Stability Analysis for Incremental Adaptive Dynamic Programming with Approximation Errors

Li, Yifei; Van Kampen, Erik Jan

DOI
10.1061/JAEEEZ.ASENG-5097

Publication date
2024

Document Version
Final published version

Published in
Journal of Aerospace Engineering

Citation (APA)
Li, Y., & Van Kampen, E. J. (2024). Stability Analysis for Incremental Adaptive Dynamic Programming with Approximation Errors. *Journal of Aerospace Engineering*, *37*(1), Article 04023097. https://doi.org/10.1061/JAEEEZ.ASENG-5097

ASCE

Stability Analysis for Incremental Adaptive Dynamic Programming with Approximation Errors

Yifei Li[1] and Erik-Jan van Kampen, Ph.D.[2]

Abstract: This paper provides a convergence and stability analysis of the incremental value iteration algorithm under the influence of various errors. Incremental control is firstly used to linearize the continuous-time nonlinear system, recursive least squares (RLS) identification is then introduced to identify the incremental model online. Based on the incremental model, the value iteration algorithm is used to design an optimal adaptive controller, with an analytical optimal control law. Moreover, the convergence of the developed incremental value iteration algorithm is proved. The stability of the controller is analyzed using Lyapunov stability theory. Finally, a flight control simulation verifies the robustness of the controller to various initial conditions, as well as adaptation to actuator faults. **DOI: 10.1061/JAEEEZ.ASENG-5097.**

Introduction

Reinforcement learning (RL) is a field of machine learning (ML) that is best characterized by interaction with the environment (Sutton and Barto 2014). In the control field, RL has been widely used to solve optimal control problems of a class of systems with unknown dynamics, namely approximate dynamic programming (ADP) (Bertsekas 2019; Jiang and Jiang 2017; Lewis and Liu 2013; Sutton et al. 1992; Sharma and York 2018), which was first proposed by Werbos (1977). Accompanying the continuous and high-dimensional control spaces is the exponential growth of computational complexity, known as the curse of dimensionality (Werbos 1977; Powell 1977). The curse of dimensionality is mitigated by utilizing parameterized approximators (Haykin 2009), such as artificial neural networks (ANNs), polynomial functions, and quadratic functions.

Since the 2000s, ADP methods have been successfully applied to aerospace systems, such as morphing aircraft (Wang et al. 2019a), satellites (Zhou et al. 2020), and continuum robots (Jiang et al. 2022). To deal with the high nonlinearity and uncertainty of aerospace systems, ADP methods often need an ANN to approximate the system's dynamical model. This model network needs to be trained offline using a representative simulation model before it is applied online, causing increased computation load. Moreover, the offline training relies on a simulation model with high fidelity, which is difficult to obtain, resulting in a reality gap between offline training and online implementation.

Incremental control techniques can deal with the control of nonlinear systems with uncertainties by establishing a local incremental model, under the assumption that the system is sampled at a sufficiently high frequency. Many nonlinear control methods are combined with incremental control, such as incremental nonlinear dynamic inversion (INDI) (Sieberling et al. 2010; Wang et al. 2019c; Liu et al. 2022), incremental backstepping (IBS) (Wang et al. 2018; Acquatella et al. 2013), incremental sliding mode control (ISMC) (Wang and Sun 2022; Wang et al. 2021), and incremental adaptive dynamic programming (IADP) (Zhou et al. 2015, 2016, 2018). The method used in incremental control to linearize the nonlinear model involves using a Taylor expansion with respect to local states and neglecting the higher-order terms. As a result, a first-order approximation of state derivative is obtained. However, neglecting higher-order terms of a Taylor series results in a model approximation error. For INDI methods, Wang et al. (2019c) analyzed the insensitivity to the residual cancellation error (including higher-order terms). For IADP methods, Zhou et al. (2016, 2018) and Sun and Kampen (2021) assume that the higher-order terms are negligible for sufficiently high sampling frequency. The reason to use an incremental model is that it uses a linearized model of the original nonlinear system. Using recursive least squares (RLS) identification to identify this linearized model can be faster than identifying a nonlinear model. As a result, the IADP algorithms become more suitable for online implementation.

The stability analysis of ADP is well developed using Lyapunov stability theory in the literature (Liu and Wei 2014; Tamimi et al. 2008; Heydari 2014, 2015, 2018). Balakrishnan et al. (2018) provide an overview of ADP-based feedback controller stability analysis. ADP can be classified into two categories, i.e., value iteration (VI) and policy iteration (PI). The convergence proof of PI can be seen in previous studies (Liu and Wei 2014; Guo et al. 2018). For VI, the convergence proof was first developed for general nonlinear control-affine systems by Tamimi et al. (2008). Heydari (2014) present the convergence of VI-based ADP algorithms, including heuristic dynamic programming (HDP) and dual heuristic programming (DHP) to solve infinite-horizon optimal control problems. HDP algorithms approximate the cost function, while DHP algorithms approximate the gradient of cost function. Heydari (2015) considered the approximation error at each iteration to prove the convergence and stability of VI. Later, Heydari (2018) presented a general conclusion of value iteration stability, considering the effects of value function and control policy approximation errors. However, the aforementioned theoretical results are based on perfect knowledge of the system dynamics. For IADP algorithms, the nonlinear system dynamics are approximated by an incremental linear model, which introduces model approximation error and affects the convergence to the optimal value.

[1]Dept. of Control and Operation, Delft Univ. of Technology, Delft, South Holland 2629 HS, Netherlands (corresponding author). Email: Y.Li-34@tudelft.nl

[2]Dept. of Control and Operation, Delft Univ. of Technology, Delft, South Holland 2629 HS, Netherlands. ORCID: https://orcid.org/0000-0002-5593-4471. Email: E.vanKampen@tudelft.nl

Note. This manuscript was submitted on February 9, 2023; approved on August 3, 2023; published online on October 9, 2023. Discussion period open until March 9, 2024; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Aerospace Engineering***, © ASCE, ISSN 0893-1321.

Motivated by the above analysis, the contributions of this paper are summarized as follows:

- In the stability analysis of incremental value iteration algorithm, we consider the approximation error of the incremental model. This consideration is necessary for practical control problems, because the nonlinear system models are usually not perfectly identified.
- A sufficient and necessary condition for asymptotic stability of incremental value iteration is provided. This condition describes how the approximation error at each iteration affects the iteration tolerance and system stability.

This study analyzes the stability of the incremental value iteration algorithm, considering the incremental model approximation error. To the authors' best knowledge, the effect of incremental model approximation error on the stability of approximate value iteration has not been discussed in the literature.

The remainder of this paper is structured as follows. The second section presents the framework of the incremental value iteration algorithm for optimal tracking problems. The third section provides the error analysis of the approximate value iteration. The fourth section presents the convergence analysis of the incremental value iteration under various approximation errors, as well as the stability analysis of nonlinear optimal control using the incremental value iteration algorithm. The fifth section provides the simulation results of a flight control problem. In the final section, concluding remarks are given.

Incremental Value Iteration for Optimal Tracking Control

This section develops optimal tracking control using the incremental value iteration algorithm. The incremental model is first developed using Taylor expansion, leading to a linear system model. Then, the RLS algorithm is introduced to identify the system model parameters. Finally, the value iteration algorithm is used to design a tracking controller for the incremental model.

Incremental Model

The purpose of the incremental control method is to approximate a nonlinear system with a time-varying linear system model at each discrete time step (Sieberling et al. 2010). Although most physical systems are continuous, the control of physical systems is usually considered in a discrete-time domain. Incremental control assumes high-frequency sampling to reduce the approximation error. In practical applications, the sampling time is constrained by hardware. In previous studies (Liu et al. 2022; Sieberling et al. 2010; Zhou et al. 2015, 2016, 2018), incremental control was applied to the control of the nonlinear system with unknown dynamics and uncertainties.

The discrete-time nonlinear system subjected to the control input is expressed as

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \qquad k \in \mathbb{N} \tag{1}$$

where $f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a smooth nonlinear function associated with state vector $\boldsymbol{x}_k$ and input vector $\boldsymbol{u}_k$. n, m are positive integers denoting the dimensions of the state and control spaces. k represents the discrete-time index. $\mathbb{N}$ represents the set of nonnegative integers.

Using the Taylor expansion of Eq. (1) at state $\boldsymbol{x}_k$, the following as

$$\begin{aligned}\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{F}_{k-1}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) + \boldsymbol{G}_{k-1}(\boldsymbol{u}_k - \boldsymbol{u}_{k-1}) \\ + O[(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})^2, (\boldsymbol{u}_k - \boldsymbol{u}_{k-1})^2]\end{aligned} \tag{2}$$

where $\boldsymbol{F}_{k-1} = \partial f(\boldsymbol{x}, \boldsymbol{u})/\partial \boldsymbol{x}|_{\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}} \in \mathbb{R}^{n\times n}$ is the system transition matrix, and $\boldsymbol{G}_{k-1} = \partial f(\boldsymbol{x}, \boldsymbol{u})/\partial \boldsymbol{u}|_{\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}} \in \mathbb{R}^{n\times m}$ is the input distribution matrix at time step $k-1$ for discretized systems. $O[(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})^2, (\boldsymbol{u}_k - \boldsymbol{u}_{k-1})^2]$ are the higher-order terms of the Taylor expansion series.

Eq. (2) can be rewritten in an incremental formulation as

$$\Delta \boldsymbol{x}_{k+1} = \boldsymbol{F}_{k-1}\Delta \boldsymbol{x}_k + \boldsymbol{G}_{k-1}\Delta \boldsymbol{u}_k + O[(\Delta \boldsymbol{x}_k)^2, (\Delta \boldsymbol{u}_k)^2)] \tag{3}$$

where $\Delta \boldsymbol{x}_{k+1} = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ is the state increment at time index $k+1$ with respect to k. $\Delta \boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$, $\Delta \boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$ are the state and control increments at time index k with respect to $k-1$.

The nonlinear system can be represented by this time-varying incremental model. This linear model needs to be available online to provide the model information to the incremental value iteration algorithm instead of using a global nonlinear system model. With high-frequency sample data and a relatively slow-varying system, the time-varying matrices $\boldsymbol{F}_{k-1}$ and $\boldsymbol{G}_{k-1}$ can be identified online using the RLS method (Isermann and Munchhof 2011).

Recursive Least-Squares Identification

RLS is an online algorithm, which reduces the computational effort and provides an update of the parameter estimates at each sample step. Compared to nonrecursive identification methods, recursive methods do not store the previous measured data (Isermann and Munchhof 2011).

To present the RLS algorithm, the augmented system state is defined as

$$\boldsymbol{X}_k = \begin{bmatrix}\Delta \boldsymbol{x}_k \\ \Delta \boldsymbol{u}_k\end{bmatrix} \tag{4}$$

The augmented system matrices are defined as

$$\hat{\boldsymbol{\Theta}}_{k-1} = [\hat{\boldsymbol{F}}_{k-1}\hat{\boldsymbol{G}}_{k-1}]^T \tag{5}$$

Conduct a one-step prediction of augmented state $\Delta \hat{\boldsymbol{x}}_{k+1}^T$ as

$$\Delta \hat{\boldsymbol{x}}_{k+1}^T = \boldsymbol{X}_k^T \hat{\boldsymbol{\Theta}}_{k-1} \tag{6}$$

The error $\boldsymbol{\varepsilon}_k$ between $\Delta \boldsymbol{x}_{k+1}^T$ and $\Delta \hat{\boldsymbol{x}}_{k+1}^T$ is defined as

$$\boldsymbol{\varepsilon}_k = \Delta \boldsymbol{x}_{k+1}^T - \Delta \hat{\boldsymbol{x}}_{k+1}^T \tag{7}$$

The estimate of the augmented system matrix $\hat{\boldsymbol{\Theta}}_{k-1}$ is updated as

$$\hat{\boldsymbol{\Theta}}_k = \hat{\boldsymbol{\Theta}}_{k-1} + \frac{\Lambda_{k-1}\boldsymbol{X}_k}{\kappa + \boldsymbol{X}_k^T \Lambda_{k-1}\boldsymbol{X}_k}\boldsymbol{\varepsilon}_k \tag{8}$$

where Λ_{k-1} is the equal weighted estimation of the covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\Theta}}_k - \hat{\boldsymbol{\Theta}}_{k-1})$, which describes the confidence of the estimated $\hat{\Theta}_k$. Λ_{k-1} is updated by

$$\Lambda_k = \frac{1}{\kappa}\left[\Lambda_{k-1} - \frac{\Lambda_{k-1}\boldsymbol{X}_k\boldsymbol{X}_k^T\Lambda_{k-1}}{\kappa + \boldsymbol{X}_k^T\Lambda_{k-1}\boldsymbol{X}_k}\right] \tag{9}$$

where $\kappa \in (0, 1)$ is the forgetting factor, which weights older measurements exponentially. The value of κ provides a balance between the performance of noise rejection and time-varying parameter estimation. When $\kappa \to 1$, the RLS algorithm becomes equally weighted and behaves better at noise rejection; when $\kappa \to 0$, the RLS algorithm shows adaptation to new measurements, and thus adapts to time-varying parameters. For a satisfying performance in practice, κ is suggested to be assigned as $0.9 < \kappa < 0.995$.

