Incremental model-based heuristic dynamic programming with output feedback applied to aerospace system identification and control

Sun, Bo; Van Kampen, Erik Jan

DOI
10.1109/CCTA41146.2020.9206261

Publication date
2020

Document Version
Accepted author manuscript

Published in
CCTA 2020 - 4th IEEE Conference on Control Technology and Applications

Citation (APA)
Sun, B., & Van Kampen, E. J. (2020). Incremental model-based heuristic dynamic programming with output feedback applied to aerospace system identification and control. In *CCTA 2020 - 4th IEEE Conference on Control Technology and Applications* (pp. 366-371). Article 9206261 (CCTA 2020 - 4th IEEE Conference on Control Technology and Applications). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/CCTA41146.2020.9206261

Incremental Model-Based Heuristic Dynamic Programming with Output Feedback Applied to Aerospace System Identification and Control*

Bo Sun[1] and Erik-Jan van Kampen[1]

Abstract— **Sufficient information about system dynamics and inner states is often unavailable to aerospace system controllers, which requires model-free and output feedback control techniques, respectively. This paper presents a novel self-learning control algorithm to deal with these two problems by combining the advantages of heuristic dynamic programming and incremental modeling. The system dynamics is completely unknown and only input/output data can be acquired. The controller identifies the local system models and learns control polices online both by tuning the weights of neural networks. The novel method has been applied to a multi-input multi-output nonlinear satellite attitude tracking control problem. The simulation results demonstrate that, compared with the conventional actor-critic-identifier-based heuristic dynamic programming algorithm with three networks, the proposed adaptive control algorithm improves online identification of the nonlinear system with respect to precision and speed of convergence, while maintaining similar performance compared to the full state feedback situation.**

I. INTRODUCTION

The last hundred years have witnessed the rapid development of aerospace systems, which combines many great technological achievements of humankind. However, as structures of aerospace systems become more diverse and their tasks get more complex, challenges are put up on control systems. One of the most challenging things is the absence of sufficient knowledge of the system dynamics. Especially for aerospace systems, accurate system information can be impossible to acquire [1] due to complexity and nonlinearity.

Recently, adaptive dynamic programming (ADP) has been paid great attention to because of its self-learning property [2] and close relationship with adaptive optimal control [3]. As a class of reinforcement learning (RL) methods, ADP can also methodically adjust the control policy based on observed responses without accurately modeled dynamics of the system or the environment [1]. Because of these advantages, a number of ADP-based methods have been successfully developed for model-free flight controller design [4]–[7].

As an extension of traditional ADP, adaptive critic designs (ACDs) break the shackles in linear methods, and have been successfully applied to adaptive optimal control problems [3]. ACDs normally exploit nonlinear function approximators, such as artificial neural networks (ANNs), to approximate evaluation (critic) and improvement (actor) of the control policy, and consequently they can be applied to problems with more complicated rewards. Based on the information outputted by the critic network, ACDs are generally classified into heuristic dynamic programming (HDP), dual heuristic programming (DHP) and global dual heuristic programming (GDHP) [8]. Among them, HDP, whose critic network directly approximates the cost-to-go, provides a basic structure and is the most popular algorithm. To speed up the learning process and increase the success ratio, an extra structure, usually ANN, is introduced to approximate the system model in [9]–[12]. However, because training ANNs usually needs much effort before the parameters converge, offline training or information of partial system dynamics is often required. To tackle these limitations, incremental technique is introduced to improve the performance of online application of ACDs, which leads to incremental model-based ACDs (IACDs) [1], [13], [14].

Based on full-state feedback (FSF), IACDs have shown impressive advantages over conventional ACDs in various flight control tasks. However, real applications are often more complex and sometimes not only the internal dynamics, but also the information to infer its internal states is unavailable due to structural constraints or internal sensor faults. For example, infrared cameras used as docking sensors can only output the tracking errors between the spacecrafts for navigation, rather than explicit positions [6]. Unexpected faults might happen in delicate sensors, such as air data sensors [15], resulting in inaccurate measurement information. These situations can lead to output feedback (OF) problems.

This paper aims to improve the HDP method by involving the incremental technique and OF. Different from conventional HDP, the proposed method, IHDP with OF (IHDP-OF), employs an extended incremental model to approximate the local dynamics of the original nonlinear system instead of the global model [16], under the assumption that the sampling time is small enough [1]. IHDP-OF combines the methods proposed in [13] and [6], while outperforms them in dealing with partial observability and more complex cost function, respectively.

The remainder of the paper is organized as follows. Section II introduces the incremental model to build a direct mapping between output/input measurements. Section III presents the structure and weights update of the actor and critic networks in the IHDP method. Section IV verifies the proposed IHDP-OF method by applying it to an attitude control task of a satellite and comparing the performance with the HDP with OF (HDP-OF) and the IHDP with FSF (IHDP-FSF). Finally, we discuss the conclusions and future

*The first author's Ph.D. is financially supported by China Scholarship Council, project reference number 201806290007.

