Reinforcement-learning-based adaptive optimal flight control with output feedback and input constraints

Sun, Bo; van Kampen, Erik Jan

DOI
10.2514/1.G005715

Publication date
2021

Document Version
Accepted author manuscript

Published in
Journal of Guidance, Control, and Dynamics

Citation (APA)
Sun, B., & van Kampen, E. J. (2021). Reinforcement-learning-based adaptive optimal flight control with output feedback and input constraints. *Journal of Guidance, Control, and Dynamics*, *44*(9), 1685-1691. https://doi.org/10.2514/1.G005715

Reinforcement Learning-Based Adaptive Optimal Flight Control with Output Feedback and Input Constraints

Bo Sun [*] and Erik-Jan van Kampen [†]
Delft University of Technology, Delft, The Netherlands, 2629 HS

I. Introduction

This note aims at improving the present incremental model-based global dual heurisitic programming algorithm proposed in our recent work [1] by taking the output-feedback situation and input constraints into consideration. Different from the common incremental model that is based on full-state feedback, an extended incremental model utilizing previous input/output data is introduced to identify locally linearized system dynamics for nonlinear systems. A non-quadratic performance function combined with a constrained-output actor network guarantees that the produced control input command satisfies actuator saturation constraints. Through numerical simulations, the effectiveness and the feasibility of the proposed method are verified.

Reinforcement learning (RL) has become a promising tool for improving autonomy in various types of aerospace systems [1–6] because of its self-learning property sprouting from psychological and neuroscientific perspectives on animal behaviour [7]. By interacting with the environment, RL can make a system learn optimal policies to achieve goals with limited or no priori knowledge of its dynamics or environment, and have the capability of adapting to changing situations [2, 7]. These advantages enable RL provide a normative solution to adaptive optimal control [8].

One branch of RL is adaptive dynamic programming (ADP), which is developed from dynamic programming (DP), and it is performed in a forward-in-time way that allows for an online implementation [9]. ADP is often implemented with artificial neural networks (ANNs) and the actor-critic scheme [10], leading to adaptive critic design (ACD) [11], whose simple diagram is depicted in Fig. 1. ACDs not only handle the well-known "curse of dimensionality" [10], but also have a stronger generalization capability to deal with nonlinearity [9, 12–15]. According to the information that the critic network approximates, ACDs can generally be categorized into three groups as heuristic dynamic programming (HDP) [16, 17], dual HDP (DHP) [11, 18], and global DHP (GDHP) [1, 9, 19].Among them, GDHP combines the information utilized by HDP and DHP, i.e., the performance function and its derivatives. The conventional GDHP utilizes a straight-forward form [19], where two kinds of outputs of the critic network share the same input and hidden layers, which brings in couplings. Furthermore, due to the approximating property, inconsistent errors exist between these outputs. Nevertheless, the structure with explicit analytical calculations proposed in [1] can overcome these limitations and therefore it is investigated in this note.

[*] PhD student, Control and Operations Department, Faculty of Aerospace Engineering, South Holland

[†] Assistant Professor, Control and Operations Department, Faculty of Aerospace Engineering, South Holland, AIAA member

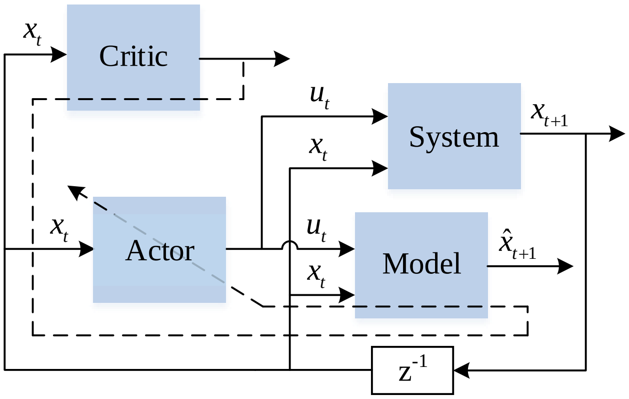

Fig. 1 The simple diagram of ACDs, where the actor network generates the control policy that is evaluated by the critic network, and the model network is utilized to approximate system dynamics. Solid lines are the feedforward flow of signals, the dashed line denotes the updating pathway. x and u respectively denotes the system state and the control input, $\hat{x}_{t+1}$ is the estimated value of x_{t+1}, and the subscript denotes the time instant.

Although [8] claims that RL can perform in a *direct* manner with no need of identifying the system dynamics, as shown in Fig. 1, a third model module, next to the actor and critic networks, is often introduced to provide system transition information and thus to speed up learning and increase the success ratio [16]. By convention, ACDs rely on an ANN to approximate the global system dynamics [9, 16], which can be intractable to obtain for complex aerospace systems and can face difficulties when changing conditions are encountered [1, 11, 18]. Consequently, an incremental model (IM) is utilized in [1, 4, 6, 11, 18] to identify the locally linearized dynamics online so as to reduce the dependency on global models. As an improved version of GDHP, the IM-based GDHP (IGDHP) has shown better performance in optimal tracking control problems (OTCPs) in the full-state feedback (FSF) condition [1].

Nevertheless, for real systems, sometimes not only the system dynamics, but also the measurements of some internal states are not available [17], which leads to output-feedback (OPFB) problems that cannot be tackled by the current IGDHP method. Although some ADP algorithms have been developed for OPFB [4, 6, 20], these algorithms are derived in a linear form with a quadratic performance function. These existing algorithms depend on solving a linear Riccati equation, and therefore are unable to handle complex nonlinear demands in the optimal control task, such as input constraints. However, handling input constraints is a common demand for control systems in many applications such as aerospace systems [21, 22]. A non-quadratic performance function is introduced to cope with actuator saturation constraints for optimal regulation control problems (ORCPs) [23], but cannot directly be applied to OTCPs [14]. Although an augmented system is proposed for HDP in [14, 15] to tackle this limitation, introducing the reference signal in addition to the tracking error into the actor and critic networks can slow down learning.

Motivated by overcoming the limitations existing in the current IGDHP method, this note first of all develops an extended incremental model to deal with OPFB problems. Then, a non-quadratic performance function as well as bounding actor network is introduced to handle input constraints. Finally, numerical simulations are executed to verify the novel IGDHP method.

II. Incremental Model with Output Feedback

The situation investigated in this note is that the system dynamics are unknown and only input/output information can be acquired, so an incremental model is constructed to approximate the state transformation.

Consider an affine nonlinear discrete system represented by:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) + g(\mathbf{x}_t)\mathbf{u}(\mathbf{x}_t), \tag{1}$$

$$\mathbf{y}_t = h(\mathbf{x}_t), \tag{2}$$

where $\mathbf{x}_t \in \mathbb{R}^n$, $\mathbf{u}_t \in \mathbb{R}^m$ and $\mathbf{y}_t \in \mathbb{R}^p$ are the system state vector, control input vector and measurable output state vector at the time instant t, respectively, $f(\mathbf{x}_t) \in \mathbb{R}^n$, $g(\mathbf{x}_t) \in \mathbb{R}^{n\times m}$ and $h(\mathbf{x}_t)$ denote the drift dynamics, input dynamics and output dynamics of the system, respectively. It is assumed that $f(\mathbf{x}_t)$, $g(\mathbf{x}_t)$ and $h(\mathbf{x}_t)$ are Lipschitz continuous on their domains. The nonlinear system is assumed to be both controllable and observable. For simplicity, $\mathbf{u}(\mathbf{x}_t)$ is represented by $\mathbf{u}_t$ in the rest of this paper.