It has been proved in Isermann and Munchhof (2011) that RLS identification is a bias-free method, under the condition that the output has been affected by a white Gaussian noise. Therefore, the estimation error is bounded.

Assumption 1. The estimation errors using RLS are bounded as $\|\boldsymbol{F}_k - \hat{\boldsymbol{F}}_k\| \leq \Delta_{\boldsymbol{F}_k}$, $\|\boldsymbol{G}_k - \hat{\boldsymbol{G}}_k\| \leq \Delta_{\boldsymbol{G}_k}$, $k \in \mathbb{N}$. $\Delta_{\boldsymbol{F}_k}$, $\Delta_{\boldsymbol{G}_k}$ are constant bounds.

According to assumption 1, Eq. (3) can be rewritten as

$$\begin{aligned}\Delta \boldsymbol{x}_{k+1} = {} & \hat{\boldsymbol{F}}_{k-1}\Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}_{k-1}\Delta \boldsymbol{u}_k + (\boldsymbol{F}_{k-1} - \hat{\boldsymbol{F}}_{k-1})\Delta \boldsymbol{x}_k \\ & + (\boldsymbol{G}_{k-1} - \hat{\boldsymbol{G}}_{k-1})\Delta \boldsymbol{u}_k + O(\Delta \boldsymbol{x}_k^2, \Delta \boldsymbol{u}_k^2)\end{aligned} \tag{10}$$

where the estimated system matrix $\hat{\boldsymbol{F}}_{k-1}$ and control matrix $\hat{\boldsymbol{G}}_{k-1}$ are used to represent the incremental model in practice.

Eq. (10) is an exact representation, where the higher-order terms of Taylor expansion and the RLS estimation error are presented as $O(\Delta \boldsymbol{x}_k^2, \Delta \boldsymbol{u}_k^2)$ and $(\boldsymbol{F}_{k-1} - \hat{\boldsymbol{F}}_{k-1})\Delta \boldsymbol{x}_k$, $(\boldsymbol{G}_{k-1} - \hat{\boldsymbol{G}}_{k-1})\Delta \boldsymbol{u}_k$, respectively. The incremental model used in previous studies (Zhou et al. 2015, 2016, 2018; Sun and Kampen 2021) is in fact an approximation of the nonlinear system and keeps the first-order part of Taylor expansion, but the approximation error is not discussed.

The incremental model is simplified as

$$\Delta \boldsymbol{x}_{k+1} = \hat{\boldsymbol{F}}_{k-1}\Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}_{k-1}\Delta \boldsymbol{u}_k + \Delta_{\text{IME}} \tag{11}$$

where $\Delta_{\text{IME}} = (\boldsymbol{F}_{k-1} - \hat{\boldsymbol{F}}_{k-1})\Delta \boldsymbol{x}_k + (\boldsymbol{G}_{k-1} - \hat{\boldsymbol{G}}_{k-1})\Delta \boldsymbol{u}_k + O(\Delta \boldsymbol{x}_k^2, \Delta \boldsymbol{u}_k^2)$ is the total error of using incremental model approximation and RLS estimation.

Incremental Value Iteration–Based Optimal Control Algorithm

The incremental value iteration algorithm assumes a time-varying linear model and can be used for nonlinear tracking control problems. In addition, this method optimizes the control increment by minimizing a cost function. Incremental value iteration does not assume the principle of time-scale separation. The principle of time-scale separation in flight control systems means that the inner control loop that is used to stabilize attitude and angular rates, and the outer loop that tracks vehicle position, can be treated separately because the attitude dynamics are faster than the translational dynamics.

The utility function is defined as follows:

$$r(\boldsymbol{x}_k, \boldsymbol{u}_k) = (\boldsymbol{x}_k - \boldsymbol{x}_k^{\text{ref}})^T Q(\boldsymbol{x}_k - \boldsymbol{x}_k^{\text{ref}}) + \boldsymbol{u}_k^T R \boldsymbol{u}_k \tag{12}$$

where Q and R are positive definite matrices, and $\boldsymbol{x}_k^{\text{ref}}$ is the reference signal for the system state. The cost function is the cumulative sum of utility functions starting from state $\boldsymbol{x}_k$ driven by a policy

$$V(k) = \sum_{l=k}^{\infty} \gamma^{l-k} r(\boldsymbol{x}_l, \boldsymbol{u}_l) \tag{13}$$

where the discount factor $\gamma \in (0, 1)$ represents the importance of future utility functions.

The Bellman equation (Sutton and Barto 2014) is then derived as

$$V(k) = r(\boldsymbol{x}_k, \boldsymbol{u}_k) + \gamma V(k+1) \tag{14}$$

Remark 1. The forgetting factor $\gamma < 1$ ensures that the future discounted utility functions converge to 0 as $l \to \infty$, which is a finite-horizon optimal control problem. Intuitively, the future utility functions do not have the same importance as near-horizon utility functions. When $\gamma = 1$, as in typical infinite-horizon optimal control problems, the bound of $V(k)$ goes to infinity and the stability result fails. The value of γ affects the convergence rate of value iteration. The smaller γ is, the faster the value iteration algorithm converges.

The reconstruction of exact cost functions $V(k)$, $V(k+1)$ in the right-hand side of Eq. (14) is one challenge of approximate value iteration. For this purpose, a parameterized approximator is usually used, but approximation error is inevitably introduced. More details on approximation error are discussed in the next section. This paper adopts a quadratic cost function to approximate the exact cost function

$$\hat{V}(k) = \boldsymbol{e}_k^T P \boldsymbol{e}_k \tag{15}$$

The reason to use quadratic approximator is that the cost function is assumed to be a quadratic form. Specifically, the cost function is the cumulative sum of future utility functions, which has infinite terms and is difficult to calculate.

According to Eq. (15), one has $\hat{V}(k+1) = \boldsymbol{e}_{k+1}^T P \boldsymbol{e}_{k+1}$. However, $\boldsymbol{e}_{k+1}$ is not available at time index k, so one has to predict $\boldsymbol{e}_{k+1}$ using the constructed incremental model in Eq. (11). To this end, the exact $\boldsymbol{e}_{k+1}$ is derived as

$$\begin{aligned}\boldsymbol{e}_{k+1} &= \boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^{\text{ref}} \\ &= \boldsymbol{x}_k + \hat{\boldsymbol{F}}_{k+1}\Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}_{k-1}\Delta \boldsymbol{u}_k + \Delta_{\text{IME}} - \boldsymbol{x}_k^{\text{ref}} - \Delta \boldsymbol{x}_{k+1}^{\text{ref}} \\ &= (\boldsymbol{x}_k - \boldsymbol{x}_k^{\text{ref}}) + \hat{\boldsymbol{F}}_{k-1}\Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}\Delta \boldsymbol{u}_k + \Delta_{\text{IME}} - \Delta \boldsymbol{x}_{k+1}^{\text{ref}} \\ &= \boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1}\Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}_{k-1}\Delta \boldsymbol{u}_k + \Delta_{\text{IME}} - \Delta \boldsymbol{x}_{k+1}^{\text{ref}}\end{aligned} \tag{16}$$

Omitting the incremental model approximation error Δ_{IME} and the increment of reference signal $\Delta \boldsymbol{x}_{k+1}^{\text{ref}}$, the prediction of $\boldsymbol{e}_{k+1}$ is calculated as

$$\hat{\boldsymbol{e}}_{k+1} = \boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1}\Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}_{k-1}\Delta \boldsymbol{u}_k \tag{17}$$

Then, the modified approximated cost function, denoted as $\hat{V}'(k+1)$, is defined as

$$\hat{V}'(k+1) = \hat{\boldsymbol{e}}_{k+1}^T P \hat{\boldsymbol{e}}_{k+1} \tag{18}$$

Using $\hat{V}'(k+1)$ in Eq. (18) to construct $V(k+1)$ in l (14), one has

$$\begin{aligned}\hat{V}(k) &\approx r(\boldsymbol{x}_k, \boldsymbol{u}_k) + \gamma \hat{V}'(k+1) \\ &= \boldsymbol{e}_k^T Q \boldsymbol{e}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k + \gamma \hat{\boldsymbol{e}}_{k+1}^T P \hat{\boldsymbol{e}}_{k+1} \\ &= \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k)^T R (\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k) \\ &\quad + \gamma (\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1}\Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}_{k-1}\Delta \boldsymbol{u}_k)^T \\ &\quad \times P (\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1}\Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}_{k-1}\Delta \boldsymbol{u}_k)\end{aligned} \tag{19}$$

Remark 2. In Eq. (19), the quadratic function $\hat{V}'(k+1) = \hat{\boldsymbol{e}}_{k+1}^T P \hat{\boldsymbol{e}}_{k+1}$ is used to construct the exact cost function $V(k+1)$. This approximation can be divided into two parts: the first part is using $\boldsymbol{e}_{k+1}^T P \boldsymbol{e}_{k+1}$ to approximate $V(k+1) = \sum_{l=k+1}^{\infty} \gamma^{l-k} r(\boldsymbol{x}_l, \boldsymbol{u}_l)$,; the second part is using $\hat{\boldsymbol{e}}_{k+1}$ to approximate $\boldsymbol{e}_{k+1}$.

Remark 3. From Eq. (19), one can conclude that the approximated cost function $\hat{V}(k)$ is a function of state variables ($\boldsymbol{e}_k$, $\Delta \boldsymbol{x}_k$, $\boldsymbol{u}_{k-1}$, $\Delta \boldsymbol{u}_k$), estimated incremental model matrices ($\hat{\boldsymbol{F}}_{k-1}$, $\hat{\boldsymbol{G}}_{k-1}$), and cost function matrices Q, R.