[1] B. Sun and E. van Kampen are with the Department of Control and Operations, Delft University of Technology, 2629HS Delft, The Netherlands. Email:{B.Sun-1, E.vanKampen}@tudelft.nl.

research in section V.

II. Incremental Model Identification

A. Incremental Model with Full State Feedback

Aerospace systems are highly nonlinear and their system dynamics can be generally described by:

$$\dot{\mathbf{x}}(t) = f[\mathbf{x}(t), \mathbf{u}(t)] \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the current state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ is the current control vector, and$f[\mathbf{x}(t), \mathbf{u}(t)] \in \mathbb{R}^n$ builds the system dynamics over time.

The linear approximation of the system (1) around time instant t_0 can be achieved by taking the first order Taylor series expansion and omitting higher-order terms, which is described as follows:

$$\begin{aligned}\dot{\mathbf{x}}(t) \approx \dot{\mathbf{x}}(t_0) &+ \mathbf{F}[\mathbf{x}(t_0), \mathbf{u}(t_0)][\mathbf{x}(t) - \mathbf{x}(t_0)] \\ &+ \mathbf{G}[\mathbf{x}(t_0), \mathbf{u}(t_0)][\mathbf{u}(t) - \mathbf{u}(t_0)]\end{aligned} \tag{2}$$

where $\mathbf{F}[\mathbf{x}(t_0), \mathbf{u}(t_0)] = \frac{\partial f[\mathbf{x}(t),\mathbf{u}(t)]}{\partial \mathbf{x}(t)}|_{\mathbf{x}(t_0),\mathbf{u}(t_0)} \in \mathbb{R}^{n \times n}$ denotes the system transition matrix and $\mathbf{G}[\mathbf{x}(t_0), \mathbf{u}(t_0)] = \frac{\partial f[\mathbf{x}(t),\mathbf{u}(t)]}{\partial \mathbf{u}(t)}|_{\mathbf{x}(t_0),\mathbf{u}(t_0)} \in \mathbb{R}^{n \times m}$ denotes the input distribution matrix. Then, an incremental model with FSF can be utilized to represent (2):

$$\Delta\dot{\mathbf{x}}(t) \approx \mathbf{F}[\mathbf{x}(t_0), \mathbf{u}(t_0)]\Delta\mathbf{x}(t) + \mathbf{G}[\mathbf{x}(t_0), \mathbf{u}(t_0)]\Delta\mathbf{u}(t) \tag{3}$$

Assuming the sampling frequency is sufficiently high and system dynamics vary relatively slowly, the system model can be described approximately as the following discrete form [1]:

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

where Δt is the sampling time, $\mathbf{F}_{t-1} = \frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}|_{\mathbf{x}_{t-1},\mathbf{u}_{t-1}} \in \mathbb{R}^{n \times n}$ denotes the discrete system transition matrix and $\mathbf{G}_{t-1} = \frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}|_{\mathbf{x}_{t-1},\mathbf{u}_{t-1}} \in \mathbb{R}^{n \times m}$ denotes the discrete input distribution matrix at time instant $t-1$. In the FSF situation, a recursive least squares (RLS) algorithm [1] can be used to identify matrices $\mathbf{F}_{t-1}$ and $\mathbf{G}_{t-1}$ online. During every update, only the latest data will be used.

B. Incremental Model with Output Feedback

The system output can be described as:

$$\mathbf{y}(t) = h[\mathbf{x}(t)] \tag{5}$$

where $\mathbf{y}(t) \in \mathbb{R}^p$, and $h[\mathbf{x}(t)]$ denotes the output function. Similarly, with a constant, sufficiently small sampling time Δt, the incremental dynamics of (5) can be represented as:

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

where $\mathbf{H}_t = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}_t} \in \mathbb{R}^{p \times n}$ denotes the discrete observation matrix.

However, unlike (1), there is no direct transition between the outputs at different time instants in the physical sense, so the system output cannot be represented only by the input/output measurements at one time step before. Consequently, the information provided in (6) should be utilized, i.e. the system is observable, so that the unavailable internal states can be reconstructed to provide transition information with the adequate observations [5]. Given the measured input/output data over a sufficiently long time horizon N, and N satisfies $N \geq n/p$, the output increment $\Delta\mathbf{y}_{t+1}$ can be presented uniquely as follows:

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

where $\underline{\mathbf{F}}_t \in \mathbb{R}^{p \times Np}$ denotes the extended discrete system transition matrix, $\underline{\mathbf{G}}_t \in \mathbb{R}^{p \times Nm}$ denotes the extended discrete input distribution matrix, $\underline{\mathbf{G}}_{t,11} \in \mathbb{R}^{p \times m}$ and $\underline{\mathbf{G}}_{t,12} \in \mathbb{R}^{p \times (N-1)m}$ denote partitioned matrices from $\underline{G}_t$. $\overline{\Delta\mathbf{y}}_{t,N} = [\Delta\mathbf{y}_t^{\mathrm{T}}, \Delta\mathbf{y}_{t-1}^{\mathrm{T}}, \cdots, \Delta\mathbf{y}_{t-N+1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Np}$ and $\overline{\Delta\mathbf{u}}_{t,N} = [\Delta\mathbf{u}_t^{\mathrm{T}}, \Delta\mathbf{u}_{t-1}^{\mathrm{T}}, \cdots, \Delta\mathbf{u}_{t-N+1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{Nm}$ are the acquired output/input data from N previous steps, respectively.

C. Extended RLS

The RLS algorithm is applied to identify the pending matrices $\underline{\mathbf{F}}_t$ and $\underline{\mathbf{G}}_t$ online. For convenience, (7) can be rewritten in a vector form:

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

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

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

where $\hat{\cdot}$ stands for the estimated or approximated value.

A sliding window technique is employed to store sufficient historic data for online identification [16]. In this situation, there are N sets of parameters in $\underline{\Theta}_t$ waiting for determination, and therefore during each update, $L \geq N$ sets of stored historic data samples should be provided for identification, where L is the width of data window.

The core procedure of the RLS approach [1] can be given as follows:

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

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

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

where $\epsilon_t \in \mathbb{R}^p$ is the prediction error, $\underline{\mathrm{Cov}}_t \in \mathbb{R}^{(p+m)N \times (p+m)N}$ denotes the estimation covariance matrix, which is symmetric and semi-positive definite, and γ_{RLS} is the forgetting factor.

III. Actor-Critic Structure

Both HDP and IHDP are developed based on the actor-critic structure, so in this section we will discuss the details about the implementation of the actor-critic structure as well as the combination with the incremental model derived in Section II.

A. The Critic

HDP is the most widely used ACD method because of its simple form of the critic, which approximates the cost-to-go directly. Among aerospace system control problems, one of the most common tasks is to track a given reference signal. In this paper, the one-step cost function (reward) is designed as:

$$r_t = r(\hat{\mathbf{y}}_t, \mathbf{y}_t^{\text{ref}}) = (\hat{\mathbf{y}}_t - \mathbf{y}_t^{\text{ref}})^{\mathrm{T}} Q_c (\hat{\mathbf{y}}_t - \mathbf{y}_t^{\text{ref}}) = \tilde{\mathbf{y}}_t^{\mathrm{T}} Q_c \tilde{\mathbf{y}}_t \tag{13}$$

where $\hat{\mathbf{y}}$ is the estimated output vector, $\mathbf{y}^{\text{ref}}$ is the reference signal, $\tilde{\mathbf{y}}_t$ denotes the tracking error vector and $Q_c \in \mathbb{R}^{p \times p}$ is a non-negative definite weight matrix. Note that the reward consists of the estimated value $\hat{\mathbf{y}}_t$ instead of the true value $\mathbf{y}_t$ because the controller is directly linked with the incremental model. The cost-to-go $J(\tilde{\mathbf{y}}_t)$ is defined as the cumulative sum of upcoming rewards r_t since the time instant t:

$$J(\tilde{\mathbf{y}}_t) = \sum_{l=t}^{\infty} \gamma^{l-t} r_l \tag{14}$$

where $\gamma \in (0, 1)$ denotes the forgetting factor to decide how the rewards at different time instants are weighted. Because future rewards are unavailable, an ANN, called the critic network, is utilized to approximate the true cost-to-go, whose estimated value is represented as $\hat{J}$, and the temporal difference (TD) technique is applied to tune the ANN weights iteratively. The target of TD technique is to minimize the error between the present and successive estimations, which can be given as:

$$e_c(t) = \hat{J}(\tilde{\mathbf{y}}_t) - r_t - \gamma \hat{J}(\tilde{\mathbf{y}}_{t+1}) \tag{15}$$

For convenience, a overall estimated error function $E_c(t)$ is utilized to eliminate the influence of signs:

$$E_c(t) = \frac{1}{2} e_c^2(t) \tag{16}$$

To minimize $E_c(t)$, so a gradient-descent algorithm with a learning rate $\eta_c > 0$ is applied to update the critic weights:

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) - \eta_c \cdot \frac{\partial E_c(t)}{\partial \mathbf{w}_c(t)} \tag{17}$$

where

$$\frac{\partial E_c(t)}{\partial \mathbf{w}_c(t)} = \frac{\partial E_c(t)}{\partial \hat{J}(\tilde{\mathbf{y}}_t)} \cdot \frac{\partial \hat{J}(\tilde{\mathbf{y}}_t)}{\partial \mathbf{w}_c(t)} = e_c(t) \cdot \frac{\partial \hat{J}(\tilde{\mathbf{y}}_t)}{\partial \mathbf{w}_c(t)} \tag{18}$$

