AIAA SciTech Forum
6-10 January 2025, Orlando, FL
AIAA SCITECH 2025 Forum

10.2514/6.2025-2081

Predictive Cost Adaptive Control of Fixed-Wing Aircraft without Prior Modeling

Riley J. Richards*, Juan A. Paredes †, and Dennis S. Bernstein‡
University of Michigan, Ann Arbor, Michigan, 48109

Autopilots for fixed-wing aircraft are typically designed based on linearized aerodynamic models consisting of stability and control derivatives obtained from wind-tunnel testing. The resulting local controllers are then pieced together using gain scheduling. For applications where the aerodynamics are unmodeled, the present paper proposes an autopilot based on predictive cost adaptive control (PCAC). As an indirect adaptive control extension of model predictive control, PCAC uses recursive least squares (RLS) with variable-rate forgetting for online, closed-loop system identification. At each time step, RLS-based system identification updates the coefficients of an input-output model whose order is specified by the user. For MPC, the receding-horizon optimization is performed by solving a quadratic program at each iteration. The present paper investigates the performance of PCAC for fixed-wing aircraft without using any prior aerodynamic model or offline data collection.

I. Introduction

A fundamental necessity for autonomous atmospheric flight vehicles is a reliable autopilot for controlling the attitude and flight path. For a fixed-wing vehicle, stability and control derivatives are typically determined through wind-tunnel testing or computational modeling over a range of Mach number, angle of attack, and sideslip angle. This modeling data is then used to develop an autopilot based on gain scheduling, feedback linearization, or dynamic inversion [1, 2]. In practice, however, the aerodynamics of an aircraft may be too expensive to model with high accuracy or may change due to atmospheric conditions, such as icing, as well as damage. This possibility motivates the need to develop an adaptive autopilot that can learn and respond to unknown, changing conditions [3–5]. The present paper introduces a novel adaptive autopilot for fixed-wing aircraft based on model predictive control (MPC). MPC is widely viewed as the most effective modern control technique, due to its ability to enforce state and control constraints in both linear and nonlinear systems through receding-horizon optimization [6–9].

As its name suggests, MPC requires a model for optimization. In the absence of a reliable model, data-driven techniques can be used to update the plant model during operation [10–12]. In effect, data-driven techniques perform closed-loop system identification, at least to a level of accuracy that is sufficient for the feedback controller to achieve desired closed-loop performance. The interplay between system identification and control is a fundamental, longstanding problem in control theory [13–17].

To develop an adaptive autopilot for fixed-wing aircraft, the present paper focuses on predictive cost adaptive control (PCAC) [18, 19], which is an indirect adaptive control extension of MPC. For online, closed-loop system identification, PCAC uses recursive least squares (RLS) with variable-rate forgetting [20–24]. At each time step, RLS-based system identification updates the coefficients of a SISO or MIMO input-output model, where the model order is a hyperparameter specified by the user. For MPC, the receding-horizon optimization can be performed by either the backward-propagating Riccati equation [6, 25] or quadratic programming. In this paper, all MPC optimization is performed using quadratic programming.

The objective of the present paper is to investigate the performance of PCAC as a data-driven autopilot for an aircraft with unmodeled kinematics, dynamics, and aerodynamics. In particular, PCAC is implemented as a cold-start indirect adaptive controller, where the plant model order is specified as a hyperparameter, but otherwise no plant model is assumed to be available. The identified model updated by RLS is linear, and thus it is suitable for modeling the aircraft dynamics near trim. In practice, an autopilot designed to operate over a wide range of flight conditions depends on gain scheduling of multiple linear controllers. The goal of this study is to investigate, via numerical experiments, the viability

*Ph.D. Candidate, Department of Aerospace Engineering, rileyric@umich.edu.
†Ph.D., Department of Aerospace Engineering, jparedes@umich.edu.
‡Professor, Department of Aerospace Engineering, dsbaero@umich.edu.

and potential performance of PCAC under conditions of high uncertainty, in effect, no prior modeling information, without the need for gain scheduling.

For this study, we first consider scenarios involving the longitudinal and lateral dynamics of a linearized fixed-wing aircraft model provided by Athena Vortex Lattice (AVL). Next, we consider the longitudinal dynamics of a nonlinear fixed-wing aircraft model provided in the MATLAB aerospace toolbox.

The contents of the paper are as follows. Section II introduces the predictive control problem. Section III reviews the PCAC formulation and algorithm. Section IV applies PCAC to a linear 6DOF aircraft model for longitudinal and lateral control, and Section V applies PCAC to a nonlinear 3DOF aircraft model for longitudinal control. Finally, Section VI presents conclusions and future research directions.

Notation: $\mathbf{z} \in \mathbb{C}$ denotes the Z-transform variable. $x_{(i)}$ denotes the ith component of $x \in \mathbb{R}^n$. $\operatorname{sprad}(A)$ denotes the spectral radius of $A \in \mathbb{R}^{n\times n}$. The symmetric matrix $P \in \mathbb{R}^{n\times n}$ is positive semidefinite (resp., positive definite) if all of its eigenvalues are nonnegative (resp., positive). $\operatorname{vec} X \in \mathbb{R}^{nm}$ denotes the vector formed by stacking the columns of $X \in \mathbb{R}^{n\times m}$, and $\otimes$ denotes the Kronecker product. I_n is the $n \times n$ identity matrix, and $0_{n\times m}$ is the $n \times m$ zeros matrix and $1_{n\times m}$ is the $n \times m$ ones matrix.