The optimal approximated cost function $\hat{V}^*(k)$ is defined as

$$\hat{V}^*(k) = \min_{\Delta \boldsymbol{u}_k}[\boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k)^T R(\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k) + \gamma \hat{V}^*(k+1)] \tag{20}$$

The optimal control increment $\Delta \boldsymbol{u}^*(k)$ is given as

$$\Delta \boldsymbol{u}^*(k) = \operatorname*{argmin}_{\Delta \boldsymbol{u}_k}[\boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k)^T R(\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k) + \gamma \hat{V}^*(k+1)] \tag{21}$$

The optimal control increment can be solved by taking the first derivative of the right-hand side of Eq. (21). However, the optimal solution is not exactly the solution of $\partial \hat{V}(k)/\partial \Delta u_k = 0$, but the closest analytical solution to it

$$\begin{aligned}\frac{\partial \hat{V}(k)}{\partial \Delta \boldsymbol{u}_k} &\approx 2R(\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k) + 2\gamma \hat{\boldsymbol{G}}_{k-1}^T P(\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1} \Delta \boldsymbol{x}_k + \hat{\boldsymbol{G}}_{k-1} \Delta \boldsymbol{u}_k) \\ &= 2(R + \gamma \hat{\boldsymbol{G}}_{k-1}^T P \hat{\boldsymbol{G}}_{k-1}) \Delta \boldsymbol{u}_k \\ &\quad + 2[R\boldsymbol{u}_{k-1} + \gamma \hat{\boldsymbol{G}}_{k-1}^T P(\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1} \Delta \boldsymbol{x}_k)] = 0\end{aligned} \tag{22}$$

From Eq. (22), one has

$$\begin{aligned}&2(R + \gamma \hat{\boldsymbol{G}}_{k-1}^T P \hat{\boldsymbol{G}}_{k-1}) \Delta \boldsymbol{u}_k + 2[R\boldsymbol{u}_{k-1} + \gamma \hat{\boldsymbol{G}}_{k-1}^T P(\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1} \Delta \boldsymbol{x}_k)] = 0 \\ &(R + \gamma \hat{\boldsymbol{G}}_{k-1}^T P \hat{\boldsymbol{G}}_{k-1}) \Delta \boldsymbol{u}_k = -[R\boldsymbol{u}_{k-1} + \gamma \hat{\boldsymbol{G}}_{k-1}^T P(\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1} \Delta \boldsymbol{x}_k)] \\ &\Delta \boldsymbol{u}_k = -(R + \gamma \hat{\boldsymbol{G}}_{k-1}^T P \hat{\boldsymbol{G}}_{k-1})^{-1}[R\boldsymbol{u}_{k-1} + \gamma \hat{\boldsymbol{G}}_{k-1}^T P(\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1} \Delta \boldsymbol{x}_k)]\end{aligned} \tag{23}$$

Therefore, the optimal incremental control $\Delta \boldsymbol{u}_k^*$ is given as

$$\Delta \boldsymbol{u}_k^* = -(R + \gamma \hat{\boldsymbol{G}}_{k-1}^T P \hat{\boldsymbol{G}}_{k-1})^{-1}[R\boldsymbol{u}_{k-1} + \gamma \hat{\boldsymbol{G}}_{k-1}^T P(\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1} \Delta \boldsymbol{x}_k)] \tag{24}$$

The optimal control derived in Eq. (24) depends on the knowledge of kernel matrix P, which is calculated by solving the Bellman equation [Eq. (19)]. Eq. (19) is unsolvable directly because P is implicitly contained in $\Delta \boldsymbol{u}_k$. Therefore, the following iterative computation is used to obtain an approximated solution:

1. Policy Improvement. The policy improves for the current kernel matrix P^i as

$$\begin{aligned}\Delta \boldsymbol{u}_k^i = &-(R + \gamma \hat{\boldsymbol{G}}_{k-1}^T P^i \hat{\boldsymbol{G}}_{k-1})^{-1} \\ &\times [R\boldsymbol{u}_{k-1} + \gamma \hat{\boldsymbol{G}}_{k-1}^T P^i(\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1} \Delta \boldsymbol{x}_k)]\end{aligned} \tag{25}$$

$$\boldsymbol{u}_k^i = \boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k^i \tag{26}$$

2. Policy Evaluation. The kernel matrix series $\{P^0, P^1, \ldots, P^{i_{\max}}\}$ in approximate cost function $\hat{V}(k)$ is calculated recursively, using the equation derived in Eq. (19) as

$$\boldsymbol{e}_k^T P^{i+1} \boldsymbol{e}_k \approx \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1} \tag{27}$$

Remark 4. The optimality of the control policy in Eq. (25) is partially achieved by adopting a changing kernel matrix P^i. Using the Bellman equation in Eq. (27) improves the precision of value function approximation, resulting into an improved matrix P^{i+1}, which makes $\hat{V}(k)$ closer to $V(k)$. Therefore, the control derived by $\hat{V}(k)$ in Eq. (22) is closer to the optimal control derived by $V(k)$.

The incremental value iteration algorithm is summarized in algorithm 1.

Algorithm 1. Incremental Value Iteration Algorithm

Required Input:

state $\boldsymbol{x}_k$, $\boldsymbol{x}_{k+1}$, state reference $\boldsymbol{x}_k^{\text{ref}}$, $\boldsymbol{x}_{k+1}^{\text{ref}}$

Initialization:

Choose maximum iteration number $i_{\max}$

Choose forgetting factor γ, cost function matrices Q, R

Choose initial kernel matrix P^0, initial control u_0

Choose initial system matrices $\hat{\Theta}_0 = [\hat{\boldsymbol{F}}_0, \hat{\boldsymbol{G}}_0]^T$, initial covariance matrix Λ_0

RLS Identification:

1: $\Delta \hat{\boldsymbol{x}}_{k+1}^T = X_k^T \hat{\Theta}_{k-1}$

2: $\Delta \boldsymbol{x}_{k+1} = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$

3: $\boldsymbol{\varepsilon}_k = \Delta \boldsymbol{x}_{k+1}^T - \Delta \hat{\boldsymbol{x}}_{k+1}^T$

4: $\hat{\Theta}_k = \hat{\Theta}_{k-1} + \dfrac{\Lambda_{k-1} \boldsymbol{X}_k}{\kappa + \boldsymbol{X}_k^T \Lambda_{k-1} \boldsymbol{X}_k} \boldsymbol{\varepsilon}_k$

5: $\Lambda_k = \dfrac{1}{\kappa}\left[\Lambda_{k-1} - \dfrac{\Lambda_{k-1} \boldsymbol{X}_k \boldsymbol{X}_k^T \Lambda_{k-1}}{\kappa + \boldsymbol{X}_k^T \Lambda_{k-1} \boldsymbol{X}_k}\right]$

Value Iteration:

for $i = 0$ to $i_{\max}$

1: $\boldsymbol{e}_k \leftarrow \boldsymbol{x}_k - \boldsymbol{x}_k^{\text{ref}}$

2: $\boldsymbol{e}_{k+1} \leftarrow \boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^{\text{ref}}$

3: $\Delta \boldsymbol{u}_k^i \leftarrow -\left(R + \gamma \hat{\boldsymbol{G}}_{k-1}^T P^i \hat{\boldsymbol{G}}_{k-1}\right)^{-1} \times \left[R\boldsymbol{u}_{k-1} + \gamma \hat{\boldsymbol{G}}_{k-1}^T P^i \left(\boldsymbol{e}_k + \hat{\boldsymbol{F}}_{k-1} \Delta \boldsymbol{x}_k\right)\right]$

4: $\boldsymbol{u}_k^i \leftarrow \boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k^i$

5: Solve $\boldsymbol{e}_k^T P^{i+1} \boldsymbol{e}_k = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1}$, obtain P^{i+1}

end for

Approximation Error Analysis

The exact reconstruction of the cost function in exact value iteration is usually impossible except for in simple problems, because the cost function is defined as a sum of future utility functions, as in Eq. (13). Approximating the cost function with a quadratic function inevitably introduces approximation errors in every iteration. This section formulates the approximation errors in every iteration, which would affect the stability and convergence of incremental value iteration.

To explore the effect of approximation error, rewrite the policy evaluation in Eq. (27) as

$$\boldsymbol{e}_k^T P^{i+1} \boldsymbol{e}_k = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1} + \epsilon_i(\boldsymbol{x}_k), \quad \forall\ i \in [0, i_{\max}] \tag{28}$$

For $i = 0, 1, \ldots, i_{\max}$, Eq. (28) is rewritten as

$$\begin{aligned}&\boldsymbol{e}_k^T P^1 \boldsymbol{e}_k = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^0 \hat{\boldsymbol{e}}_{k+1} + \epsilon_0(\boldsymbol{x}_k) \\ &\boldsymbol{e}_k^T P^2 \boldsymbol{e}_k = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^1 \hat{\boldsymbol{e}}_{k+1} + \epsilon_1(\boldsymbol{x}_k) \\ &\qquad \vdots \\ &\boldsymbol{e}_k^T P^{i+1} \boldsymbol{e}_k = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1} + \epsilon_i(\boldsymbol{x}_k) \\ &\qquad \vdots \\ &\boldsymbol{e}_k^T P^{i_{\max}+1} \boldsymbol{e}_k = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^{i_{\max}} \hat{\boldsymbol{e}}_{k+1} + \epsilon_{i_{\max}}(\boldsymbol{x}_k)\end{aligned} \tag{29}$$

where $\epsilon_i(\boldsymbol{x}_k)$ is the approximation error in the $(i+1)$th iteration, which is defined as

$$\epsilon_i(\boldsymbol{x}_k) = \min_{\Delta \boldsymbol{u}_k} \left\{ \boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_{k-1}^T + \Delta \boldsymbol{u}_k) R (\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k) + \gamma \left[\sum_{l=k}^{\infty} \gamma^{l-k} r(\boldsymbol{x}_l, \boldsymbol{u}_l) \right] \right\} - \min_{\Delta \boldsymbol{u}_k} [\boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_{k-1}^T + \Delta \boldsymbol{u}_k) R (\boldsymbol{u}_{k-1}^T + \Delta \boldsymbol{u}_k) + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1}] \tag{30}$$

Because $\boldsymbol{e}_k^T Q \boldsymbol{e}_k$ is not a function associated with control increment $\Delta \boldsymbol{u}_k$, $\epsilon_i(\boldsymbol{x}_k)$ can be rewritten as

$$\epsilon_i(\boldsymbol{x}_k) = \min_{\Delta \boldsymbol{u}_k} \left\{ (\boldsymbol{u}_{k-1}^T + \Delta \boldsymbol{u}_k) R (\boldsymbol{u}_{k-1}^T + \Delta \boldsymbol{u}_k) + \gamma \left[\sum_{l=k}^{\infty} \gamma^{l-k} r(\boldsymbol{x}_l, \boldsymbol{u}_l) \right] \right\} - \min_{\Delta \boldsymbol{u}_k} [(\boldsymbol{u}_{k-1}^T + \Delta \boldsymbol{u}_k) R (\boldsymbol{u}_{k-1}^T + \Delta \boldsymbol{u}_k) + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1}] \tag{31}$$

Main Results

This section discusses the convergence and stability of incremental value iteration. The first part analyzes the continuity of the minimization operator. The second part provides the convergence proof by introducing two exact iterations as the upper and lower bounds of approximate value iteration with approximation error. The third part derives an asymptotic stability condition of approximate value iteration.

Continuity Analysis

It has been verified that smooth function approximators "uniformly" approximate a function if the function is continuous (Haykin 2009). Otherwise, the approximation accuracy is not guaranteed to be suitable on new states that are not used in the training.

Rewrite the optimal control of exact value iteration in Heydari (2014) as

$$\boldsymbol{u}_k^* \in \underset{\boldsymbol{u}_k}{\operatorname{argmin}} [U(\boldsymbol{x}_k, \boldsymbol{u}_k) + \gamma V^*(k+1)] \tag{32}$$

Recall the optimal incremental control of incremental value iteration in Eq. (21)

$$\Delta \boldsymbol{u}_k^* = \underset{\Delta \boldsymbol{u}_k}{\operatorname{argmin}} [\boldsymbol{e}_k^T Q \boldsymbol{e}_k + (\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k)^T R (\boldsymbol{u}_{k-1} + \Delta \boldsymbol{u}_k) + \gamma V^*(k+1)] \tag{33}$$

The general value iteration does not always have analytical solution of $\boldsymbol{u}_k^*$. Incremental value iteration does not suffer from this shortcoming because the system is linearized as an incremental model. Based on the previously presented results, the tracking error prediction $\hat{\boldsymbol{e}}_{k+1}$ leads to an analytical form of approximated cost function as in Eq. (19). In Heydari (2015), the continuity of value iteration is not guaranteed. The discontinuity comes from the control policy because it is solved by Eq. (32), where the minimization operator argmin$(\cdot)$ is not necessarily continuous. However, incremental value iteration provides a direct mapping between the control increment $\Delta \boldsymbol{u}_k$ and approximated cost function $\hat{V}(k)$ according to Eq. (19). In Eq. (19), the matrices Q, R and forgetting factor γ are constant, and the system matrix $\boldsymbol{F}_{k-1}$ and control matrix $\boldsymbol{G}_{k-1}$ are fixed in time step $k-1$. The tracking error $\boldsymbol{e}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{\text{ref}}$ at step k is continuously changing. Therefore, it is concluded that $\hat{V}(k)$ at step k is continuously changing with respect to $\Delta \boldsymbol{u}_k$. As a result, the operator argmin$(\cdot)$ in Eq. (33) can be transferred into the equation to solve $\partial \hat{V}(k)/\partial(\Delta \boldsymbol{u}_k) = 0$. The conclusion is that incremental value iteration meets the continuity condition of general value iteration algorithm.