B. The Actor

The aim of the actor network is outputting a control action to minimize the successive approximated cost-to-go $\hat{J}(\tilde{\mathbf{y}}_{t+1})$:

$$\mathbf{u}_t^* = \arg\min_{\mathbf{u}_t} E_a(t+1) \tag{19}$$

in which $E_a(t+1)$ is the overall error function, which can be defined as a quadratic form:

$$E_a(t+1) = \frac{1}{2} \hat{J}^2(\tilde{\mathbf{y}}_{t+1}) \tag{20}$$

The analytical solution is often intractable, so an ANN is also introduced to produce the control action $\mathbf{u}_t$. As shown in Fig. 1, the 3rd back-propagation path indicates the actor weights update direction:

$$\mathbf{w}_a(t+1) = \mathbf{w}_a(t) - \eta_c \cdot \frac{\partial E_a(t+1)}{\partial \mathbf{w}_a(t)} \tag{21}$$

in which $\eta_a > 0$ is the learning rate, and

$$\frac{\partial E_a(t+1)}{\partial \mathbf{w}_a(t)} = \frac{\partial E_a(t+1)}{\partial \hat{J}(\tilde{\mathbf{y}}_{t+1})} \cdot \frac{\partial \hat{J}(\tilde{\mathbf{y}}_{t+1})}{\partial \hat{\mathbf{y}}_{t+1}} \cdot \frac{\partial \hat{\mathbf{y}}_{t+1}}{\partial \mathbf{u}_t} \cdot \frac{\partial \mathbf{u}_t}{\partial \mathbf{w}_a(t)} \tag{22}$$

Substitute (7) into (22) and then (22) can be rewritten as:

$$\frac{\partial E_a(t+1)}{\partial \mathbf{w}_a(t)} = \hat{J}(\tilde{\mathbf{y}}_{t+1}) \cdot \frac{\partial \hat{J}(\tilde{\mathbf{y}}_{t+1})}{\partial \hat{\mathbf{y}}_{t+1}} \cdot \underline{\mathbf{G}}_{t,11} \cdot \frac{\partial \mathbf{u}_t}{\partial \mathbf{w}_a(t)} \tag{23}$$

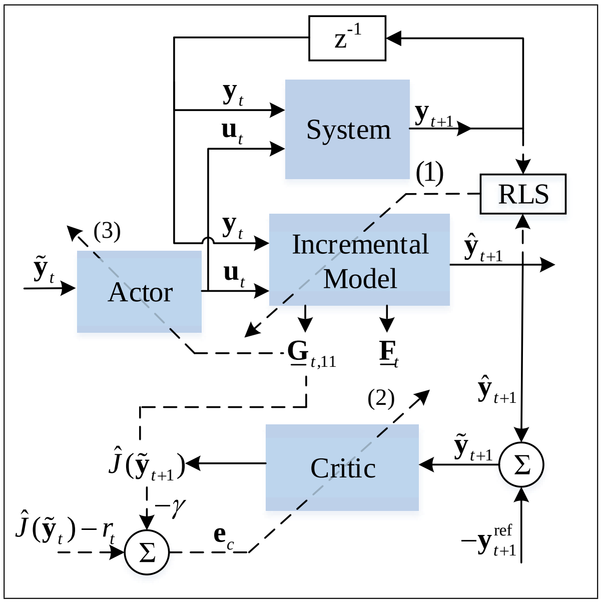


Fig. 1. The architecture of IHDP with output feedback.

IV. Simulation and Discussion

A. Aerospace System Model

Liquid sloshing is a kind of common internal dynamics in the aerospace systems with liquid fuel. Despite a lot of research [6], [17], its accurate model is still extremely difficult to obtain and therefore many model-based control methods cannot be applied to these systems. What is more, due to the limitations of weight or mechanical structure, the information of some internal states may not be available, which results in output feedback problems. Consequently, this paper considers a satellite perturbed by liquid sloshing to evaluate the proposed IHDP-OF method.

As shown in Fig. (2), a attitude control problem of a satellite in 2-dimensional plane is taken into account, where the liquid sloshing is approximately represented by a mechanical system with a pendulum [6], [17]. Subscript p denotes the liquid fuel, and m_p and I_p denote its mass equivalent and moment of inertia, respectively. The other part of the satellite is represented by subscript s. F_s denotes the longitudinal motion thrust that acts on the center of mass of the satellite and is considered to be constant in this paper. The velocity of the satellite is decomposed into the axial component v_x and the transverse component v_z. a and b denote the pendulum length and the distance between the satellite center of mass and the connected point, respectively. ψ denotes the angle between the pendulum and the satellite longitudinal axis, and θ denotes the attitude angle of the satellite. f_s denotes the transverse force and M_s denotes the pitch moment, whose commands are generated by the controller to complete the attitude control task.