According to [1, 11], if the system is full-state feedback, around time instant $t-1$, the nonlinear system (1) can approximately be written into the following linear equation by taking the first order Taylor series expansion and omitting second and higher-order terms:

$$\begin{aligned} \mathbf{x}_{t+1} &\approx f(\mathbf{x}_{t-1}) + \mathbf{F}_{t-1}(\mathbf{x}_t - \mathbf{x}_{t-1}) + g(\mathbf{x}_{t-1})\mathbf{u}_{t-1} + \mathbf{G}_{t-1}(\mathbf{u}_t - \mathbf{u}_{t-1}) \\ &= \mathbf{x}_t + \mathbf{F}_{t-1}(\mathbf{x}_t - \mathbf{x}_{t-1}) + \mathbf{G}_{t-1}(\mathbf{u}_t - \mathbf{u}_{t-1}), \end{aligned} \tag{3}$$

where $\mathbf{F}_{t-1} = \frac{\partial f^{\mathsf{T}}(\mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}} \in \mathbb{R}^{n\times n}$ and $\mathbf{G}_{t-1} = \frac{\partial g^{\mathsf{T}}(\mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}} \in \mathbb{R}^{n\times m}$ are the state transition matrix and the input distribution matrix, respectively. $\mathbf{F}_{t-1}$ and $\mathbf{G}_{t-1}$ are bounded due to the Lipschitz continuity of $f(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ in Eq. (1).

Equation (3) can be rewritten as:

$$\Delta\mathbf{x}_{t+1} \approx \mathbf{F}_{t-1}\Delta\mathbf{x}_t + \mathbf{G}_{t-1}\Delta\mathbf{u}_t. \tag{4}$$

Similarly, the system output equation (2) can also be linearized using Taylor expansion around time instant t:

$$\mathbf{y}_{t+1} \approx \mathbf{y}_t + \mathbf{H}_t(\mathbf{x}_{t+1} - \mathbf{x}_t), \tag{5}$$

where $\mathbf{H}_t = \frac{\partial h^{\mathsf{T}}(\mathbf{x}_t)}{\partial \mathbf{x_t}} \in \mathbb{R}^{p\times n}$ denotes the observation matrix. Equation (5) can be rewritten as:

$$\Delta \mathbf{y}_{t+1} \approx \mathbf{H}_t \Delta \mathbf{x}_{t+1}, \tag{6}$$

To identify and control the new incremental model presented by Eqs. (4) and (6), the following two assumptions are required.

Assumption 1 *The linearization does not change the property of controllablity and observability of the original system given by Eqs.* (1) *and* (2)*, i.e.* $(\mathbf{F}_{t-1}, \mathbf{G}_{t-1})$ *is controllable and* $(\mathbf{F}_{t-1}, \mathbf{H}_t)$ *is observable.*

Assumption 2 *The system is deterministic within the range of M steps, where $M \geq n/p$.*

Remark 1 *Assumption 2 is the prerequisite that the system can be identified via input/output data. It has practical significance in that real systems are often influenced by stochastic factors such as measurement noises, unmodeled states and unforeseen disturbances, while in a small range of time horizon, the impact is small enough to be ignored. Assumption 2 can be satisfied when the sampling frequency is high enough.*

It has been proved that for a deterministic observable system, the unmeasurable internal states (full states) can be reconstructed uniquely with adequate previous observations and control inputs [4, 6, 20]. Therefore, provided the input/output data over a sufficiently long time horizon, $[t - N + 1, N]$, $n/p \leq N \leq M$, we can construct a new system called the extended system. The extended system regards the previous increments of the input/output data as its system states. It can determine the next output increment $\Delta \mathbf{y}_{t+1}$ uniquely as follows:

$$\begin{aligned} \Delta \mathbf{y}_{t+1} &\approx \underline{\mathbf{F}}_t \overline{\Delta \mathbf{y}}_{t,N} + \underline{\mathbf{G}}_t \overline{\Delta \mathbf{u}}_{t,N} \\ &= \underline{\mathbf{F}}_{11,t} \Delta \mathbf{y}_t + \underline{\mathbf{F}}_{12,t} \overline{\Delta \mathbf{y}}_{t-1,N-1} + \underline{\mathbf{G}}_{11,t} \Delta \mathbf{u}_t + \underline{\mathbf{G}}_{12,t} \overline{\Delta \mathbf{u}}_{t-1,N-1}, \end{aligned} \tag{7}$$

where $\underline{\mathbf{F}}_t \in \mathbb{R}^{p \times Np}$ and $\underline{\mathbf{G}}_t \in \mathbb{R}^{p \times Nm}$ are the transition matrix and input distribution matrix of the extended discrete system, respectively, $\overline{\Delta \mathbf{u}}_{t,N} = [\Delta \mathbf{u}_t^\mathsf{T}, \Delta \mathbf{u}_{t-1}^\mathsf{T}, \cdots, \Delta \mathbf{u}_{t-N+1}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{Nm}$ and $\overline{\Delta \mathbf{y}}_{t,N} = [\Delta \mathbf{y}_t^\mathsf{T}, \Delta \mathbf{y}_{t-1}^\mathsf{T}, \cdots, \Delta \mathbf{y}_{t-N+1}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{Np}$ are the measured input/output data from N previous steps, respectively, $\underline{\mathbf{F}}_{11,t} \in \mathbb{R}^{p \times p}$ and $\underline{\mathbf{F}}_{12,t} \in \mathbb{R}^{p \times (N-1)p}$ are partitioned matrices from $\underline{F}_t$, and $\underline{\mathbf{G}}_{11,t} \in \mathbb{R}^{p \times m}$ and $\underline{\mathbf{G}}_{12,t} \in \mathbb{R}^{p \times (N-1)m}$ are partitioned matrices from $\underline{G}_t$. Assume that the generalised inverse of $\underline{\mathbf{G}}_{11,t}$ exists such that $\underline{\mathbf{G}}_{11,t}^{-1} \underline{\mathbf{G}}_{11,t} = \mathbf{I}_m \in \mathbb{R}^{m \times m}$, where $\mathbf{I}_m$ denotes the identity matrix and the subscript m gives the dimensionality.

In this way, the original nonlinear system is approximated by a locally linear incremental model and a direct mapping from the control input at the time instant t to the output at the time instant $t + 1$ is built. Then, a recursive least squares (RLS) approach using a sliding window technique [17] is adopted to identify the matrices $\underline{\mathbf{F}}_t$ and $\underline{\mathbf{G}}_t$ online. Rewrite

Eq. (7) in a vector form as:

$$\Delta \mathbf{y}_{t+1} \approx \begin{bmatrix} \overline{\Delta \mathbf{y}}_{t,N}^{\mathsf{T}} & \overline{\Delta \mathbf{u}}_{t,N}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{F}}_t^{\mathsf{T}} \\ \\ \underline{\mathbf{G}}_t^{\mathsf{T}} \end{bmatrix}. \tag{8}$$

Define $\overline{\mathbf{Y}}_t = \left[\overline{\Delta \mathbf{y}}_{t,N}^{\mathsf{T}}, \overline{\Delta \mathbf{u}}_{t,N}^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{N(p+m)\times 1}$, which is the input information of the extended incremental model identification, and $\underline{\Theta}_t = \left[\underline{\mathbf{F}}_t, \underline{\mathbf{G}}_t\right]^{\mathsf{T}} \in \mathbb{R}^{N(p+m)\times p}$, which is the extended matrix to be determined using the RLS algorithm. Therefore, the output prediction equation can presented as follows:

$$\Delta \hat{\mathbf{y}}_{t+1} = \overline{\mathbf{Y}}_t^{\mathsf{T}} \cdot \underline{\hat{\Theta}}_t, \tag{9}$$

where $\underline{\hat{\Theta}}_t = \left[\underline{\hat{\mathbf{F}}}_t, \underline{\hat{\mathbf{G}}}_t\right]^{\mathsf{T}}$ is the approximated value of $\underline{\Theta}_t$, and the symbol $\hat{\cdot}$ denotes the approximated/estimated value.