Convergence Analysis

Tamimi et al. (2008) showed that the exact value iteration is first proved to converge to the optimal cost function and optimal control. In practice, the approximators are introduced when the cost function and control policy are not exactly known. As a result, the approximation errors will affect both the convergence and stability of the approximate value iteration algorithm. This subsection will analyze the boundedness and convergence of approximated cost function sequences $\{\hat{V}^i(k)\}_{i=0}^{\infty}$. To this end, a bound on the approximation error per iteration is established, as in assumption 2.

Assumption 2. The value function approximation error defined in Eq. (30) is bounded as $|\epsilon^i(\boldsymbol{x}_k)| \leq \eta r(\boldsymbol{x}_k, 0), \quad \forall\ i \in \mathbb{N}$ for some $\eta \in [0, 1)$. The parameter η corresponds to the accuracy of the function approximator.

Define $\{\bar{V}^i(k)\}_{i=0}^{\infty}$ and $\{\underline{V}^i(k)\}_{i=0}^{\infty}$ as cost function sequences initiated from $\bar{V}^0(k)$ and $\underline{V}^0(k)$, which are generated by the following value iterations with approximation error bounds:

$$\bar{V}^{i+1}(k) = \boldsymbol{e}_k^T P^{i+1} \boldsymbol{e}_k = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1} + \eta r(\boldsymbol{x}_k, 0) \tag{34}$$

$$\underline{V}^{i+1}(k) = \boldsymbol{e}_k^T P^{i+1} \boldsymbol{e}_k = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k + \gamma \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1} - \eta r(\boldsymbol{x}_k, 0) \tag{35}$$

Lemma 1. Let $|\epsilon^i(\boldsymbol{x}_k)| \leq \eta r(\boldsymbol{x}_k, 0), \quad \forall\ i \in \mathbb{N}$ for some $\eta \in [0, 1)$. If the recursive relations given by Eqs. (28), (34), and (35) are initialized such that $\underline{V}^0(k) \leq \hat{V}^0(k) \leq \bar{V}^0(k)$, then one has $\underline{V}^i(k) \leq \hat{V}^i(k) \leq \bar{V}^i(k), \quad \forall\ i \in \mathbb{N}$. Moreover, if $\underline{V}^0(k) = \hat{V}^0(k) = \bar{V}^0(k)$, then $\underline{V}^i(k)$ and $\bar{V}^i(k)$ are, respectively, the greatest lower bound and the least upper bound of $\hat{V}^i(k)$ for $\epsilon^i(\boldsymbol{x}_k) \in [-\eta r(\boldsymbol{x}_k, 0,) + \eta r(\boldsymbol{x}_k, 0)]$.

Proof. Lemma 1 is proved using mathematical induction. The first step is to prove that $\underline{V}^0(k) \leq \hat{V}^0(k) \leq \bar{V}^0(k)$. The second step is assuming $\underline{V}^i(k) \leq \hat{V}^i(k) \leq \bar{V}^i(k)$ to prove the assumptions in Eqs. (34) and (28). As a result, it can be concluded that $\hat{V}^{i+1}(k) \leq \bar{V}^{i+1}(k)$ because $\epsilon^i(\boldsymbol{x}_k) \leq \eta r(\boldsymbol{x}_k, 0)$ and $\hat{V}^i(k) \leq \bar{V}^i(k)$. Therefore, one has $\hat{V}^i(k) \leq \bar{V}^i(k), \quad \forall\ i \in \mathbb{N}$. The proof of $\underline{V}^i(k) \leq \hat{V}^i(k)$, $\forall\ i \in \mathbb{N}$ is similar when comparing Eq. (35) with Eq. (28) and using mathematical induction. Proof of the last part of the lemma follows from assuming $\varepsilon^i(\boldsymbol{x}_k) = \eta r(\boldsymbol{x}_k, 0), \quad \forall\ i \in \mathbb{N}$ (respectively, $\epsilon^i(\boldsymbol{x}) = -\eta r(\boldsymbol{x}_k, 0), \quad \forall\ i \in \mathbb{N}$), which leads to $\hat{V}^i(k) \leq \bar{V}^i(k)$ (respectively, $\hat{V}^i(k) \underline{V}^i(k)$). Therefore, there is no greater lower bound or lesser upper bound for $\hat{V}^i(k)$.

Considering the value iterations in Eqs. (34) and (35), it is seen that $\bar{V}^i(\cdot)$ and $\underline{V}^i(\cdot)$ are, respectively, the value functions at the ith iteration of exact value iterations for cost functions as follows:

$$\bar{V}(k) = \gamma^{l-k} \sum_{l=k}^{\infty} [r(\boldsymbol{x}_l, \boldsymbol{u}_l) + \eta r(\boldsymbol{x}_l, 0)] \tag{36}$$

$$\underline{V}(k) = \gamma^{l-k} \sum_{l=k}^{\infty} [r(\boldsymbol{x}_l, \boldsymbol{u}_l) - \eta r(\boldsymbol{x}_l, 0)] \tag{37}$$

The following lemma 2 provides sufficient conditions for their convergence to the respective optimal cost functions.

Lemma 2. The exact value iterations given by Eqs. (34) and (35) converge to the optimal value of cost functions [Eqs. (36) and (37)], respectively, if they are initialized by smooth functions $\bar{V}^0(\cdot)$ and $\underline{V}^0(\cdot)$, such that $0 \le \bar{V}^0(k) \le (1+\eta)r(\boldsymbol{x}_k, 0), \quad \forall\, \boldsymbol{x}_k \in \Omega$ and $0 \le \underline{V}^0(k) \le (1-\eta)r(\boldsymbol{x}_k, 0), \quad \forall\, \boldsymbol{x}_k \in \Omega$, where $\eta \in [0, 1)$, $\Omega \subset \mathbb{R}^n$ is a compact set containing the given system.

Proof. The proof follows the results in the literature (Tamimi et al. 2008; Heydari 2014) because the value iterations of $\bar{V}(k)$ in Eq. (34) and $\underline{V}(k)$ in Eq. (35) are in fact exact value iterations.

Following lemma 1 and lemma 2, theorem 2 proves the boundedness of the elements in approximate cost function sequences $\{\hat{V}^i(k)\}_{i=0}^{\infty}$.

Theorem 2. Let $|\epsilon^i(\boldsymbol{x}_k)| \le \eta r(\boldsymbol{x}_k, 0), \quad \forall\, \boldsymbol{x}_k \in \Omega$, then the elements of sequence $\{\hat{V}^i(k)\}_{i=0}^{\infty}$ as $i \to \infty$ are bounded by the optimal cost functions in Eqs. (36) and (37) denoted with $\bar{V}^*(k)$ and $\underline{V}^*(k)$, respectively, in the sense that the greatest lower bound of $\hat{V}^i(k)$ converges to $\underline{V}^*(k)$ and the least upper bound of $\hat{V}^i(k)$ converges to $\bar{V}^*(k)$ as $i \to \infty$.

Proof. The proof follows from the boundedness of $\{\hat{V}^i(k)\}_{i=0}^{\infty}$ given in lemma 1 and the convergence of the bounds for smooth $\underline{V}^0(k)$ and $\bar{V}^0(k)$, which satisfy $0 \le \underline{V}^0(k) = \hat{V}^0(k) = \bar{V}^0(k) \le (1-\eta)r(\boldsymbol{x}_k, 0), \quad \forall\, \boldsymbol{x}_k \in \Omega$ based on lemma 2.

Based on the result in theorem 2, the following theorem 3 analyzes the convergence of approximate value iteration when $\eta \to 0$.

Theorem 3. Let $|\epsilon^i(\boldsymbol{x}_k)| \le \eta r(\boldsymbol{x}_k, 0), \quad \forall\, \boldsymbol{x}_k \in \Omega, \ \forall\, i \in \mathbb{N}$ for some $\eta \in [0, 1)$. Let the approximate value iteration given by Eq. (28) be initialized such that $0 \le \hat{V}^0(k) \le (1-\eta)r(\boldsymbol{x}_k, 0)$, $\forall\, \boldsymbol{x}_k \in \Omega$. As $\eta \to 0$, for example by selecting a richer approximator, the results from the approximate value iteration converge to the results from the exact value iteration uniformly in compact set Ω. More specifically, the least upper bound and the greatest lower bound of $\hat{V}^i(k)$ for $i \to \infty$ converge uniformly to the optimal cost function associated with cost function as $\eta \to 0$.

Proof. Define $V^*(k)$ as the optimal value of cost function $V(k)$, and define $\tilde{V}^*(k)$ as the optimal value of a new cost function $\tilde{V}(k)$. $\tilde{V}^*(k)$ is defined as

$$\tilde{V}^*(k) := \sum_{l=k}^{\infty} \gamma^{l-k} r(\boldsymbol{x}_l^{h^*}, 0) \tag{38}$$

where $x_l^{h^*} := f(x_{l-1}^{h^*}, h^*(\boldsymbol{x}_{l-1}^{h^*})), \quad \forall\, l \in \mathbb{N} - \{0\}$ and $\boldsymbol{x}_0^{h^*} := \boldsymbol{x}_0$. Notably, $\tilde{V}(k)$ is a cost function that considers a utility function as $r(\boldsymbol{x}_k, 0)$, without control $\boldsymbol{u}_k$ in utility function.

It is concluded that the exact value iterations of $V(k)$ and $\bar{V}(k)$, $\underline{V}(k)$ are in fact special cases of approximate values iteration of $\hat{V}(k)$ with different value of approximation error $\epsilon^i(\boldsymbol{x}_k)$, i.e., $\epsilon^i(\boldsymbol{x}_k) = 0$, $\epsilon^i(\boldsymbol{x}_k) = \eta r(\boldsymbol{x}_k)$, $\epsilon^i(\boldsymbol{x}_k) = -\eta r(\boldsymbol{x}_k)$, respectively (see Fig. 1). Therefore, one has

$$V^*(k) \le \bar{V}^*(k) \tag{39}$$

where $\bar{V}^*(k)$ is the optimal value of cost function Eq. (36); otherwise, the control resulting from $\bar{V}^*(k)$ will be the optimal control for cost function Eq. (13).

According to the definition of $\bar{V}^*(k)$, one has

$$\bar{V}^*(k) \le V^*(k) + \eta \tilde{V}^*(k) \tag{40}$$

otherwise $\bar{V}^*(k)$ will not be the optimal value function of cost function Eq. (36). Note that both sides of inequality Eq. (40) include infinite sums of $r(\boldsymbol{x}_k, \boldsymbol{u}_k) + \eta r(\boldsymbol{x}_k, 0)$ terms, but they are evaluated along different trajectories (i.e., the applied control policies are different). The summation in the left-hand side is based on the control that minimizes cost function Eq. (36), and the summation in the right-hand side is based on the control that minimizes cost function Eq. (13).

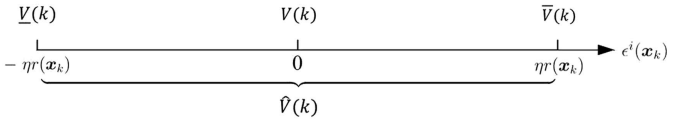

Fig. 1. The relation between $\hat{V}(k)$ and $\underline{V}(k)$, $\bar{V}(k)$, $V(k)$.