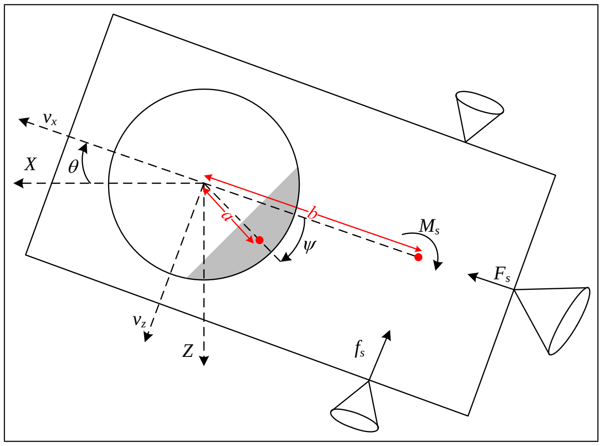


Fig. 2. A approximated model of the satellite model with liquid sloshing using pendulum (adapted from [6]).

Although a prior model is not necessary for IHDP-OF, for verification and validation, the satellite model is presented. The dynamic and kinematic state equations of the satellite with liquid sloshing are as follows [17]:

$$\begin{aligned}&(m_s+m_p)(\dot{v}_x+v_z\dot{\theta})+m_sb\dot{\theta}+\\&m_pa(\ddot{\psi}+\ddot{\theta})\sin(\psi)+m_pa(\dot{\psi}+\dot{\theta})^2\cos(\psi)=F_s\end{aligned}\tag{24}$$

$$\begin{aligned}&(m_s+m_p)(\dot{v}_z-v_x\dot{\theta})+m_sb\ddot{\theta}+\\&m_pa(\ddot{\psi}+\ddot{\theta})\cos(\psi)-m_pa(\dot{\psi}+\dot{\theta})^2\sin(\psi)=f_s\end{aligned}\tag{25}$$

$$m_sb(\dot{v}_z-v_x\dot{\theta})+(I_s+m_sb^2)\ddot{\theta}-\kappa\dot{\psi}=M_s+bf_s\tag{26}$$

$$\begin{aligned}&(m_pa^2+I_p)(\ddot{\psi}+\ddot{\theta})+m_pa[(\dot{v}_x+v_z\dot{\theta})\sin(\psi)+\\&(\dot{v}_z-v_x\dot{\theta})\cos(\psi)]+\kappa\dot{\psi}=0\end{aligned}\tag{27}$$

in which κ denotes the damping constant. According to [6], the rotational variables can be isolated from the translational variables:

$$\begin{aligned}&m_sb[f_s-m_sb\ddot{\theta}-m_pa(\ddot{\psi}+\ddot{\theta})\cos(\psi)+\\&m_pa(\dot{\psi}+\dot{\theta})^2\sin(\psi)]+(m_s+m_p)\cdot\\&[(I_s+m_sb^2)\ddot{\theta}-\kappa\dot{\psi}]=(m_s+m_p)(M_s+bf_s)\end{aligned}\tag{28}$$

$$\begin{aligned}&m_pa\{\sin(\psi)[F_s-m_sb\dot{\theta}-m_pa(\ddot{\psi}+\ddot{\theta})\sin(\psi)]+\\&\cos(\psi)[f_s-m_sb\ddot{\theta}-m_pa(\ddot{\psi}+\ddot{\theta})\cos(\psi)]\}+\\&(m_s+m_p)(m_pa^2+I_p)(\ddot{\psi}+\ddot{\theta})+(m_s+m_p)\kappa\dot{\psi}=0\end{aligned}\tag{29}$$

Equations (28) and (29) approximately describe the rotation motion of the satellite with liquid sloshing without any translational variables and therefore can be separately used for attitude control problem.

B. Implementation Issues

Let $\mathbf{x}=[\theta,\dot{\theta},\psi,\dot{\psi}]^{\mathrm{T}}$, $\mathbf{y}=[\theta,\psi]^{\mathrm{T}}$, and $\mathbf{u}=[f_s,M_s]^{\mathrm{T}}$ denote the state, the output and the control input of the system, respectively. The parameters of satellite dynamics used in the simulations are: $m_s=600\mathrm{kg}$, $I_s=720\mathrm{kg/m}^2$, $m_p=100\mathrm{kg}$, $I_p=90\mathrm{kg/m}^2$, $a=0.3\mathrm{m}$, $b=0.3\mathrm{m}$, $\kappa=1.5(\mathrm{kg}\cdot\mathrm{m}^2)/\mathrm{s}$ and $F_s=500\mathrm{N}$.

