guaranteeing the quadratic stability of the closed-loop system and minimizing the upper bound of the predefined cost function. The results have been validated via an example of 2-DoF helicopter, showing the superiority of the proposed strategy over the conventional ADP and PSO methods. This, we believe, is an important step towards extending RL and ADP methods to more general time-varying nonlinear systems.

REFERENCES

[1] R. K. Yedavalli, *Robust Control of Uncertain Dynamic Systems*. New York, NY, USA: Springer, 2016.

[2] F. Amato, *Robust Control of Linear Systems Subject to Uncertain Time-Varying Parameters*. Berlin, Germany: Springer, 2006.

[3] L. Wang, X. Gao, S. Cai, and X. Xiong, "Robust finite-time H_∞ filtering for uncertain discrete-time nonhomogeneous Markovian jump systems," *IEEE Access*, vol. 6, pp. 52561–52569, 2018.

[4] K. Zhou and P. P. Khargonekar, "Stability robustness bounds for linear state-space models with structured uncertainty," *IEEE Trans. Autom. Control*, vol. AC-32, no. 7, pp. 621–623, Jul. 1987.

[5] L. Xie and C. E. de Souza, "Robust H_∞ control for linear systems with norm-bounded time-varying uncertainty," *IEEE Trans. Autom. Control*, vol. 37, no. 8, pp. 1188–1191, Aug. 1992.

[6] R. K. Yedavalli, "Perturbation bounds for robust stability in linear state space models," *Int. J. Control*, vol. 42, no. 6, pp. 1507–1517, Jun. 1985.

[7] K. Zhou and P. P. Khargonekar, "Robust stabilization of linear systems with norm-bounded time-varying uncertainty," *Syst. Control Lett.*, vol. 10, no. 7, pp. 17–20, Jan. 1988.

[8] P. P. Khargonekar, I. R. Petersen, and K. Zhou, "Robust stabilization of uncertain linear systems: Quadratic stabilizability and H_∞ control theory," *IEEE Trans. Autom. Control*, vol. 35, no. 3, pp. 356–361, Mar. 1990.

[9] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. New York, NY, USA: Courier Corporation, 2007.

[10] D. Mehdi, M. Al Hamid, and F. Perrin, "Robustness and optimality of linear quadratic controller for uncertain systems," *Automatica*, vol. 32, no. 7, pp. 1081–1083, Jul. 1996.

[11] D. S. Bernstein and M. M. Haddad, "Robust stability and performance analysis for linear dynamic systems," *IEEE Trans. Autom. Control*, vol. 34, no. 7, pp. 751–758, Jul. 1989.

[12] M. Khammash and J. B. Pearson, "Analysis and design for robust performance with structured uncertainty," *Syst. Control Lett.*, vol. 20, no. 3, pp. 179–187, Mar. 1993.

[13] S. S. L. Chang and T. Peng, "Adaptive guaranteed cost control of systems with uncertain parameters," *IEEE Trans. Autom. Control*, vol. AC-17, no. 4, pp. 474–483, Aug. 1972.

[14] I. R. Petersen and D. C. McFarlane, "Optimizing the guaranteed cost in the control of uncertain linear systems," in *Robustness of Dynamic Systems with Parameter Uncertainties*. Basel, Switzerland: Birkhäuser, 1992, pp. 241–250.

[15] X. Luo, Y. Lv, R. Li, and Y. Chen, "Web service QoS prediction based on adaptive dynamic programming using fuzzy neural networks for cloud services," *IEEE Access*, vol. 3, pp. 2260–2269, 2015.

[16] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.

[17] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, 3rd Quart., 2009.

[18] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[19] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, Feb. 2009.

[20] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[21] T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems," *Automatica*, vol. 50, no. 10, pp. 2624–2632, Oct. 2014.

[22] T. Bian and Z.-P. Jiang, "Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design," *Automatica*, vol. 71, pp. 348–360, Sep. 2016.

[23] Y. Jiang and Z.-P. Jiang, *Robust Adaptive Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2017.

[24] X. Yang, D. Liu, and Q. L. Wei, "Online approximate optimal control for affine non-linear systems with unknown internal dynamics using adaptive dynamic programming," *IET Control Theory Appl.*, vol. 8, no. 16, pp. 1676–1688, Nov. 2014.

[25] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Inf. Sci.*, vol. 235, pp. 3–35, Jun. 2013.

[26] B. Zhou, "On asymptotic stability of linear time-varying systems," *Automatica*, vol. 68, pp. 266–276, Jun. 2016.

[27] L. Tong, X. Li, J. Hu, and L. Ren, "A PSO optimization scale-transformation stochastic-resonance algorithm with stability mutation operator," *IEEE Access*, vol. 6, pp. 1167–1176, 2018.

[28] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832–847, Dec. 2011.

[29] E. V. Kumar, G. S. Raaja, and J. Jerome, "Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter," *Appl. Soft Comput.*, vol. 41, pp. 77–90, Apr. 2016.

[30] M. Meissner, M. Schmuker, and G. Schneider, "Optimized particle swarm optimization (OPSO) and its application to artificial neural network training," *Bioinformatics*, vol. 7, no. 1, p. 125, 2006.

[31] K. Mason, J. Duggan, and E. Howley, "A meta optimisation analysis of particle swarm optimisation velocity update equations for watershed management learning," *Appl. Soft Comput.*, vol. 62, pp. 148–161, Jan. 2018.

[32] M. Taherkhani and R. Safabakhsh, "A novel stability-based adaptive inertia weight for particle swarm optimization," *Appl. Soft. Comput.*, vol. 38, pp. 281–295, Jan. 2016.

[33] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi, "Fuzzy self-tuning PSO: A settings-free algorithm for global optimization," *Swarm Evol. Comput.*, vol. 39, pp. 70–85, Apr. 2018.

[34] M.-G. Gan, M. Zhang, C.-Y. Zheng, and J. Chen, "An adaptive sliding mode observer over wide speed range for sensorless control of a brushless DC motor," *Control Eng. Pract.*, vol. 77, pp. 52–62, Aug. 2018.

[35] I. R. Petersen and D. C. McFarlane, "Optimal guaranteed cost control and filtering for uncertain linear systems," *IEEE Trans. Autom. Control*, vol. 39, no. 9, pp. 1971–1977, Sep. 1994.

[36] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*. New York, NY, USA: Wiley, 1972.

[37] D. L. Kleinman, "On an iterative technique for Riccati equation computations," *IEEE Trans. Autom. Control*, vol. AC-13, no. 1, pp. 114–115, Feb. 1968.

[38] J. Xu, Y. Tan, J. Gao, and E. Feng, "Pricing currency option based on the extension principle and defuzzification via weighting parameter identification," *J. Appl. Math*, vol. 2013, Jan. 2013, Art. no. 623945.

[39] Q. Quanser, *2 DOF Helicopter User Control Manual*. Markham, ON, Canada: Quanser Inc., 2006.

[40] N. Hansen, R. Ros, N. Mauny, M. Schoenauer, and A. Auger, "Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5755–5769, Dec. 2011.

MENG ZHANG received the B.E. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree in control science and engineering with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation.

His main research interests include reinforcement learning and adaptive optimal control.

Mr. Zhang received the Best Paper Award at the 11th Asian Control Conference.

MING-GANG GAN (M'17) received the B.E. and Ph.D. degrees in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2001 and 2007, respectively. From 2015 to 2016, he was a Visiting Scholar with New York University.

He is currently a Professor with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology. His main research interests include intelligent information processing and intelligent control.

Dr. Gan received the Best Paper Award at the 11th Asian Control Conference.