From the inequalities in Eqs. (39) and (40), one has

$$|V^*(k) - \bar{V}^*(k)| \le \eta \tilde{V}^*(k) \tag{41}$$

Let $\tilde{V}^*_{\max}(k) := \sup_{\boldsymbol{x}(k) \in \Omega} \tilde{V}^*(k)$, where $\tilde{V}^*_{\max}(k)$ is the upper bound of $\tilde{V}^*(k)$. Therefore, one can rewrite Eq. (41) as

$$|V^*(k) - \bar{V}^*(k)| \le \eta \tilde{V}^*(k)_{\max} \tag{42}$$

Inequality Eq. (42) proves the convergence of $\bar{V}^*(k)$ to the optimal value $V^*(k)$ associated with cost function Eq. (13) as $\eta \to 0$. Moreover, the right-hand side of inequality Eq. (42) is a constant so that it is independent of initial time; this convergence is uniform. Let $\tilde{\underline{V}}^*(k)$ be defined as

$$\tilde{\underline{V}}^*(k) := \sum_{l=k}^{\infty} \gamma^{l-k} r(\boldsymbol{x}_l^{\underline{h}^*}, 0) \tag{43}$$

where $\underline{h}^*(\cdot)$ is the optimal control policy for cost function $\underline{V}(\cdot)$, i.e., the summation in the right-hand side of Eq. (43) is evaluated along the trajectory that is optimal with respect to $\underline{V}(\cdot)$ given by Eq. (37). Similarly, one has that $\underline{V}^*(k) \le V^*(k)$ and $V^*(k) \le \underline{V}^*(k) + \eta \tilde{\underline{V}}^*(k)$, which leads to

$$|V^*(k) - \underline{V}^*(k)| \le \eta \tilde{\underline{V}}^*(k) \tag{44}$$

Defining $\tilde{\underline{V}}^*_{\max}(k) := \sup_{\boldsymbol{x}(k) \in \Omega} \tilde{\underline{V}}^*(k)$, a similar uniform convergence can be concluded because the right-hand side of inequality Eq. (44) will be upper bounded by the t_0 -independent term $\eta \tilde{\underline{V}}^*_{\max}(k)$. It should be noted that $\tilde{\underline{V}}^*_{\max}(k)$ will be a finite constant as long as $\eta \in [0, 1)$, due to the upper boundedness of $V^*(k)$, which leads to an upper-bounded $\underline{V}^*(k)$, because $\underline{V}^*(k) \le V^*(k)$. Consider the utility function as $r(\boldsymbol{x}_k, \boldsymbol{u}_k) = Q(\boldsymbol{x}_k) + \boldsymbol{u}_k^T R \boldsymbol{u}_k$, which is derived from Eq. (37), so that

$$\begin{aligned}
\underline{V}^*(k) &= \sum_{l=k}^{\infty} \gamma^{l-k} [r(\boldsymbol{x}_l, \boldsymbol{u}_l) - \eta r(\boldsymbol{x}_l, 0)] \\
&= \sum_{l=k}^{\infty} \gamma^{l-k} [Q(\boldsymbol{x}_l^{\underline{h}^*}) + \underline{h}^{*T}(\boldsymbol{x}_l^{\underline{h}^*}) R \underline{h}^*(\boldsymbol{x}_l^{\underline{h}^*}) - \eta Q(\boldsymbol{x}_l^{\underline{h}^*})] \\
&= \sum_{l=k}^{\infty} \gamma^{l-k} [(1-\eta) Q(\boldsymbol{x}_l^{\underline{h}^*}) + \underline{h}^{*T}(\boldsymbol{x}_l^{\underline{h}^*}) R \underline{h}^*(\boldsymbol{x}_l^{\underline{h}^*})]
\end{aligned} \tag{45}$$

From the result of theorem 2, $\underline{V}^*(k)$ is bounded, which leads to a bounded $\sum_{l=k}^{\infty}(1-\eta)Q(\boldsymbol{x}_l^{\underline{h}^*})$, and the boundedness of the latter leads to a bounded cost function $\tilde{\underline{V}}^*(k) = \sum_{l=k}^{\infty} \gamma^{l-k} Q(\boldsymbol{x}_l^{\underline{h}^*})$ when $0 \le \eta < 1$. Therefore, the right-side of Eq. (44) is bounded and approaches 0 as $\eta \to 0$. This result proves that the exact value iteration of $\underline{V}^*(k)$ converges to the optimal value of exact value iteration $V^*(k)$ as $\eta \to 0$.

Stability Analysis

This subsection first introduces two errors: the convergence error of approximate value iteration and approximation error of control policy using a smooth approximator, or actor. Then, the stability result of the approximate value iteration considering the aforementioned two errors is provided.

Let the approximate value iteration be terminated at the ith iteration, once a convergence tolerance, denoted with positive (semi-) definite function $\delta(\boldsymbol{x}_k)$, is achieved, i.e., when

$$|\hat{V}^{i+1}(k) - \hat{V}^i(k)| \leq \delta(\boldsymbol{x}_k), \quad \forall\, \boldsymbol{x}_k \in \Omega \tag{46}$$

Heydari (2015) considered the approximation error of using an actor to approximate control policy. For incremental value iteration, an analytical optimal solution is provided; thus, the approximator error does not exist. However, compared to exact value iteration, incremental value iteration assumes that the nonlinear system is approximated as a linear incremental model, leading to the model approximation error, which is included in prediction error of $\hat{\boldsymbol{e}}_{k+1}$ with respect to $\boldsymbol{e}_{k+1}$, and thus is included in modified approximated cost function $\hat{V}'(k+1) = \hat{\boldsymbol{e}}_{k+1}^T P \hat{\boldsymbol{e}}_{k+1}$.

Rewrite Eq. (46) as

$$-\delta(\boldsymbol{x}_k) \leq \hat{V}^{i+1}(k) - \hat{V}^i(k) \leq \delta(\boldsymbol{x}_k), \quad \forall\, \boldsymbol{x}_k \in \Omega \tag{47}$$

Thus

$$\hat{V}^i(k) - \delta(\boldsymbol{x}_k) \leq \hat{V}^{i+1}(k) \leq \hat{V}^i(k) + \delta(\boldsymbol{x}_k), \quad \forall\, \boldsymbol{x}_k \in \Omega \tag{48}$$

From the right side of Eq. (48), one has

$$\hat{V}^i(k) + \delta(\boldsymbol{x}_k) \geq \hat{V}^{i+1}(k) \tag{49}$$

Using Eq. (28) in Eq. (49) yields

$$\hat{V}^i(k) \geq \boldsymbol{e}_t^T Q \boldsymbol{e}_k + (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + \gamma \hat{V}^i(k+1) + \epsilon_i(\boldsymbol{x}_k) - \delta(\boldsymbol{x}_k), \quad \forall\, \boldsymbol{x}_k \in \Omega \tag{50}$$

The asymptotic stability for the discrete nonlinear system Eq. (1) is defined as

$$\Delta \hat{V}^i(k+1) \coloneqq \hat{V}^i(k+1) - \hat{V}^i(k) \leq 0, \quad \forall\, \boldsymbol{x}_k \in \Omega \tag{51}$$

The equality in Eq. (51) holds only at the equilibrium $\boldsymbol{x}_k = 0$.

From Eq. (50), one gets

$$\hat{V}^i(k+1) - \hat{V}^i(k) \leq -\boldsymbol{e}_k^T Q \boldsymbol{e}_k - (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i + (1-\gamma)\hat{V}^i(k+1) - \epsilon_i(\boldsymbol{x}_k) + \delta(\boldsymbol{x}_k) \tag{52}$$

According to the stability condition in Eq. (51), one has

$$-\boldsymbol{e}_k^T Q \boldsymbol{e}_k - (\boldsymbol{u}_k^i)^T R \boldsymbol{u}_k^i - \epsilon_i(\boldsymbol{x}_k) + \delta(\boldsymbol{x}_k) + (1-\gamma)\hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1} \leq 0 \tag{53}$$

Remark 5. This inequality is the sufficient and necessary condition of asymptotic stability for a nonlinear dynamic system [Eq. (1)] subjected to the designed optimal control in Eq. (36), considering the approximation error at each iteration $\epsilon^i(\boldsymbol{x}_k)$ and iteration tolerance $\delta(\boldsymbol{x}_k)$. Notably, the utility function $r(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{e}_k^T Q \boldsymbol{e}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k$, one-step value iteration error $\epsilon_i(\boldsymbol{x}_k)$, convergence tolerance $\delta(\boldsymbol{x}_k)$, and value function prediction $\hat{V}^i(k+1) = \hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1}$ are the factors that affect the system stability. $\epsilon_i(\boldsymbol{x}_k)$, $\delta(\boldsymbol{x}_k)$ are independent of control $\boldsymbol{u}_k$. The utility function $r(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is an essential factor to affect the system stability, i.e., the larger $r(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is, the more stable the system is. The term $(1-\gamma)\hat{\boldsymbol{e}}_{k+1}^T P^i \hat{\boldsymbol{e}}_{k+1}$ is also related to $\boldsymbol{u}_k$, but it is relatively small because $1-\gamma$ is close to zero.

Remark 6. $\epsilon^i(\boldsymbol{x}_k)$ describes the difference between using exact value iteration and approximation value iteration. The value of $\epsilon^i(\boldsymbol{x}_k)$ can be either positive or negative, which has different effects on stability. $\epsilon^i(\boldsymbol{x}_k) > 0$ indicates that the Lyapunov function of the approximate value iteration represented by $\hat{V}^i(k+1)$ is smaller than $V^i(k+1)$ of the exact value iteration, so that it is easier to have $\hat{V}^i(k+1) < \hat{V}^i(k)$ and make the system more stable. On the contrary, $\epsilon^i(\boldsymbol{x}_k) < 0$ indicates that the Lyapunov function $\hat{V}^i(k+1)$ is larger than $V^i(k+1)$, so that it is harder to have $\hat{V}^i(k+1) < \hat{V}^i(k)$ and it makes the system more unstable. $\delta(\boldsymbol{x}_k)$ measures to what extent the value iteration goes, i.e., a large $\delta(\boldsymbol{x}_k)$ indicates the current iteration index i is not sufficient to get an accurate estimate of $V^*(\boldsymbol{x}_k)$, which may lead to instability of the system.

Remark 7. Asymptotic convergence of the cost function $\hat{V}^i(k)$ in the exact value iteration, described in Eq. (49), is the backbone of deriving the stability condition for approximate value iteration (including incremental approximate value iteration) in Eq. (53). The control has to first guarantee that the oscillation in the convergence of the approximated cost function is bounded in the presence of $\epsilon^i(\boldsymbol{x}_k)$.

Numerical Example: Flight Control Problem

This section assesses the developed incremental value iteration algorithm on a practical flight control problem. Firstly, the longitudinal attitude dynamics of an aircraft model are provided. Secondly, the flight dynamics are discretized from a continuous-time model to a discrete-time linear model, by using Taylor expansion. Simulation results are presented to analyze the performance of the designed adaptive flight controller.

Continuous-Time Dynamical Model

A nonlinear longitudinal dynamical model of the aerial vehicle (Sonneveldt 2011) is provided as:

$$\begin{aligned} \dot{\alpha} &= q + \frac{\bar{q}S}{mV} C_z(\alpha, q, M_\alpha, \delta_e) + \frac{g}{V} \\ \dot{q} &= \frac{qSd}{I_{yy}} C_m(\alpha, q, M_\alpha, \delta_e) \end{aligned} \tag{54}$$

where $\bar{q} = 1/2\rho V^2$ is dynamic pressure, S is a reference area, m is mass, V is speed, d is reference length, and I_{yy} is the pitching moment of inertia. C_z and C_m are the aerodynamic force and moment coefficients, which are nonlinear functions.

The following aerodynamic parameters of this model are valid for $-10° < \alpha < 10°$:

$$\begin{aligned} C_z(\alpha, q, M_\alpha, \delta_e) &= C_{z1}(\alpha, M_\alpha) + B_z \delta_e \\ C_m(\alpha, q, M_\alpha, \delta_e) &= C_{m1}(\alpha, M_\alpha) + B_m \delta_e \\ B_z &= b_1 M_\alpha + b_2 \\ B_m &= b_3 M_\alpha + b_4 \\ C_{z1}(\alpha, M_\alpha) &= \phi_{z1}(\alpha) + \phi_{z2} M_\alpha \\ C_{z2}(\alpha, M_\alpha) &= \phi_{m1}(\alpha) + \phi_{m2} M_\alpha \\ \phi_{z1}(\alpha) &= h_1 \alpha^3 + h_2 \alpha|\alpha| + h_3 \alpha \\ \phi_{m1}(\alpha) &= h_4 \alpha^3 + h_5 \alpha|\alpha| + h_6 \alpha \\ \phi_{z2} &= h_7 \alpha|\alpha| + h_8 \alpha \\ \phi_{m2} &= h_9 \alpha|\alpha| + h_{10} \alpha \end{aligned} \tag{55}$$

Table 1. Parameter initialization values

Parameter	Value
Q	$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$
R	0.1
γ_V	0.99
γ_{RLS}	0.99
κ	0.99
$\hat{F}_0$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\hat{G}_0$	$[0 \quad 0]^T$
Λ_0	$10^8 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
P_0	0
ΔT	0.01

where $b_1, \ldots, b_4, h_1, \ldots, h_{10}$ are constant coefficients in the flight envelope, and the March number M_α is set to be 2.2.