Learning rates are initially large numbers and gradually decrease with the weights being tuned. To avoid the weights going to infinity, the weights are bounded between $[-20,20]$. Both the critic and the actor employ a fully connected, single hidden layer ANN. As a balance between approximation precision and computational burden, the neuron number of hidden layer in both networks is 20. The activation function σ in the hidden neurons is set to be a sigmoid function:

$$\sigma(o)=\frac{1-e^{-o}}{1+e^{-o}}\tag{30}$$

For the incremental model, let the initial $\underline{\mathrm{Cov}}_t$ be an identity matrix multiplied by 10^7 and $\gamma_{\mathrm{RLS}}=0.99995$. To keep relatively low computational burden, let $L=N$ based on the derivation in Section II, which can obtain satisfying performance. As a comparison, a model network is used in conventional HDP, and it is configured same to the critic and actor network. The sampling frequency is 100Hz.

The performance of the ANN and the incremental model relies on the sufficient exploration, which is represented by persistent excitation (PE) condition. To better explore the state space and control policy, a predefined probing noise is often added to the control command [10]. This paper introduces the 3211 doublets only at the beginning to excite the fresh modules and introduces an small input disturbance, which is a sum of sinusoidal signals, throughout the control task. For flight control system design, measurement uncertainties are unavoidable in the real world, and thus need to be taken into account. Therefore, for OF problems, zero-mean normal distributed white noises are added on the control inputs towards real systems and the measurements from real systems in the numerical simulations. The standard deviations of the noises are $0.01\mathrm{N}$, $0.005\mathrm{N}\cdot\mathrm{m}$, $0.005°$ and

0.005° for f_s, M_s, θ and ψ, respectively. How to satisfy and evaluate PE condition is still an open problem, while these disturbance and noise can improve the exploration to better achieve PE condition.

C. Simulation Results

The one-step predictions using the incremental model with OF (IM-OF), the incremental model with full state feedback (IM-FSF) and the ANN-based global model with OF (NN-OF) are compared given a determined control policy:

$$\begin{aligned} f_s(t) &= -10\cos(0.3t + 0.5\pi) \\ M_s(t) &= 3\text{sq}(0.2t - 0.5\pi) \end{aligned} \tag{31}$$

where sq(•) is a square wave function.

As shown in Figs. 3 and 4, the one-step state predictions are feasible in this open-loop condition with these three different methods. Nevertheless, it takes more time for the ANN to predict the outputs of the next time step accurately at the beginning, which can cause severe results if the control policy is not stable at the beginning. On the other hand, the incremental model can generate accurate predictions after only a few measurements no shorter than the chosen sliding window. Besides, there are some obvious outliers shown in Fig. 3 even after the NN-OF identifier has converged. This phenomenon happens when the control commands cross the x axis and change their signature, leading to sharp prediction errors. On the contrary, the incremental model-based methods can adapt very fast to this sudden change and no outlier appears.

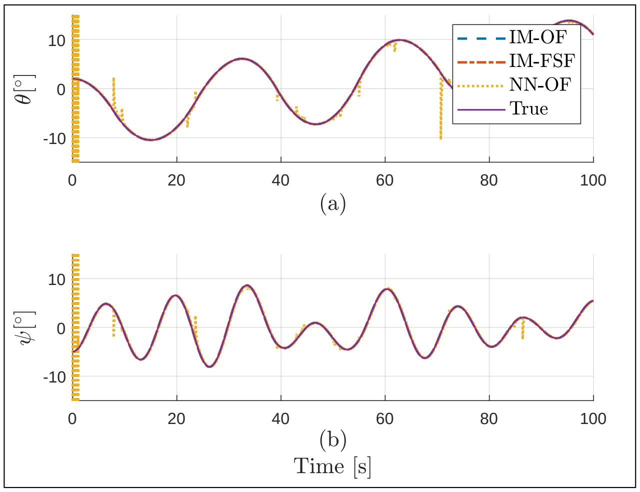


Fig. 3. One-step prediction of the system outputs, θ and ψ.

Table I takes a close look at the probability distributions of the prediction errors after all identifiers have converged, i.e. $t > 5$s. It is clear that the IM-FSF has the smallest means and variances of prediction errors, while the NN-OF has the largest ones. Consequently, it can be concluded that the incremental model outperforms the ANN-based global model in this online identification task.