The sliding window is utilized to store historical data $\overline{\mathbf{Y}}_t$ for determining $\underline{\hat{\Theta}}_t$ in each time step, and the main procedure of the RLS approach is given as follows [17]:

$$\epsilon_t = \Delta \mathbf{y}_{t+1}^{\mathsf{T}} - \Delta \hat{\mathbf{y}}_{t+1}^{\mathsf{T}}, \tag{10}$$

$$\underline{\hat{\Theta}}_t = \underline{\hat{\Theta}}_{t-1} + \frac{\underline{\mathrm{Cov}}_{t-1} \overline{\mathbf{Y}}_t}{\gamma_{\mathrm{RLS}} + \overline{\mathbf{Y}}_t^{\mathsf{T}} \underline{\mathrm{Cov}}_{t-1} \overline{\mathbf{Y}}_t} \epsilon_t, \tag{11}$$

$$\underline{\mathrm{Cov}}_t = \frac{1}{\gamma_{\mathrm{RLS}}} \left(\underline{\mathrm{Cov}}_{t-1} - \frac{\underline{\mathrm{Cov}}_{t-1} \mathbf{X}_t \mathbf{X}_t^{\mathsf{T}} \underline{\mathrm{Cov}}_{t-1}}{\gamma_{\mathrm{RLS}} + \mathbf{X}_t^{\mathsf{T}} \underline{\mathrm{Cov}}_{t-1} \mathbf{X}_t} \right), \tag{12}$$

where $\epsilon_t \in \mathbb{R}^p$ denotes the prediction error, $\underline{\mathrm{Cov}}_t \in \mathbb{R}^{(p+m)N\times(p+m)N}$ is the estimation covariance matrix , and γ_{RLS} denotes the forgetting factor. As to initialization settings of the RLS approach, we set $\underline{\hat{\mathbf{F}}}_0 = [\mathbf{I}_p, 0]$, and $\underline{\hat{\mathbf{G}}}_0$ as a zero matrix. The covariance matrix is initialized as an identity matrix multiplied by a large positive value [6] and we choose 10^7 in this note, i.e., $\underline{\mathrm{Cov}}_0 = 10^7 \mathbf{I}_{(p+m)N}$.

III. Optimal Tracking Control Problem (OTCP)

This section deals with the input constraints in the OTCP by designing a non-quadratic function. The OTCP aims to find the optimal control policy $\mathbf{u}_t^*$ such that the system described by (1) and (2) can track the reference trajectory $\mathbf{y}_t^{\mathrm{ref}} \in \mathbb{R}^p$ in an optimal manner by minimizing a predefined performance function. Furthermore, the control input must be constrained by a bound vector $\mathbf{u}_{\mathrm{b}}$, i.e. $|u_{i,t}| \leq u_{\mathrm{b}i}$ for $\forall u_{\mathrm{b}i} > 0, i = 1, \cdots, m$, where $u_{i,t}$ and u_{bi} denotes the elements of $\mathbf{u}_t$ and $\mathbf{u}_{\mathrm{b}}$, respectively.

To simplify the derivation and implementation of the algorithm, the reference trajectory $\mathbf{y}_t^{\mathrm{ref}}$ is supposed to satisfy the following assumption [6]:

Assumption 3 *The reference signal is slow-varying in comparison to the system dynamics, such that the increment of the reference signal between two time instants can be ignored.*

Accordingly, considering Eq. (7), the output tracking error at the time instant $t+1$ can be presented as:

$$\begin{aligned} \mathbf{e}_{t+1} &= \mathbf{y}_{t+1} - \mathbf{y}_{t+1}^{\text{ref}} \\ &\approx \mathbf{y}_t + \underline{\mathbf{F}}_t \overline{\Delta \mathbf{y}}_{t,N} + \underline{\mathbf{G}}_t \overline{\Delta \mathbf{u}}_{t,N} - (\mathbf{y}_t^{\text{ref}} + \Delta \mathbf{y}_{t+1}^{\text{ref}}) \\ &\approx \mathbf{e}_t + \underline{\mathbf{F}}_t \overline{\Delta \mathbf{y}}_{t,N} + \underline{\mathbf{G}}_t \overline{\Delta \mathbf{u}}_{t,N} \\ &\approx \mathbf{e}_t + \underline{\mathbf{F}}_t \overline{\Delta \mathbf{e}}_{t,N} + \underline{\mathbf{G}}_t \overline{\Delta \mathbf{u}}_{t,N}. \end{aligned} \tag{13}$$

Based on Assumption 3 and Eq. (13), the effect caused by the dynamics of reference signal is approximately shielded between two sampling instants. Therefore, the original OTCP is transformed into an ORCP, so that the non-quadratic performance function used in [12, 14, 15, 23] can be adopted to generate constrained control input.

Then the following performance function is introduced for this new input-constrained ORCP:

$$J(\mathbf{e}_t, \mathbf{u}_t) = \sum_{l=t}^{\infty} \gamma^{l-t} c(\mathbf{e}_l, \mathbf{u}_l), \tag{14}$$

where γ is the discount factor with $0 < \gamma \leq 1$ and $c(\mathbf{e}_l, \mathbf{u}_l)$ is the one-step cost function that is defined as:

$$c(\mathbf{e}_t, \mathbf{u}_t) = \mathbf{e}_t^{\top} \mathbf{Q} \mathbf{e}_t + Y(\mathbf{u}_t), \tag{15}$$

where $\mathbf{Q} \in \mathbb{R}^{p \times p}$ is positive semi-definite and is set to be a diagonal matrix in this note, and $Y(\mathbf{u}_t)$ is a positive-definite integral function defined as:

$$Y(\mathbf{u}_t) = 2 \sum_{i=1}^{m} \int_0^{u_{i,t}} u_{bi} \psi^{-1}(\upsilon_i / u_{bi}) R_i d\upsilon_i, \tag{16}$$

where $\psi(\cdot)$ is a bounded element-wise function satisfying $|\psi(\cdot)| \leq 1$, and is a monotonic odd function with its derivative bounded by a constant ψ_M, i.e. $||d\psi(s)/ds|| \leq \psi_M$, $\forall s \in \mathbb{R}$, and R_i denotes the element of the positive definite weight matrix $\mathbf{R} = \text{diag}([R_1, \cdots, R_m]) \in \mathbb{R}^{m \times m}$, in which diag$(\cdot)$ reshapes the vector to a diagonal matrix. Without loss of generality, the well-known hyperbolic tangent function $\psi(\cdot) = \tanh(\cdot)$ is chosen as bounding function. Note that $Y(\mathbf{u}_t)$ is positive definite since $\psi^{-1}(\cdot)$ is a monotonic odd function and $\mathbf{R}$ is positive definite. For simplicity, $J(\mathbf{e}_t, \mathbf{u}_t)$ is denoted by J_t and $c(\mathbf{e}_t, \mathbf{u}_t)$ is denoted by c_t hereafter.