For convenience of controller design, a simplified model is given as

$$\begin{aligned} \dot{\alpha} &= q - \frac{C_Y^\alpha}{mV}\alpha - \frac{C_Y^{\delta_e}}{mV}\delta_e + \frac{g}{V} \\ \dot{q} &= \frac{M_z^\alpha}{I_{yy}}\alpha + \frac{M_z^{\delta_e}}{I_{yy}}\delta_e + \frac{M_z^q}{I_{yy}}q \end{aligned} \tag{56}$$

Selecting the state vector as $\boldsymbol{x} = [\alpha, q]^T$ and the control input as elevator deflection δ_e, the state equation of the tracking problem can be written as a state-space form

$$\dot{x} = A(x)x + B(x)u + H(x)d \tag{57}$$

where $\boldsymbol{x} = [\alpha, q]^T$, $u = \delta_e$, $d = g$, $A(x)$, $B(x)$, $H(x)$ are defined as

$$A(x) = \begin{bmatrix} \frac{-C_Y^\alpha}{mV} & 1 \\ \frac{M_z^\alpha}{I_{yy}} & \frac{M_z^q}{I_{yy}}q \end{bmatrix}, \quad B(x) = \begin{bmatrix} -\frac{C_Y^{\delta_e}}{mV} \\ \frac{M_z^{\delta_e}}{I_{yy}} \end{bmatrix}, \quad H(x) = \begin{bmatrix} \frac{1}{V} \\ 0 \end{bmatrix} \tag{58}$$

Discrete-Time Incremental Model

The incremental model considers the increment of control input, which is established on a discrete-time model. This subsection

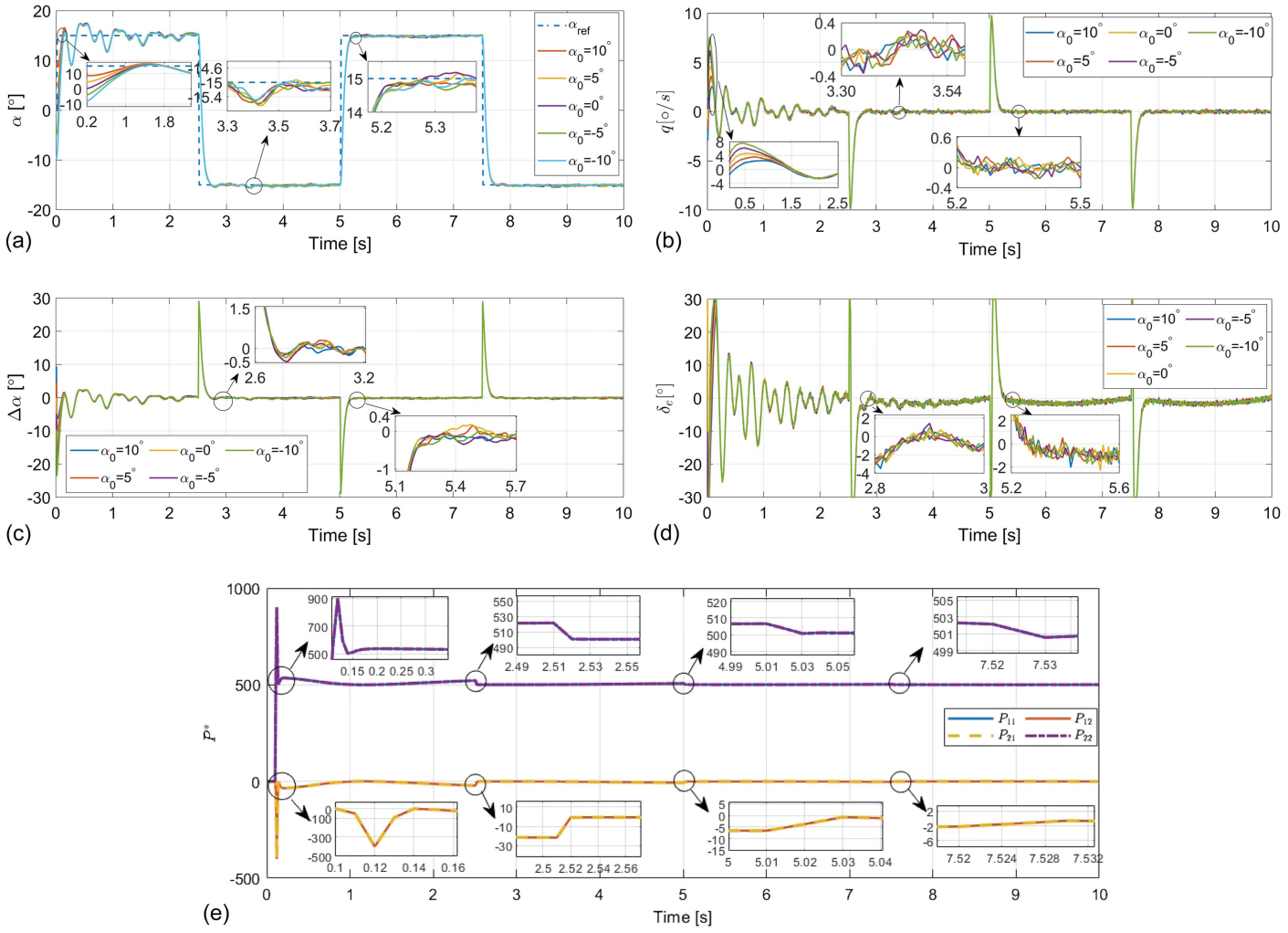

Fig. 2. Flight control response and control input in nominal case to track square wave α_{ref}: (a) α tracking trajectory; (b) pitch rate q trajectory; (c) α tracking error; (d) elevator deflection δ_e; and (e) parameters of optimal kernel matrix P^*.

J. Aerosp. Eng., 2024, 37(1): 04023097

first discretizes the continuous-time model of the aerial vehicle in Eq. (57) into a discrete-time model, with a prescribed sampling time. Then, the discrete-time model is linearized into an incremental model. The incremental model describes the dynamics of the state vector increment associated with the input control increment.

In a certain trim point of flight, the parameters in $A(x)$, $B(x)$, $H(x)$ of the continuous-time aerial vehicle dynamics in Eq. (57) are assumed to be fixed. Then, the system in Eq. (57) can be seen as a time-invariant system. Followed by this assumption, one can discretize Eq. (57) as

$$\boldsymbol{x}_{k+1} = \Phi(\boldsymbol{x}_k, \Delta T)\boldsymbol{x}_k + G(\boldsymbol{x}_k, \Delta T)u_k + H(\boldsymbol{x}_k)d_k \tag{59}$$

where $\Phi(\boldsymbol{x}_k, \Delta T) = e^{A\Delta T}$, $G(\boldsymbol{x}_k, \Delta T) = \int_0^{\Delta T} e^{A\tau}B d\tau$, and ΔT is the sampling time.

Taking the eaylor Expansion of Eq. (59) at $\boldsymbol{x}_k$ yields

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + F_{k-1}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) + G_{k-1}(u_k - u_{k-1}) \tag{60}$$

where $F_{k-1} = \frac{\partial \Phi(x_k,\Delta T)x_k + G(x_k,\Delta T)u_k + H(x)d_k}{\partial x_k}|_{\boldsymbol{x}_{k-1},u_{k-1}}$, $G_{k-1} = \frac{\partial \Phi(x_k,\Delta T)x_k + G(x_k,\Delta T)u_k + H(x)d_k}{\partial u}|_{\boldsymbol{x}_{k-1},u_{k-1}}$.

Eq. (60) is rewritten in an incremental form as

$$\Delta \boldsymbol{x}_{k+1} = F_{k-1}\Delta \boldsymbol{x}_k + G_{k-1}\Delta u_k \tag{61}$$

where $\Delta \boldsymbol{x}_{k+1} = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$, $\Delta \boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ are the increments of state vector at time step $k+1, k$. $\Delta u_k = u_k - u_{k-1}$ is the increment of control input at time step k. Notably, the matrices F_{k-1}, G_{k-1} are functions associated with sampling time ΔT.

Simulation Results

This subsection provides the flight control simulation of an aerial vehicle. The dynamics considered are the longitudinal model of angle of attack α and pitch rate q, controlled by elevator deflection δ_e. The saturation limit of the actuator is set to be $-30° \le \delta_e \le 30°$. The initialization of controller parameters is provided in Table 1. The reference signals of α are considered as

- Square Wave Reference Signal: $\alpha_{\text{ref}} = 4/45\pi \text{sign}(\sin(0.4\pi t))$
- Sine Wave Reference Signal: $\alpha_{\text{ref}} = 4/45\pi \sin(0.4\pi t)$

Persistent excitation (PE) is required in simulation for two purposes. The first one is to provide exploration in order to achieve a better policy evaluation; the second one is to excite the RLS identification process of the incremental model. To this end, the PE signal is appended to the control input as a probing noise

$$\Delta u_k' = \Delta u_k + n_k \tag{62}$$

where n_k is a signal that is a sum of sines with various amplitudes, frequency, and phase. The amplitudes of n_k are required not to be large compared to the amplitude of Δu_k so as to decrease their effects on the performance of the controller. This simulation uses a PE signal as (de Alvear Cárdenas et al. 2018)

$$n_k = 0.3e^{-k\Delta T}[\sin(-20k\Delta T) + \sin(10k\Delta T) + \cos(30k\Delta T)] \tag{63}$$

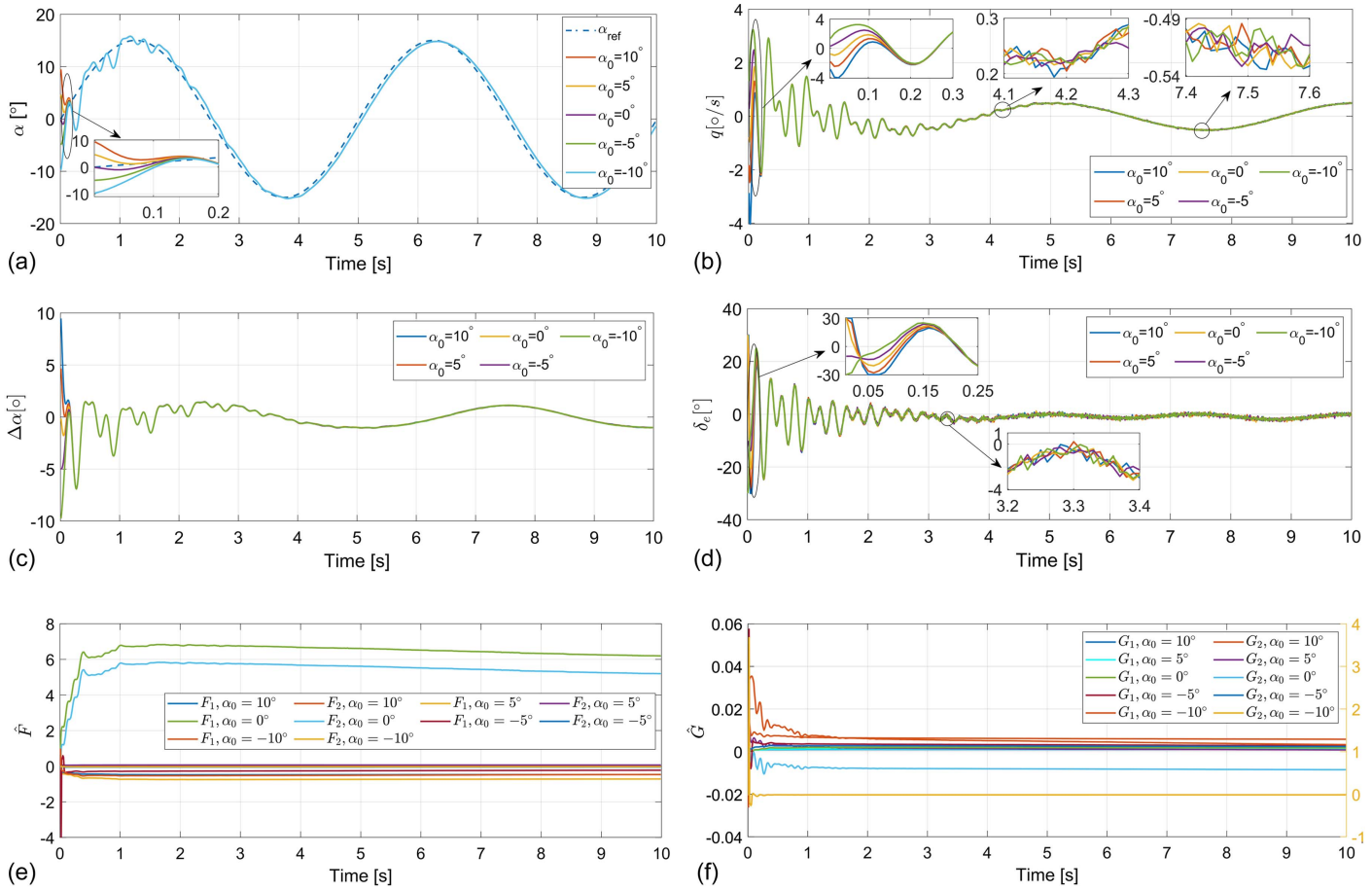

Fig. 3. Flight control response and control inputs in nominal case to track a sinusoidal α_{ref}: (a) α tracking trajectory; (b) pitch rate q trajectory; (c) α tracking error; (d) elevator deflection δ_e; (e) identification of system matrix $\hat{F}$; and (f) identification of control matrix $\hat{G}$.