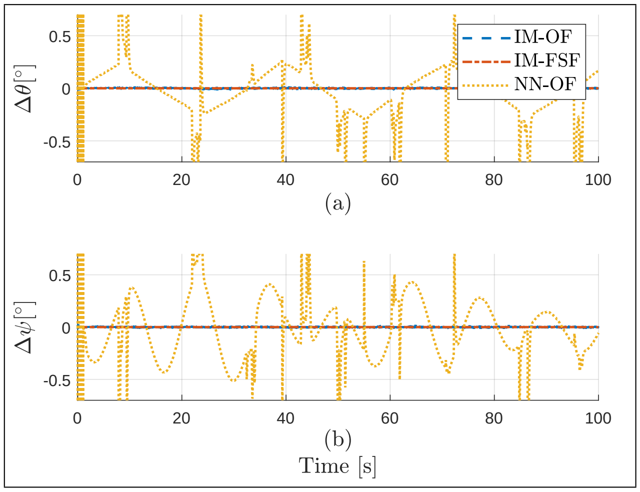


Fig. 4. Prediction errors of the system outputs, θ and ψ .

TABLE I

PROBABILITY DISTRIBUTIONS OF THE PREDICTION ERRORS

Methods	$\Delta\theta[°]$		$\Delta\psi[°]$	
	Mean	***STD***[a]	***Mean***	***STD***
IM-OF	1.5×10^{-4}	5.3×10^{-3}	-1.3×10^{-4}	5.3×10^{-3}
IM-FSF	-2.2×10^{-7}	1.9×10^{-6}	-2.7×10^{-7}	1.3×10^{-6}
NN-OF	-7.1×10^{-2}	6.1×10^{-1}	-1.6×10^{-2}	3.8×10^{-1}

[a] Abbreviation of standard deviation.

The second part applies these methods to a closed-loop control problem. Different from [5] and [6], where offline training is involved, in this paper the controller learns the control policy online. The attitude angle of satellite θ is supposed to track a given sinusoidal reference signal θ^{ref}, whose amplitude is 30° and period is 200πs. Besides, the pendulum angle ψ should be kept as close as possible to zero.

Fig. 5 gives the results of the tracking and stabilizing control tasks. The subfigure (a) shows that all three methods have similar tracking plots of the attitude angle θ. Subfigure (b) and subfigure (c) illustrates the control errors of the attitude angle $\Delta\theta$, and the pendulum angle ψ, respectively. Because of the nonlinearity of the system and coupling between system states, there are avoidable oscillations in control errors using these online self-learning methods. However, it is clear that HDP-OF has the largest control errors while the control errors given by IHDP-FSF are smallest. Apart from the amplitude, the oscillations produced by HDP-OF also have the lowest frequency because the NN-based model adapts slower than the incremental model.

Fig. 6 presents the control commands produced by the IHDP-OF method. Subfigures (a), (b) and (c) illustrate the control commands during the whole period, the initial 3211 excitation signals and a fragment of the detailed control commands, respectively. The control policy is less smooth compared to the final policy in [6], because the control policy is learned totally online without offline training process and measurement uncertainties are considered.

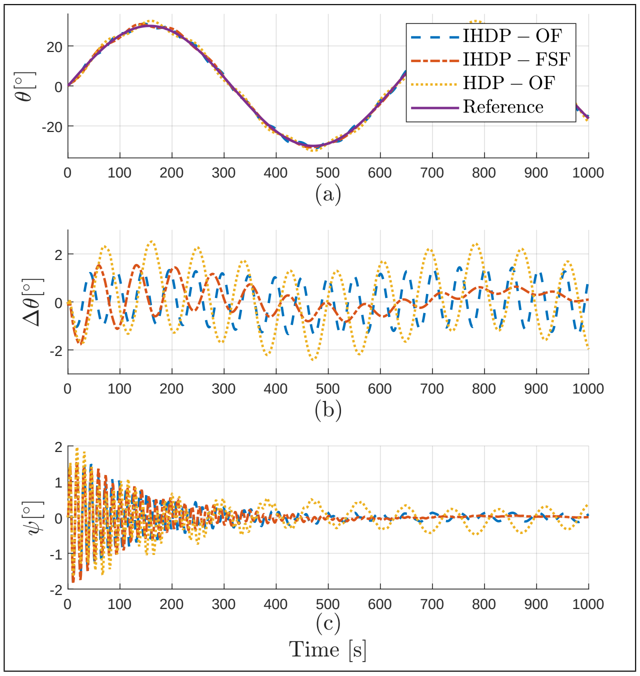
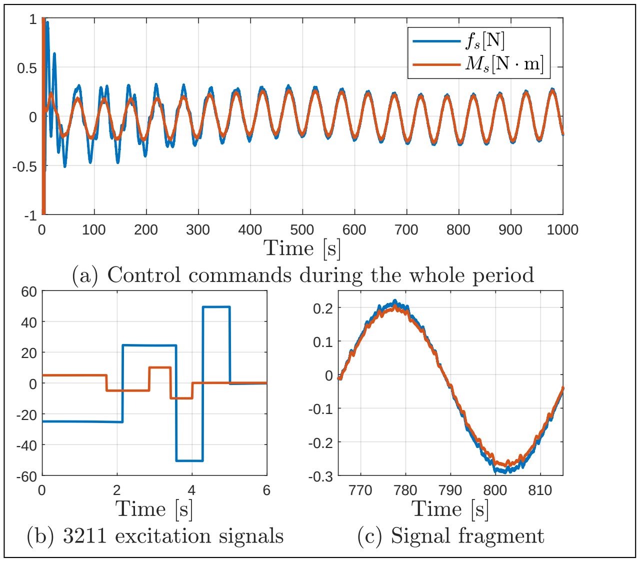


Fig. 5. Tracking performance these three methods.