According to Bellman's principle of optimality, the optimal performance function J_t^* is time invariant and satisfies the discrete-time Hamilton-Jacobi-Bellman (DTHJB) equation:

$$J_t^* = \min_{\mathbf{u}_t} (c_t + \gamma J_{t+1}^*). \tag{17}$$

Differentiate the right-hand side of Eq. (17) along the control input $\mathbf{u}_t$ and the following equation should be satisfied for the optimal control $\mathbf{u}^*_t$ [13]:

$$\frac{\partial \mathbf{c}_t}{\partial \mathbf{u}_t} + \gamma \frac{\partial \mathbf{e}^{\mathsf{T}}_{t+1}}{\partial \mathbf{u}_t} \lambda^*_{t+1} = 0. \tag{18}$$

where $\lambda^*_{t+1} = \frac{\partial J^*_{t+1}}{\partial \mathbf{e}_{t+1}}$. Then substituting Eqs. (15) and (16) in Eq. (18) yields [12, 14, 23]:

$$u^*_{i,t} = -u_{bi} \tanh(D^*_{i,t}), i = 1, \cdots, m, \tag{19}$$

and $D^*_{i,t}$ is given as:

$$D^*_{i,t} = \frac{\gamma}{2u_{bi}} R_i^{-1} \underline{g}^{\mathsf{T}}_{i,11,t} \lambda^*_{t+1}, \tag{20}$$

where $\underline{g}_{i,11,t}$ is the ith column vector of $\underline{\mathbf{G}}_{11,t}$. The control input $u^*_{i,t}$ is bounded within its permitted range $[-u_{\mathrm{b}i}, u_{\mathrm{b}i}]$, $i = 1, \cdots, m$. The nonquadratic cost (16) for $\mathbf{u}^*_t$ is:

$$Y(\mathbf{u}^*_t) = \sum_{i=1}^{m} [u_{bi} \gamma \lambda^{*\mathsf{T}}_{t+1} \underline{g}_{i,11,t} \tanh(D^*_{i,t}) + u^2_{bi} R_i \ln(1 - \tanh^2(D^*_{i,t}))]. \tag{21}$$

By substituting Eq. (21) into Eq. (17), the DTHJB equation becomes:

$$J^*_t = \mathbf{e}^{\mathsf{T}}_t \mathbf{Q} \mathbf{e}_t + \sum_{i=1}^{m} [u_{bi} \gamma \lambda^{*\mathsf{T}}_{t+1} \underline{g}_{i,11,t} \tanh(D^*_{i,t}) + u^2_{bi} R_i \ln(1 - \tanh^2(D^*_{i,t}))] + \gamma J^*_{t+1}. \tag{22}$$

In this way, the original OTCP is recast as a new ORCP subject to input constraints. The DTHJB equation (22) cannot be solved analytically in the generally nonlinear cases, and therefore the IGDHP algorithm is introduced to iteratively solve the OTCP in the next section.

IV. IGDHP Implementation

The IGDHP algorithm is introduced in this section with the IM and ANNs facilitating the implementation: the IM reflects the local dynamics of the nonlinear plant and the ANNs are utilized to build the critic network and the actor network. The architecture of the IGDHP algorithm is shown in Fig. 2. The reference signal at the time instant $t+1$, $\mathbf{y}^{\mathrm{ref}}_{t+1}$, is unavailable at the time instant t. To obviate the need for this information, the actor and critic networks are updated with the information from the time instants t and $t-1$ [1, 15].

A. The critic network

The IGDHP technique makes use of both the approximation of performance function $\hat{J}_t$ and its derivative with respect to the network input $\mathbf{e}_t$, which is denoted by $\hat{\lambda}_t$. As shown in Fig. 3, the critic network is utilized to approximate the performance function (14) with the facilitation of an ANN, which employs a feedforward structure with single

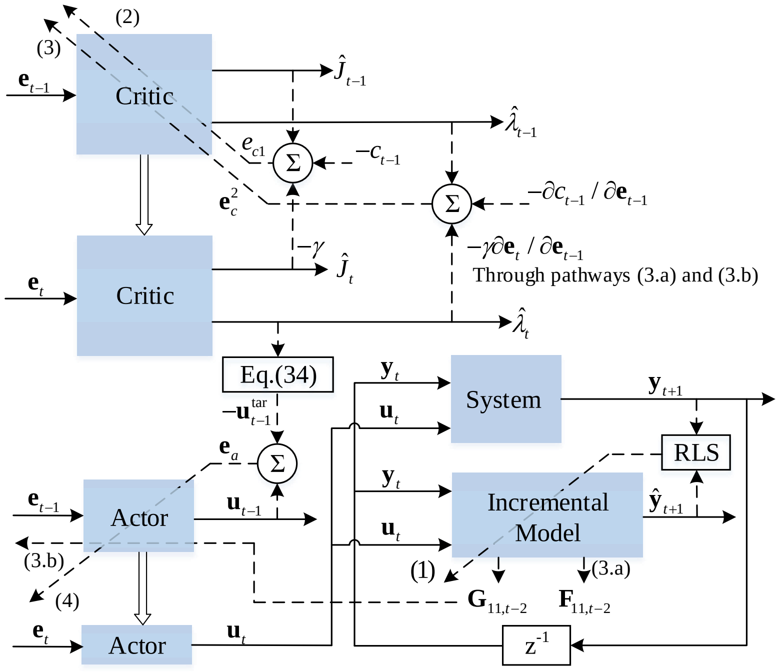

Fig. 2 The architecture of the IGDHP algorithm, where solid lines represent the feedforward flow of signals, dashed lines are backpropagation pathways, and the thick arrows represent the weight transmission.

hidden layer as follows:

$$\hat{J}_t = \mathbf{w}_{c2,t}^{\mathsf{T}} \sigma(\mathbf{w}_{c1,t}^{\mathsf{T}} \mathbf{e}_t), \tag{23}$$

where $\mathbf{w}_{c1,t}$ and $\mathbf{w}_{c2,t}$ are weight matrices between different layers and the activation function σ is chosen to be a sigmoid function. By taking the explicit analytical calculations [1], $\hat{\lambda}_t$ is given as:

$$\hat{\lambda}_t = \frac{\partial \hat{J}_t}{\partial \mathbf{e}_t} = \mathbf{w}_{c1,t}(\mathbf{w}_{c2,t} \odot \sigma'(\mathbf{w}_{c1,t}^{\mathsf{T}} \mathbf{e}_t)), \tag{24}$$

where $\odot$ is the Hadamard product, a.k.a. the element-wise product, and $\sigma'(\bullet)$ is the first order derivative of $\sigma(\bullet)$.