Robustness to Initial Values of α and Sensor Noise

This simulation verifies the performance of the flight controller on reference tracking. The robustness of the adaptive flight controller is verified by setting different initial values α_0. The reason to verify the robustness of flight controller to initial values of α is that, in practical flights, α has various initial values and is difficult to be predicted. In these cases, the flight controller should have satisfactory performances. The robustness of the flight control system to sensor noise is another essential property in practical cases. This subsection considers the measurement noises of states α, q and assumes that $n_\alpha \sim \mathcal{N}(0, 0.001)$rad, $n_q \sim \mathcal{N}(0, 0.001)$rad/s.

The first case in this simulation is to track a square wave reference signal α_{ref}. As can be seen from Figs. 2(a and c), the tracking curve is oscillating at the initial 2.5 s, because the PE signal is contained in the control input as an input disturbance. As the amplitude of PE signal vanishes after 2.5 s, the tracking error of α_{ref} reduces. In the case of different initial values α_0, varying from $-15°$ to $15°$, the adaptive controller is capable of stabilizing the tracking error $\alpha_{k\text{ref}} - \alpha_k$, in the presence of disturbance input. From Fig. 2(d), the control input δ_e oscillates before 2.5 s between $[-30°, 30°]$. Due to the presence of sensor noises n_α and n_q, δ_e shows sawtooth oscillations in the whole control process, leading to oscillations in the curves of q and α.

As can be seen from Fig. 2(e), the optimal parameters P_{11}^*, P_{12}^*, P_{21}^*, P_{22}^* of kernel matrix P are searched through value iteration at every time step k. The peaks appearing at $t_1 = 0.12$ s are caused by the inaccurate identification of system matrix F and control matrix G, because the approximated cost function $\hat{V}(k)$ is a function associated with $\hat{F}_{k-1}$, $\hat{G}_{k-1}$. Notably, some jumps appear at time periods $\Delta t_2 = [2.51\text{ s}, 2.52\text{ s}]$, $\Delta t_3 = [5.01\text{ s}, 5.03\text{ s}]$, $\Delta t_4 = [7.52\text{ s}, 7.53\text{ s}]$. These jumps are caused by a sudden change of tracking error when the reference signal α_{ref} switches. The adaptive controller has to replan a new policy to optimize $\hat{V}_k$ jumps so that they can be regarded as transitional stages from the former tracking error sequence to the current tracking error sequence.

The second case is to track a sine-form wave reference signal. As can be seen from Figs. 3(a and c), the tracking performance of state α oscillates before 2.5 s due to the presence of the PE signal. After 5 s, the tracking error $\Delta\alpha$ increases because the controller can not follow the α_{ref} when it is changing fast, and the elevator deflection δ_e has slight oscillation. In Fig. 3(d), the elevator deflection δ_e varies in a constrained range, disturbed by the PE signal. When the tracking error decreases, δ_e vanishes. In Figs. 3(e and f), the elements of the estimated system matrix $\hat{F}_{k-1}$ and control matrix $\hat{G}_{k-1}$ converge to their true values in less than 2 s. Peaks occur in the estimation curves of $\hat{F}_{21}$, $\hat{G}_2$. This phenomenon is caused by two reasons: (1) the PE signal excites the dynamical system, thus the output states are affected; and (2) the estimation of covariance matrix Λ varies from a large initial value $\Lambda_0 = 10^8 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, which works as a high gain in innovation ε_k feedback, as shown in Eq. (8).

Adaptation to Flight Faults

Aircrafts suffer from various faults in practice that introduce uncertainties to the dynamical model. An adaptive flight controller should be able to deal with these faults as well as track the reference command.

This simulation considers two common faults of aircraft actuators, which can be used to simulate the fault effects in a real flight environment (Wang et al. 2019b). The first fault is the sudden 50%

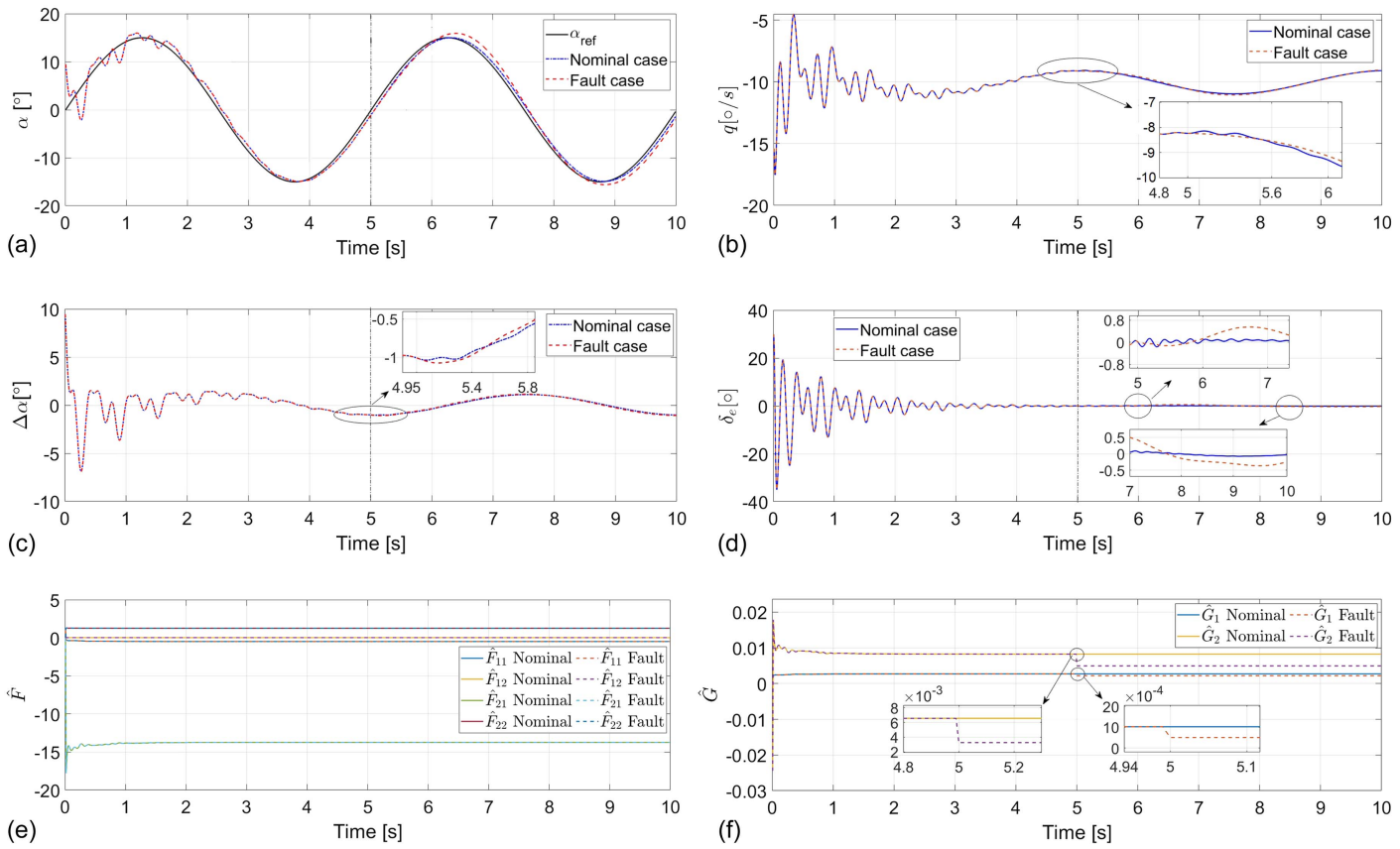

Fig. 4. Flight control response and control input in fault case of 50% control effectiveness loss ($t = 5$ s): (a) α tracking trajectory; (b) pitch rate q trajectory; (c) α tracking error; (d) elevator deflection δ_e; (e) identification of system matrix $\hat{F}$; and (f) identification of control matrix $\hat{G}$.

loss of control effectiveness at $t = 5$ s, i.e., $C_z^{\delta_e}|_{\text{fault}} = 0.5C_z^{\delta_e}|_{\text{nominal}}$, $M_z^{\delta_e}|_{\text{fault}} = 0.5M_z^{\delta_e}|_{\text{nominal}}$. In Figs. 4(a and c), when the fault occurs, the state α leaves the reference α_{ref}, the tracking error $\Delta\alpha$ shows a slight wave, and then decreases gradually. This phenomenon indicates that,the tracking of α is slightly affected by the fault. In Fig. 4(d), the elevator deflection δ_e shows an obvious wave when the fault occurs. The peaks go to 10° and −5°. Due to the loss of control effectiveness, the elevator needs more deflections to control the states α and q. In Figs. 4(e and f), the identifications of system matrix $\hat{F}$ and control system $\hat{G}$ are provided. Specifically, Fig. 4(e) shows that the loss of control effectiveness occurs at $t = 5$ s does not affect the identification of $\hat{F}$. In Fig. 4(f), due to the change of aerodynamic parameters $C_Y^{\delta_e}$, $M_z^{\delta_e}$ at $t = 5$ s, the identification of $\hat{G}$ shows a transition phase in less than 0.02 s. This is because G has been changed after the fault occurs. The RLS algorithm has to weigh between the former identified value of $\hat{G}$ and the present measured data, to modify $\hat{G}$. A short transition phase demonstrated that RLS algorithm is able to identify the fault online fast.