(a) Control commands during the whole period

(b) 3211 excitation signals

(c) Signal fragment

Fig. 6. Control commands of the IHDP- OF.

V. CONCLUSIONS

This paper takes the output feedback (OF) condition into account, and develops a new method, incremental model-based heuristic dynamic programming with OF (IHDP-OF), to accomplish an adaptive flight controller without prior knowledge of the system dynamics, and measurements referring to inner states. The proposed method expands the original IHDP with full-state feedback (FSF) by building a direct mapping between input and output using the incremental technique with historic data. The simulation results demonstrate that the incremental model accelerates the online identification and improves the precision compared to a global neural network model.

Nevertheless, there still are some problems to be dealt with in realistic applications. The most important point is to satisfy PE condition, without which, the performance of the controller degrades heavily and even divergence can happen. Insufficient exploration can be severer in the OF condition because of the data lost. In this paper, the incremental model does not significantly improve the success ratio compared to the traditional neural network-based model in the OF condition.

REFERENCES

[1] B. Sun and E. van Kampen, "Incremental model-based global dual heuristic programming with explicit analytical calculations applied to flight control," *Engineering Applications of Artificial Intelligence*, vol. 89, p. 103425, 2020.

[2] D. Wang, M. Ha, and J. Qiao, "Self-learning optimal regulation for discrete-time nonlinear systems under event-driven formulation," *IEEE Transactions on Automatic Control*, 2019. Early access.

[3] D. Wang, H. He, and D. Liu, "Adaptive critic nonlinear robust control: A survey," *IEEE transactions on cybernetics*, vol. 47, no. 10, pp. 3429–3451, 2017.

[4] F. A. Yaghmaie, S. Gunnarsson, and F. L. Lewis, "Output regulation of unknown linear systems using average cost reinforcement learning," *Automatica*, vol. 110, p. 108549, 2019.

[5] Y. Zhou, E. van Kampen, and Q. P. Chu, "Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 493–496, 2016.

[6] Y. Zhou, E. van Kampen, and Q. P. Chu, "Incremental approximate dynamic programming for nonlinear adaptive tracking control with partial observability," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 12, pp. 2554–2567, 2018.

[7] H. Modares, F. L. Lewis, and Z.-P. Jiang, "H-∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 10, pp. 2550–2562, 2015.

[8] B. Sun and E.-J. van Kampen, "Incremental model-based global dual heuristic programming for flight control," *IFAC-PapersOnLine*, vol. 52, no. 29, pp. 7–12, 2019.

[9] E. van Kampen, Q. P. Chu, and J. Mulder, "Online adaptive critic flight control using approximated plant dynamics," in *2006 International Conference on Machine Learning and Cybernetics*, pp. 256–261, IEEE, 2006.

[10] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, 2012.

[11] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor–critic–identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, 2013.

[12] F. A. Yaghmaie and D. J. Braun, "Reinforcement learning for a class of continuous-time input constrained optimal control problems," *Automatica*, vol. 99, pp. 221–227, 2019.

[13] Y. Zhou, E. van Kampen, and Q. P. Chu, "Launch vehicle adaptive flight control with incremental model based heuristic dynamic programming," in *Proceedings of the IAC 2017, Adelaide, Australia*, 2017.

[14] Y. Zhou, E. van Kampen, and Q. P. Chu, "Incremental model based online dual heuristic programming for nonlinear adaptive control," *Control Engineering Practice*, vol. 73, pp. 13–25, 2018.

[15] P. Lu, L. Van Eykeren, E. van Kampen, C. C. de Visser, and Q. P. Chu, "Adaptive three-step kalman filter for air data sensor fault detection and diagnosis," *Journal of Guidance, Control, and Dynamics*, no. null, pp. 590–604, 2015.

[16] I. Grondman, M. Vaandrager, L. Busoniu, R. Babuska, and E. Schuitema, "Efficient model learning methods for actor–critic control," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 591–602, 2012.

[17] L. C. G. de Souza and A. G. de Souza, "Satellite attitude control system design considering the fuel slosh dynamics," *Shock and Vibration*, vol. 2014, 2014.