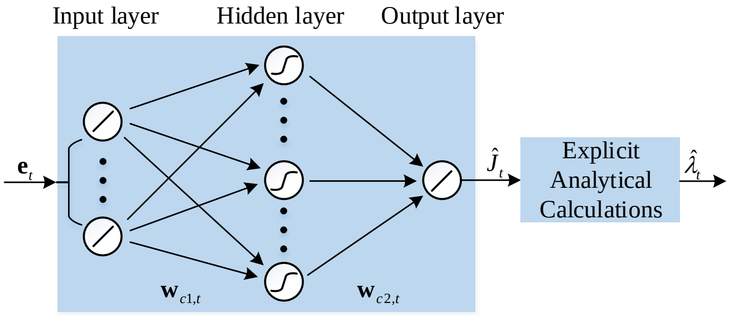

Fig. 3 The architecture of the critic network, in which an ANN is utilized to approximate performance function and then explicit analytical calculations are taken to compute first-order derivatives.

Referring to the DTHJB equation (17), the approximation error of the performance function produced by the critic network is given as:

$$e_{c1,t} = \hat{J}_{t-1} - c_{t-1} - \gamma \hat{J}_t, \tag{25}$$

and the approximation error of the derivative is given as:

$$\mathrm{e}_{c2,t} = \frac{\partial(\hat{J}_{t-1} - c_{t-1} - \gamma \hat{J}_t)}{\partial \mathbf{e}_{t-1}} = \hat{\lambda}_{t-1} - \frac{\partial c_{t-1}}{\partial \mathbf{e}_{t-1}} - \gamma \frac{\partial \mathbf{e}_t}{\partial \mathbf{e}_{t-1}} \hat{\lambda}_t. \tag{26}$$

The second item on the right hand side of Eq. (26) has an explicit calculation as follows:

$$\frac{\partial c_{t-1}}{\partial \mathbf{e}_{t-1}} = 2\mathbf{Q}\mathbf{e}_{t-1} + 2\frac{\partial \mathbf{u}_{t-1}}{\partial \mathbf{e}_{t-1}}[\mathbf{R}(\tanh^{-1}(\mathbf{u}_{t-1} \odot \mathbf{u}_{\mathrm{b}}^{\circ -1}) \odot \mathbf{u}_{\mathrm{b}})], \tag{27}$$

where $\frac{\partial \mathbf{u}_{t-1}}{\partial \mathbf{e}_{t-1}}$ is derived by the actor network in the next subsection, and $\mathbf{u}_{\mathrm{b}}^{\circ -1}$ stands for the element-wise inverse of the vector $\mathbf{u}_{\mathrm{b}}$. $\frac{\partial \mathbf{e}_t}{\partial \mathbf{e}_{t-1}}$ in the last item in Eq. (26) is composed of two parts [1, 9, 11]: one part is directly derived from the incremental model (pathway 3.a), whereas the other part starts from the incremental model and uses the control input $\mathbf{u}_{t-1}$ as the intermediate auxiliary (pathway 3.b):

$$\frac{\partial \hat{\mathbf{e}}_t}{\partial \mathbf{e}_{t-1}} = \underbrace{\mathbf{I}_p + \mathbf{F}_{11,t-2}^{\mathsf{T}}}_{\text{pathway }(3.a)} + \underbrace{\frac{\partial \mathbf{u}_{t-1}}{\partial \mathbf{e}_{t-1}} \mathbf{G}_{11,t-2}^{\mathsf{T}}}_{\text{pathway }(3.b)}, \tag{28}$$

where $\underline{\mathbf{F}}_{11,t-2} \in \mathbb{R}^{p \times p}$ is the upper-left partitioned matrix from $\underline{\mathbf{F}}_{t-2}$ and $\underline{\mathbf{G}}_{11,t-2} \in \mathbb{R}^{p \times m}$ is the upper partitioned matrix from $\underline{\mathbf{G}}_{t-2}$.

Accordingly, the overall error of the critic network combines two kinds of approximation error as:

$$E_{c,t} = \beta \frac{1}{2} e_{c1,t}^2 + (1-\beta)\frac{1}{2} \mathrm{e}_{c2,t}^{\mathsf{T}} \mathrm{e}_{c2,t}, \tag{29}$$

where β is a scalar within a range of $[0, 1]$.

The weights of the critic network are updated by a gradient-descent algorithm with a learning rate η_c to minimize the overall error $E_{c,t}$:

$$\mathbf{w}_{c,t+1} = \mathbf{w}_{c,t} - \eta_c \frac{\partial E_{c,t}}{\partial \mathbf{w}_{c,t}}, \tag{30}$$

and

$$\frac{\partial E_{c,t}}{\partial \mathbf{w}_{c,t}} = \frac{\partial \hat{J}_t}{\partial \mathbf{w}_{c,t}} \cdot \frac{\partial E_{c,t}}{\partial \hat{J}_t} + \frac{\partial \hat{\lambda}_t}{\partial \mathbf{w}_{c,t}} \cdot \frac{\partial E_{c,t}}{\partial \hat{\lambda}_t} = \beta \frac{\partial \hat{J}_t}{\partial \mathbf{w}_{c,t}} e_{c1,t} + (1-\beta)\frac{\partial \hat{\lambda}_t}{\partial \mathbf{w}_{c,t}} \mathrm{e}_{c2,t}, \tag{31}$$

where $\partial \hat{\lambda}_{c,t} / \partial \mathbf{w}_{c,t}$ is the second-order mixed gradient of $\hat{J}_t$, and the detailed explicit calculations can be found in [1].

B. The actor network

The pathway 3.b needs to compute $\frac{\partial \mathbf{u}_{t-1}}{\partial \mathbf{e}_{t-1}}$, which cannot be calculated exactly. Consequently, an actor network is introduced to produce the control input $\mathbf{u}_t$ and to facilitate backpropagation.

In this note, the output layer of the actor network employs a bounded element-wise function as Eq. (16) to be the activation function, and is multiplied by the bound vector $\mathbf{u}_b$, so that the system control $\mathbf{u}_t$ output of the actor network is bounded within the constraints, as shown in Fig. 4. The actor network is also constructed as a single-hidden-layer feedforward ANN:

$$\mathbf{u}_t = \mathbf{u}_b \odot \psi(\mathbf{w}_{a2,t}^{\mathsf{T}} \sigma(\mathbf{w}_{a1,t}^{\mathsf{T}} \mathbf{e}_t)), \tag{32}$$

where $\mathbf{w}_{a1,t}$ and $\mathbf{w}_{a2,t}$ are weight matrices between different layers of the actor network.

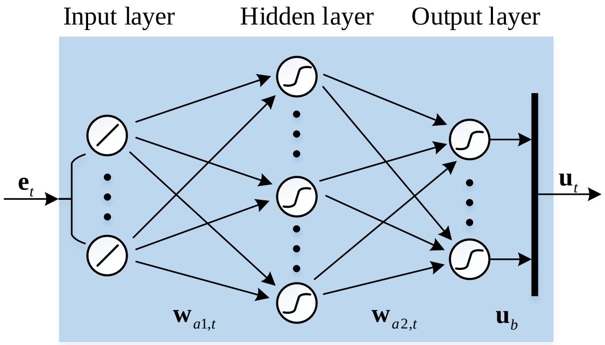

Fig. 4 The architecture of the actor network, in which the output is constrained by the bounded activation function and the bound vector.