The second fault considered is a biased elevator at $t = 5$ s, leading to a constant biased deflection, i.e., $\Delta\delta_e = 3°$. This biased

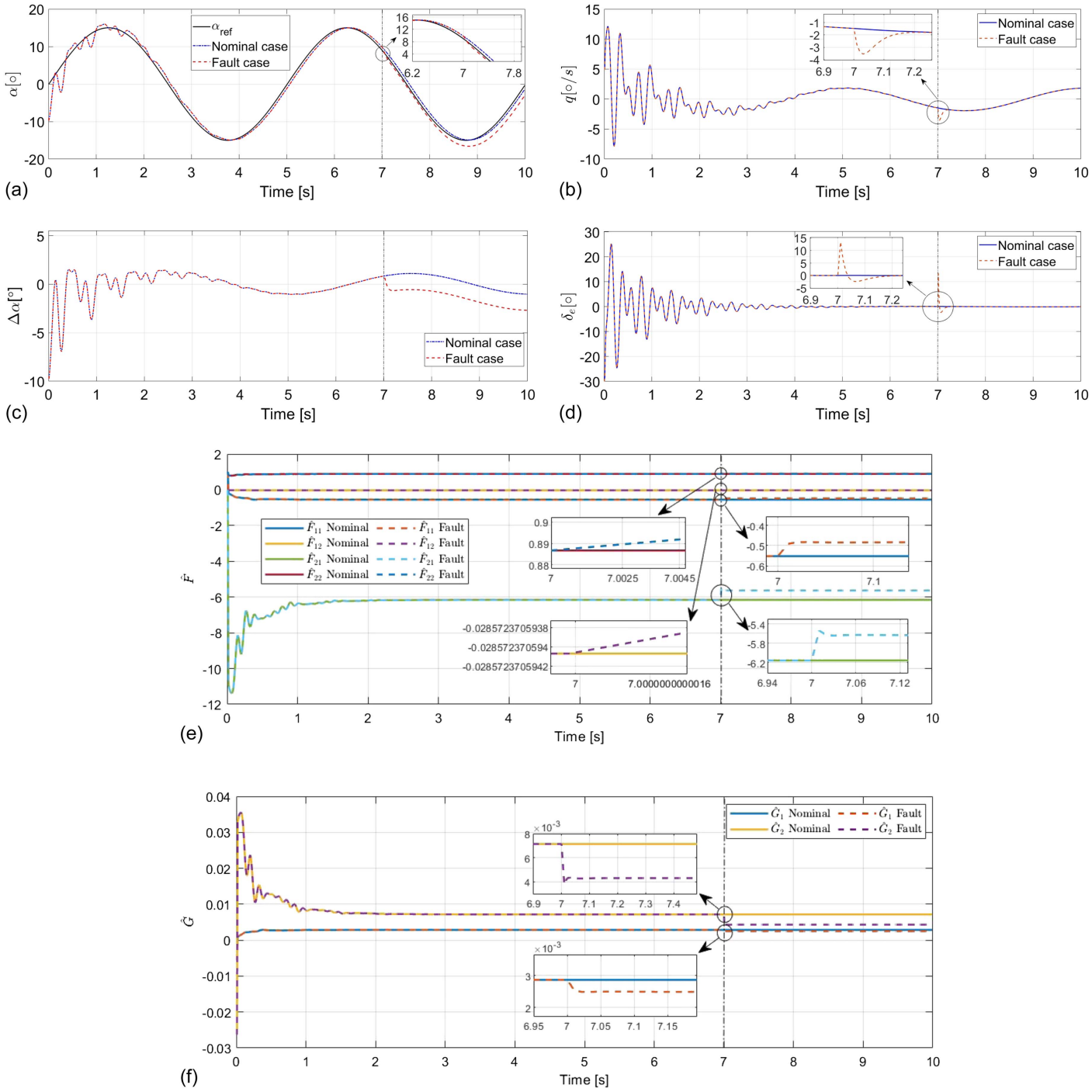

Fig. 5. Flight control response and control inputs in fault case of constant biased deflection $\Delta\delta_e = 3°(t = 7$ s): (a) α tracking trajectory; (b) pitch rate q trajectory; (c) α tracking error; (d) elevator deflection δ_e; (e) identification of system matrix $\hat{F}$; and (f) identification of control matrix $\hat{G}$.

J. Aerosp. Eng., 2024, 37(1): 04023097

deflection introduces additional lift $C_Y^{\delta_e}\Delta\delta_e$ and pitch moment $M_z^{\delta_e}\Delta\delta_e$. In Figs. 5(a and c), when the fault occurs, α fails to track α_{ref}. After an adjustment in less than 0.2 s, the pitch rate q in the fault case follows the pitch rate q in the nominal case again, indicating that the controller is capable of adapting to the constant disturbance deflection of elevator. However, a constant static error of α tracking appears despite the fact that pitch rate q in the fault case follows q in the nominal case. In Figs. 5(e and f), when the biased deflection $\Delta\delta_e$ takes effect, the elements in estimated $\hat{F}$, $\hat{G}$ in the fault case jumps from those in the nominal case. The changes of $\hat{F}$, $\hat{G}$ indicate that the constant biased deflection $\Delta\delta_e$ affects the identification result of F, G. This can be explained by the fact that $\Delta\delta_e$ produces additional input lift and pitch moment, equivalent to the effects of modified F, G under the same input δ_e, without biased deflection $\Delta\delta_e$.

Conclusion

The incremental value iteration algorithm is developed in this paper for the optimal tracking control of a nonlinear discrete-time system. Theoretical results prove that the incremental value iteration is stable when taking the nonlinear system approximation error and cost function approximation error into account. An asymptotic stability condition is developed when considering the approximation errors. Simulation examples applied to an aerial vehicle verified that the controller designed using the incremental value iteration is robust to different values of α_0. In fault-tolerant simulation, the RLS algorithm identifies the incremental model online without model information. The adaptive controller is capable of tracking the reference signals when two different faults happen.

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

References

Acquatella, P., E. Kampen, and Q. P. Chu. 2013. "Incremental backstepping for robust nonlinear flight control." In *Proc., EuroGNC 2013: 2nd CEAS Specialist Conf. on Guidance, Navigation and Control*. Brussels, Belgium: Council of European Aerospace Societies.

Balakrishnan, S. N., J. Ding, and F. Lewis. 2018. "Issues on stability of ADP feedback controllers for dynamical systems." *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 38 (4): 913–917. https://doi.org/10.1109/TSMCB.2008.926599.

Bertsekas, D. P. 2019. *Reinforcement learning and optimal control*. 1st ed. Belmont, MA: Athena Scientific.

de Alvear Cárdenas, J. I., and B. Sun, and E. J. Van Kampen. 2018. "Intelligent adaptive control using LADP and IADP applied to F-16 aircraft with imperfect measurements." In *Proc., AIAA Scitech 2021 Forum*, 1119. Reston, VA: American Institute of Aeronautics and Astronautics.

Guo, W., J. Si, F. Liu, and S. Mei. 2018. "Policy approximation in policy iteration approximate dynamic programming for discrete-time nonlinear system." *IEEE Trans. Neural Networks Learn. Syst.* 29 (7): 2794–2807. https://doi.org/10.1109/TNNLS.2017.2702566.

Haykin, S. 2009. *Neural networks and learning machines*. 3rd ed. New York: Pearson Education.

Heydari, A. 2014. "Revisiting approximate dynamic programming and its convergence." *IEEE Trans. Cybern.* 44 (12): 2733–2743. https://doi.org/10.1109/TCYB.2014.2314612.

Heydari, A. 2015. "Theoretical and numerical analysis of approximate dynamic programming with approximation errors." *J. Guid. Control Dyn.* 39 (2): 301–311. https://doi.org/10.2514/1.G001154.

Heydari, A. 2018. "Stability analysis of optimal adaptive control using value iteration with approximation errors." *IEEE Trans. Autom. Control* 63 (9): 3119–3126. https://doi.org/10.1109/TAC.2018.2790260.

Isermann, R., and M. Munchhof. 2011. *Identification of dynamic systems: An introduction with applications*. 1st ed. Berlin: Springer-Verlag.

Jiang, D., Z. Cai, Z. Liu, H. Peng, and Z. Wu. 2022. "An integrated tracking control approach based on reinforcement learning for a continuum robot in space capture missions." *J. Aerosp. Eng.* 35 (5): 1–10. https://doi.org/10.1061/(ASCE)AS.1943-5525.0001426.

Jiang, Y., and Z. P. Jiang. 2017. *Robust adaptive dynamic programming*. 1st ed. New York: Wiley.

Lewis, F., and D. R. Liu. 2013. *Reinforcement learning and approximate dynamic programming for feedback control*. 1st ed. New York: Wiley.

Liu, D., and Q. Wei. 2014. "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems." *IEEE Trans. Neural Networks Learn. Syst.* 25 (3): 621–634. https://doi.org/10.1109/TNNLS.2013.2281663.

Liu, Z., Y. Zhang, J. Liang, and H. Chen. 2022. "Application of the improved incremental nonlinear dynamic inversion in fixed-wing UAV flight tests." *J. Aerosp. Eng.* 35 (6): 1–13. https://doi.org/10.1061/(ASCE)AS.1943-5525.0001495.

Powell, W. B. 1977. "Approximate dynamic programming: Solving the curses of dimensionality." In *General systems yearbook*, 22. Hoboken, NJ: John Wiley & Sons.

Sharma, R., and G. W. P. York. 2018. "Near optimal finite-time terminal controllers for space trajectories via SDRE-based approach using dynamic programming." *Aerosp. Sci. Technol.* 75 (Apr): 128–138. https://doi.org/10.1016/j.ast.2017.12.022.

Sieberling, S., Q. P. Chu, and J. A. Mulder. 2010. "Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction." *J. Guid. Control Dyn.* 33 (6): 1732–1742. https://doi.org/10.2514/1.49978.

Sonneveldt, L. 2011. "Adaptive backstepping flight control for modern fighter aircraft." Ph.D. thesis, Dept. of Control and Operation, Delft Univ. of Technology.

Sun, B., and E. Kampen. 2021. "Incremental adaptive optimal control using incremental model-based global dual heuristic programming subject to partial observability." *Appl. Soft Comput.* 103 (May): 1–15.

Sutton, R. S., and A. G. Barto. 2014. *Reinforcement learning: An introduction*. 2nd ed. London: MIT Press.

Sutton, R. S., A. G. Barto, and R. J. Williams. 1992. "Reinforcement learning is direct adaptive optimal control." *IEEE Control Syst. Mag.* 12 (2): 19–22. https://doi.org/10.1109/37.126844.

Tamimi, A., L. Frank, and M. Khalaf. 2008. "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof." *IEEE Trans. Syst. Man Cybern.* 38 (4): 943–949. https://doi.org/10.1109/TSMCB.2008.926614.

Wang, Q., L. Gong, C. Dong, and K. Zhong. 2019a. "Morphing aircraft control based on switched nonlinear systems and adaptive dynamic programming." *Aerosp. Sci. Technol.* 93 (Oct): 1–16.

Wang, X. R., E. Kampen, and Q. P. Chu. 2019b. "Incremental sliding-mode fault-tolerant flight control." *J. Guid. Control Dyn.* 42 (2): 244–259. https://doi.org/10.2514/1.G003497.

Wang, X. R., E. Kampen, Q. P. Chu, and P. Lu. 2019c. "Stability analysis for incremental nonlinear dynamic inversion control." *J. Guid. Control Dyn.* 42 (5): 1116–1129. https://doi.org/10.2514/1.G003791.

Wang, X. R., T. Mkhoyan, and R. D. Breuker. 2021. "Nonlinear incremental control for flexible aircraft trajectory tracking and load alleviation." *Aerosp. Sci. Technol.* 27 (1): 1–9. https://doi.org/10.1016/j.ast.2012.05.006.

Wang, X. R., and S. H. Sun. 2022. "Incremental fault-tolerant control for a hybrid quad-plane UAV subjected to a complete rotor loss." *Aerosp. Sci. Technol.*, 1–9.

Wang, Y. C., W. S. Chen, S. X. Zhang, and J. W. Zhu. 2018. "Command-filtered incremental backstepping controller for small unmanned aerial vehicles." *J. Guid. Control Dyn.* 41 (4): 1–14.

Werbos, P. J. 1977. "Advanced forecasting methods for global crisis warning and models of intelligence." In *General systems yearbook*, 25–38. Denver: Annual Meetings of the Society for General Systems Research.

Zhou, Y., E. Kampen, and Q. P. Chu. 2015. "Incremental approximate dynamics programming for nonlinear flight control design." In *Proc., 3rd CEAS EuroGNC: Specialist Conf. on Guidance, Navigation and Control*, 33–40. Toulouse, France: Council of European Aerospace Societies.

Zhou, Y., E. Kampen, and Q. P. Chu. 2016. "An incremental approximate dynamic programming flight controller based on output feedback." In *Proc., AIAA Guidance, Navigation, and Control Conf.*, 1–16. Reston, VA: American Institute of Aeronautics and Astronautics.

Zhou, Y., E. van Kampen, and Q. P. Chu. 2018. "Incremental approximate dynamic programming for nonlinear adaptive tracking control with partial observability." *J. Guid. Control Dyn.* 41 (12): 2554–2567.

Zhou, Y., E. van Kampen, and Q. P. Chu. 2020. "Incremental model based online heuristic dynamic programming for nonlinear adaptive tracking control with partial observability." *Aerosp. Sci. Technol.* 105 (Oct): 1–14.