It is noted that the control input $\mathbf{u}_t$ outputted by the actor network is directly introduced to the IM and the real system, and that the actor network performs as a global approximation, so $\frac{\partial \mathbf{u}_{t-1}}{\partial \mathbf{e}_{t-1}} = \frac{\partial \mathbf{u}_t}{\partial \mathbf{e}_t}$ given the same actor weights. The actor network is supposed to approximate a target control input $\mathbf{u}_{t-1}^{\text{tar}}$ which is obtained by substituting $\hat{\lambda}_t$ into Eq. (20) and then Eq. (19) as follows:

$$\mathbf{u}_{t-1}^{\text{tar}} = -\mathbf{u}_{\text{b}} \odot \tanh(\hat{D}_{t-1}), \tag{33}$$

and

$$\hat{D}_{t-1} = \frac{\gamma}{2} \mathbf{u}_{\text{b}}^{\circ -1} \odot (\mathbf{R}^{-1} \mathbf{G}_{11,t-2}^{\mathsf{T}} \hat{\lambda}_t). \tag{34}$$

It can be seen that both the target control input $\mathbf{u}_{t-1}^{\text{tar}}$ and the real control input $\mathbf{u}_t$ are bounded by $\mathbf{u}_{\text{b}}$. Therefore, the actor network is aiming at minimizing the following error:

$$E_{a,t} = \frac{1}{2} \mathbf{e}_{a,t}^{\mathsf{T}} \mathbf{e}_{a,t}, \tag{35}$$

where

$$\mathbf{e}_{a,t} = \mathbf{u}_{t-1} - \mathbf{u}_{t-1}^{\mathrm{tar}}. \tag{36}$$

As illustrated in Fig. 2, the actor weights are updated along the 4th pathway with a learning rate η_a:

$$\mathbf{w}_{a,t+1} = \mathbf{w}_{a,t} - \eta_a \frac{\partial E_{a,t}}{\partial \mathbf{w}_{a,t}} = \mathbf{w}_{a,t} - \eta_a \frac{\partial \mathbf{u}_t}{\partial \mathbf{w}_{a,t}} \mathbf{e}_{a,t} \tag{37}$$

V. Flight Control Simulation

In order to assess the performance of the developed novel IGDHP algorithm, the longitudinal dynamics of a nonlinear aircraft [1, 24, 25] are taken into account. The initial altitude and speed of the aircraft are set to be 15000 ft and 600 ft/s, respectively, based on which, the aerodynamic model is trimmed. To simply and clearly compare different methods, only short period control is considered and the elevator deflection command is artificially bounded within $[-5\text{ deg}, 5\text{ deg}]$. The control system, discretized with a sampling frequency of 100 Hz, targets for controlling the angle of attack (AOA) of the aircraft to track a sinusoidal signal, namely $\alpha^{\mathrm{ref}} = 10\sin(0.5t)$ deg.

Zero-mean white noises with standard deviation of 4×10^{-2} deg and 1.8×10^{-3} deg are added onto the bounded elevator deflection command and measured AOA, respectively. A 3211 disturbance signal is employed to kick off the learning at the beginning so as to better satisfy the persistent excitation (PE) condition [1, 19]. Three methods are utilized for comparison, namely GDHP with input constraints, IGDHP with input constraints, and IGDHP without input constraints. All methods are implemented in the OPFB condition with the sliding window width of 3. GDHP employs a model network with preivious states and control inputs as its inputs to approximate system dynamics, whereas IGDHP approaches utilize an incremental model. All ANNs adopt the fully connected feed-forward architecture with a single hidden layer. The number of hidden layer neurons is 10 for the actor and critic networks and 20 for the model network. All weights are initialized randomly within $[-0.1, 0.1]$ to decrease the impact of initialization. For IGDHP without input constraints, the performance function is set to a quadratic form and the output layer of the actor network employs a unit linear activation function. To compare the robustness of these approaches, Monte Carlo simulations are also conducted, and the randomness is introduced by aforementioned noises and initial weights. A concept of success ratio [1, 11, 16] is introduced to indicate the performance. A successful implementation in this note is defined by the tracking errors remaining within $[-4\text{ deg}, 4\text{ deg}]$ after the first 20 seconds. The simulations are carried out on a Intel Core i7-8550U @ 1.80 GHz processor, and 8 GB RAM.

The comparison of AOA trajectories and tracking error is illustrated in Fig. 5. It can be seen that although all methods can follow the reference after short online learning stage, GDHP with input constraints takes more time to get satisfying performance and the tracking error is the largest overall. The reason causing this phenomenon lies in that GDHP utilizes a model network to identify the global system model which requires more data to update weights.

Without well approximated dynamics, the control policy cannot be appropriately generated, which in turn can have impacts on identification of the global model, i.e. Assumption 2 is not satisfied. A detailed view can be found in Fig. 6, which shows the weights between the input layer and hidden layer of the actor network $\mathbf{w}_{a1}$. The update of $\mathbf{w}_{a1}$ requires the information from the critic network, system dynamics and actor network, and therefore the trajectory of $\mathbf{w}_{a1}$ can be used to indicate the overall learning performance. It is clearly shown in Fig. 6 that the learning of GDHP is slower in comparison to the IGDHP methods, which results in a more conservative policy at the beginning stage, as presented in Fig. 7. Furthermore, due to the inaccurate information regarding system state transition, GDHP has the lowest success ratio for 1000 Monte Carlo simulations, which is merely 46.4%, compared with 99.4% and 98.0% for IGDHP with and without input constraints, respectively.

As to IGDHP methods, after the weights have converged, both methods have a similar tracking performance, which is better than that of GDHP. Nevertheless, the developed IGDHP with input constraints has a slightly higher success ratio, and the benefit is brought by the collective effect of the non-quadratic performance function and the bounded actor network. With these measures, input constraints can be overcome. As shown in Fig. 6, it is clear that the weight update of IGDHP without input constraints can be more radical at the beginning when the policy has not converged yet. During this exploration stage, IGDHP without input constraints performs similar to "bang-bang" control, with larger control command that easily causes overshoot and oscillation. To further investigate the influence of the measures to deal with input constraints, the policies directly produced by the actor network are compared between the IGDHP methods with and without input constraints. As presented in Fig. 8, given the random data of AOA and its reference within the range of $[-2 \text{ deg}, 2 \text{ deg}]$, the methods can plot a mesh surface to illustrate their learned policy at 8s. Compared to IGDHP with input constraints that has a smooth surface, IGDHP without input constraints has a sharper surface and tends to produce a large control command. The learned policy of GDHP with input constraints is similar to that of IGDHP with input constraints, and therefore its plot is omitted.

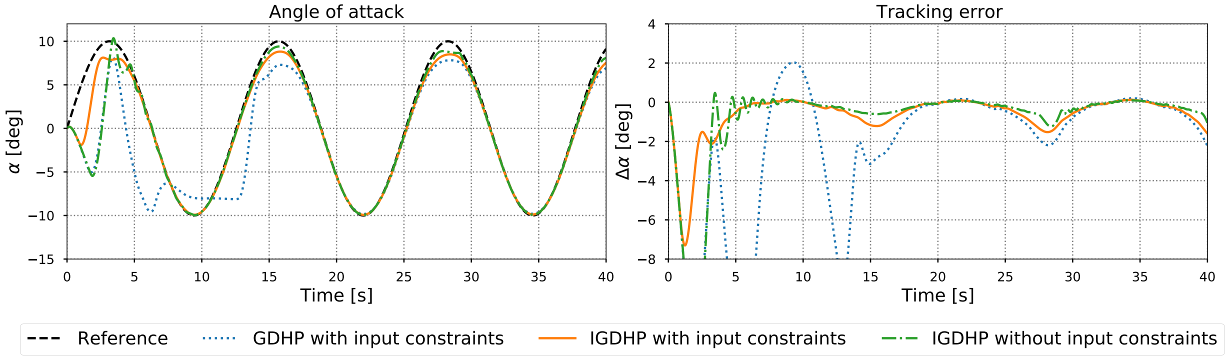

Fig. 5 The tracking performance of the online AOA tracking control task. Three methods are compared and IGDHP with input constraints is the contribution of this note.

To further verify the robustness of the designed control approach when tracking fast-varying reference signals,

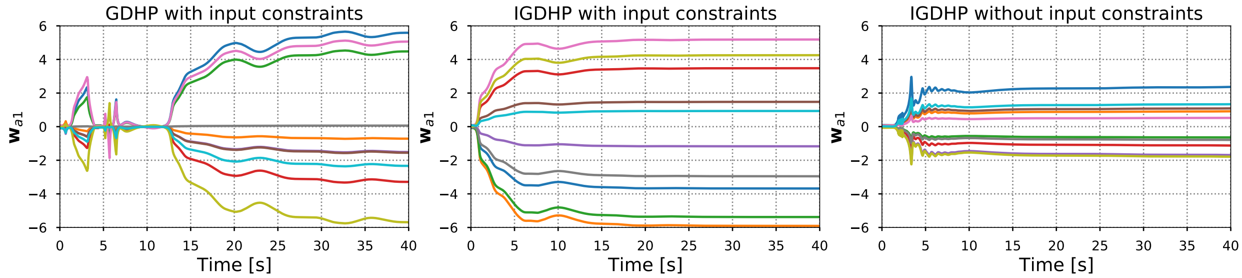

Fig. 6 The weights between the input layer and hidden layer of the actor network, w_{a1}. The plots of three approaches are presented, and the middle one refers to the proposed approach that is the contribution of this note.

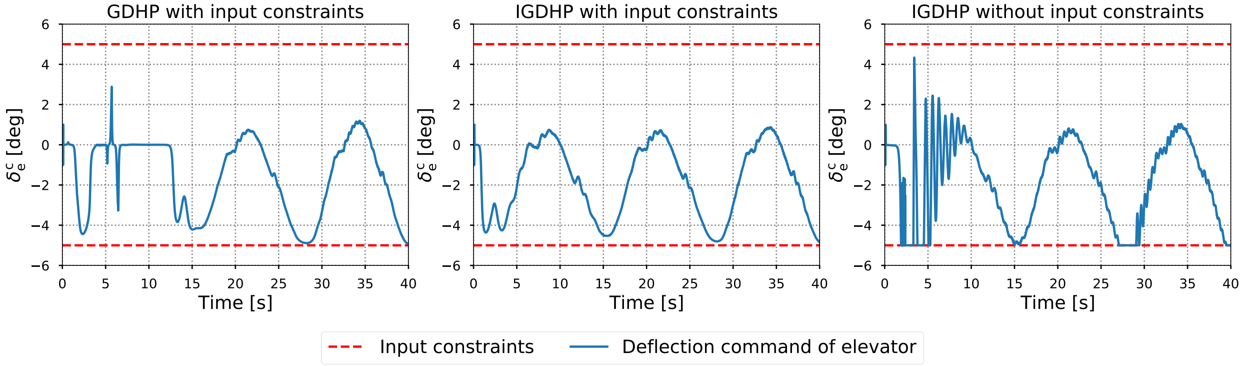

Fig. 7 The deflection command of elevator. The plots of three approaches are presented, and the middle one refers to the proposed approach that is the contribution of this note.

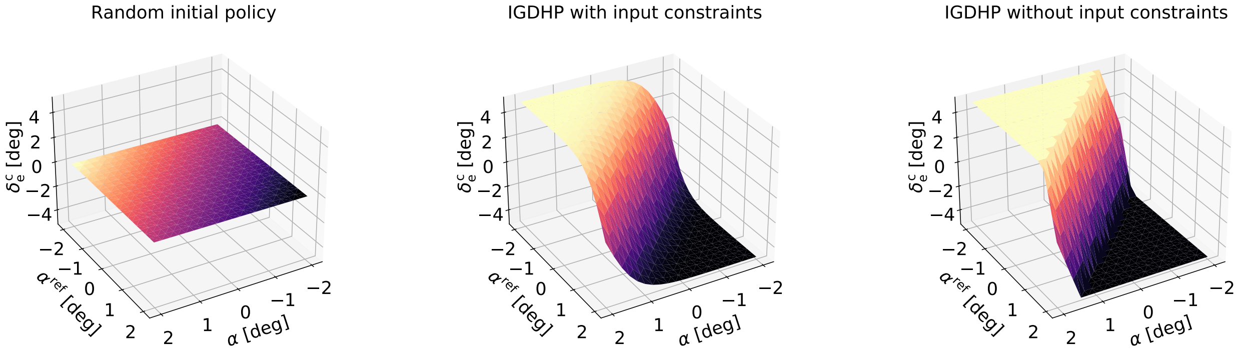

Fig. 8 The change of the policy directly produced by the actor network from the initial stage to 8π s. The first subfigure is the random initial policy, and the middle subfigure refers to the proposed approach that is the contribution of this note.

a simulation experiment in which the frequency of the reference signal keeps increasing is performed. The initial reference frequency is the same as the one in the above simulation experiments, and the control input is bounded within $[-25\ \text{deg}, 25\ \text{deg}]$. As illustrated in Fig. 9, after the initial exploration stage, the controlled AOA can track the given reference signal when it is slow-varying, and the tracking error is growing as the reference frequency is increasing. Specifically, when the reference frequency is around 5.3 times of the initial value, the tracking error for the first time exceeds 2 deg, and when the reference frequency is near 7.0 times of the initial value, the tracking error starts exceeding 4 deg. The results clarify the significance of Assumption 3 to a certain extent. Besides, it is noted that at the final stage of the simulation, the control input comes close to the input constraints but does not exceed the bound. Due to the existence of input constraints, when the reference frequency is too high, the aircraft cannot successfully complete the tracking task and the tracking error is large. Nevertheless, the simulation results demonstrate the developed IGDHP with input constraints is robust to the reference signal within a range of frequencies.

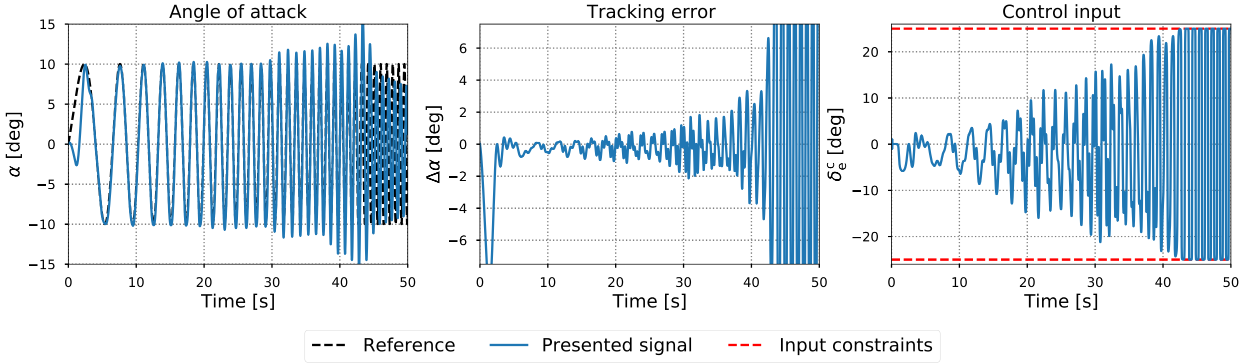

Fig. 9 The tracking performance of the developed IGDHP with input constraints when tracking a reference signal with its frequency keeping increasing.

VI. Conclusions

This note improves the current incremental model-based global dual heuristic programming (IGDHP) method for more complex application scenarios, including dealing with output feedback (OPFB) and input constraints. Different from the GDHP method that utilizes a neural network to identify the global system dynamics, the developed novel IGDHP method exploits an extended incremental model to approximate locally linear system dynamics via the previous input/output data at several previous time instants. The numerical simulation shows that the developed novel IGDHP method outperforms GDHP in convergent speed of parameters, tracking precision, and success ratio. Moreover, the input saturation constraint is overcome by combining a non-quadratic performance function and bound activation function in the output layer of the actor network. The original IGDHP method employs quadratic performance function and unit linear activation function, and compared to it, the developed novel IGDHP method has a smoother policy surface and slightly higher success ratio. In addition, through a simulation experiment, the robustness of the developed IGDHP with input constraints is verified for reference signals with a range of frequencies. The simulation results collectively demonstrate the effectiveness and the feasibility of the proposed method. Further research on better satisfying persistent excitation (PE) condition so as to achieve a non-failure control is recommended.

Acknowledgments

The authors would like to thank the Chinese Scholarship Council for financial support for B. Sun with the project reference number of 201806290007.

References

[1] Sun, B., and van Kampen, E.-J., "Incremental model-based global dual heuristic programming with explicit analytical calculations applied to flight control," *Engineering Applications of Artificial Intelligence*, Vol. 89, 2020, p. 103425. https://doi.org/10.1016/j.engappai.2019.103425.

[2] Junell, J., Mannucci, T., Zhou, Y., and van Kampen, E.-J., "Self-tuning gains of a quadrotor using a simple model for policy gradient reinforcement learning," *AIAA Guidance, Navigation, and Control Conference*, 2016, p. 1387. https://doi.org/10.2514/6.2016-1387.

[3] Ferrari, S., and Stengel, R. F., "Online adaptive critic flight control," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 5, 2004, pp. 777–786. https://doi.org/10.2514/1.12597.

[4] Zhou, Y., van Kampen, E.-J., and Chu, Q. P., "Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2016, pp. 493–496. https://doi.org/10.2514/1.G001762.

[5] Heydari, A., and Balakrishnan, S., "Adaptive critic-based solution to an orbital rendezvous problem," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2014, pp. 344–350. https://doi.org/10.2514/1.60553.

[6] Zhou, Y., van Kampen, E.-J., and Chu, Q. P., "Incremental Approximate Dynamic Programming for Nonlinear Adaptive Tracking Control with Partial Observability," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 12, 2018, pp. 2554–2567. https://doi.org/10.2514/1.G003472.

[7] Sutton, R. S., and Barto, A. G., *Reinforcement learning: An introduction*, MIT press, 2018.

[8] Sutton, R. S., Barto, A. G., and Williams, R. J., "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Magazine*, Vol. 12, No. 2, 1992, pp. 19–22. https://doi.org/10.1109/37.126844.

[9] Sun, B., and van Kampen, E.-J., "Launch Vehicle Discrete-Time Optimal Tracking Control using Global Dual Heuristic Programming," *2020 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2020, pp. 162–167. https://doi.org/10.1109/CCTA41146.2020.9206252.

[10] Lewis, F. L., and Vrabie, D., "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, Vol. 9, No. 3, 2009, pp. 32–50. https://doi.org/10.1109/MCAS.2009.933854.

[11] Zhou, Y., van Kampen, E.-J., and Chu, Q. P., "Incremental model based online dual heuristic programming for nonlinear adaptive control," *Control Engineering Practice*, Vol. 73, 2018, pp. 13–25. https://doi.org/10.1016/j.conengprac.2017.12.011.

[12] Heydari, A., and Balakrishnan, S. N., “Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics,” *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 24, No. 1, 2013, pp. 145–157. https://doi.org/10.1109/TNNLS.2012.2227339.

[13] Al-Tamimi, A., Lewis, F. L., and Abu-Khalaf, M., “Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 38, No. 4, 2008, pp. 943–949. https://doi.org/10.1109/TSMCB.2008.926614.

[14] Modares, H., and Lewis, F. L., “Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning,” *Automatica*, Vol. 50, No. 7, 2014, pp. 1780–1792. https://doi.org/10.1016/j.automatica.2014.05.011.

[15] Kiumarsi, B., and Lewis, F. L., “Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems,” *IEEE transactions on neural networks and learning systems*, Vol. 26, No. 1, 2015, pp. 140–151. https://doi.org/10.1109/TNNLS.2014.2358227.

[16] Van Kampen, E.-J., Chu, Q. P., and Mulder, J., “Continuous adaptive critic flight control aided with approximated plant dynamics,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6429. https://doi.org/10.2514/6.2006-6429.

[17] Sun, B., and van Kampen, E.-J., “Incremental Model-Based Heuristic Dynamic Programming with Output Feedback Applied to Aerospace System Identification and Control,” *2020 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2020, pp. 366–371. https://doi.org/10.1109/CCTA41146.2020.9206261.

[18] Li, H., Sun, L., Tan, W., Jia, B., and Liu, X., “Switching Flight Control for Incremental Model-Based Dual Heuristic Dynamic Programming,” *Journal of Guidance, Control, and Dynamics*, 2020, pp. 1–7. https://doi.org/10.2514/1.G004519.

[19] Sun, B., and van Kampen, E.-J., “Incremental Model-Based Global Dual Heuristic Programming for Flight Control,” *IFAC-PapersOnLine*, Vol. 52, No. 29, 2019, pp. 7–12. https://doi.org/10.1016/j.ifacol.2019.12.613.

[20] Lewis, F. L., and Vamvoudakis, K. G., “Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 41, No. 1, 2010, pp. 14–25. https://doi.org/10.1109/TSMCB.2010.2043839.

[21] Tandale, M. D., and Valasek, J., “Adaptive dynamic inversion control with actuator saturation constraints applied to tracking spacecraft maneuvers,” *Journal of the Astronautical Sciences*, Vol. 52, No. 4, 2004, pp. 517–530. https://doi.org/10.1007/BF03546415.

[22] Sonneveldt, L., Chu, Q., and Mulder, J., “Nonlinear flight control design using constrained adaptive backstepping,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007, pp. 322–336. https://doi.org/10.2514/1.25834.

[23] Modares, H., Lewis, F. L., and Naghibi-Sistani, M.-B., “Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems,” *Automatica*, Vol. 50, No. 1, 2014, pp. 193–202. https://doi.org/10.1016/j.automatica.2013.09.043.

[24] Sonneveldt, L., Van Oort, E., Chu, Q., and Mulder, J., “Nonlinear adaptive trajectory control applied to an F-16 model,” *Journal of Guidance, control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 25–39. https://doi.org/doi.org/10.2514/1.38785.

[25] Nguyen, L., Ogburn, M., Gilbert, W., Kibler, K., Brown, P., and Deal, P., “NASA Technical Paper 1538-Simulator Study of Stall/Post-Stall Characteristics of a Fighter Airplane with relaxed Longitudinal Static Stability,” *Tech. rep., NASA*, 1979.