II. Statement of the Control Problem

To reflect the practical implementation of digital controllers for physical systems, we consider continuous-time dynamics under sampled-data control using discrete-time predictive controllers. In particular, we consider the control architecture shown in Figure 1, where $\mathcal{M}$ is the target continuous-time system, $t \ge 0$, $u(t) \in \mathbb{R}^m$ is the control, and $y(t) \in \mathbb{R}^p$ is the output of $\mathcal{M}$, which is sampled to produce the measurement $y_k \in \mathbb{R}^p$, which, for all $k \ge 0$, is given by

$$y_k \triangleq y(kT_\mathrm{s}), \tag{1}$$

where $T_\mathrm{s} > 0$ is the sample time.

The predictive controller, which is updated at each step k, is denoted by $G_{\mathrm{c},k}$. For all $k \ge 0$, let $y_{\mathrm{t},k} \triangleq C_{\mathrm{t},k} y_k \in \mathbb{R}^{p_\mathrm{t}}$ be the command following output, where $C_{\mathrm{t},k} \in \mathbb{R}^{p_\mathrm{t}\times p}$, and let $r_k \in \mathbb{R}^{p_\mathrm{t}}$ be the command. The inputs to $G_{\mathrm{c},k}$ are r_k, y_k, and $y_{\mathrm{t},k}$, and the output is the requested discrete-time control $u_{\mathrm{req},k} \in \mathbb{R}^m$. The predictive controller uses y_k for system identification, and r_k and $y_{\mathrm{t},k}$ trajectory command following. Since the response of a real actuator is subjected to hardware constraints, the implemented discrete-time control is

$$u_k \triangleq \gamma_k(\sigma(u_{\mathrm{req},k})), \tag{2}$$

where $\sigma\colon \mathbb{R}^m \to \mathbb{R}^m$ is the control magnitude saturation function

$$\sigma(u) \triangleq \begin{bmatrix} \bar\sigma(u_{(1)}) \\ \vdots \\ \bar\sigma(u_{(m)}) \end{bmatrix}, \tag{3}$$

where $\bar\sigma\colon \mathbb{R} \to \mathbb{R}$ is defined by

$$\bar\sigma(u) \triangleq \begin{cases} u_{\max}, & u > u_{\max}, \\ u, & u_{\min} \le u \le u_{\max}, \\ u_{\min}, & u < u_{\min}, \end{cases} \tag{4}$$

and $u_{\min}, u_{\max} \in \mathbb{R}$ are the lower and upper magnitude saturation levels, respectively, and $\gamma_k\colon \mathbb{R}^m \to \mathbb{R}^m$ is the control rate (move-size) saturation function

$$\gamma_k(u_k) \triangleq \begin{bmatrix} \bar\gamma_k(u_{k(1)}) \\ \vdots \\ \bar\gamma_k(u_{k(m)}) \end{bmatrix}, \tag{5}$$

where $\bar\gamma_k\colon \mathbb{R} \to \mathbb{R}$ is defined by

$$\bar\gamma_k(u_k) \triangleq \begin{cases} u_{k-1} + \Delta u_{\max}, & u_k - u_{k-1} > \Delta u_{\max}, \\ u_k, & \Delta u_{\min} \le u_k - u_{k-1} \le \Delta u_{\max}, \\ u_{k-1} + \Delta u_{\min}, & u_k - u_{k-1} < \Delta u_{\min}, \end{cases} \tag{6}$$

and $\Delta u_{\min}, \Delta u_{\max} \in \mathbb{R}$ are the lower and upper move-size saturation levels, respectively. The continuous-time control signal $u(t)$ applied to the structure is generated by applying a zero-order-hold operation to u_k, that is, for all $k \geq 0$, and, for all $t \in [kT_s, (k+1)T_s)$,

$$u(t) = u_k. \tag{7}$$

The objective of the predictive controller is to yield an input signal that minimizes the norm of the difference between the command following output and the command, that is, yield u_k such that $\sum_{k=0}^{\infty} ||y_{t,k} - r_k||_2$ is minimized.

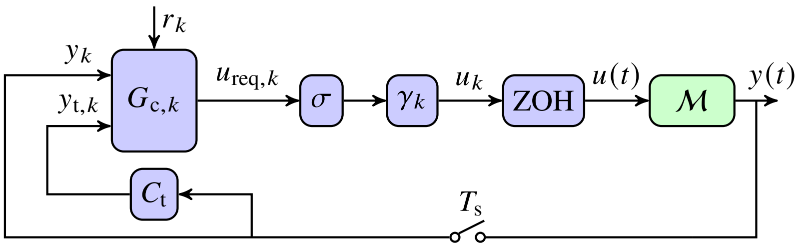

Fig. 1 *Sampled-data implementation of predictive controller for control of continuous-time system $\mathcal{M}$. All sample-and-hold operations are synchronous. The predictive controller $G_{c,k}$ generates the requested discrete-time control $u_{req,k} \in \mathbb{R}^m$ at each step k. The implemented discrete-time control is $u_k = \gamma_k(\sigma(u_{req,k}))$, where $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ represents control-magnitude saturation and $\gamma_k : \mathbb{R}^m \to \mathbb{R}^m$ represents move-size saturation. The resulting continuous-time control $u(t)$ is generated by applying a zero-order-hold operation to u_k. For this work, $\mathcal{M}$ represents a fixed-wing aircraft simulation model.*

III. Review of Predictive Cost Adaptive Control

As shown in Figure 2, PCAC combines online identification with output-feedback MPC. The PCAC algorithm is presented in this section. Subsection III.A describes the technique used for online identification, namely, RLS with variable-rate forgetting based on the F-test [24]. Subsection III.B presents the block observable canonical form (BOCF), which is used to represent the input-output dynamics model as a state space model whose state is given explicitly in terms of inputs, outputs, and model-coefficient estimates. Subsection III.C reviews the MPC technique for receding-horizon optimization.

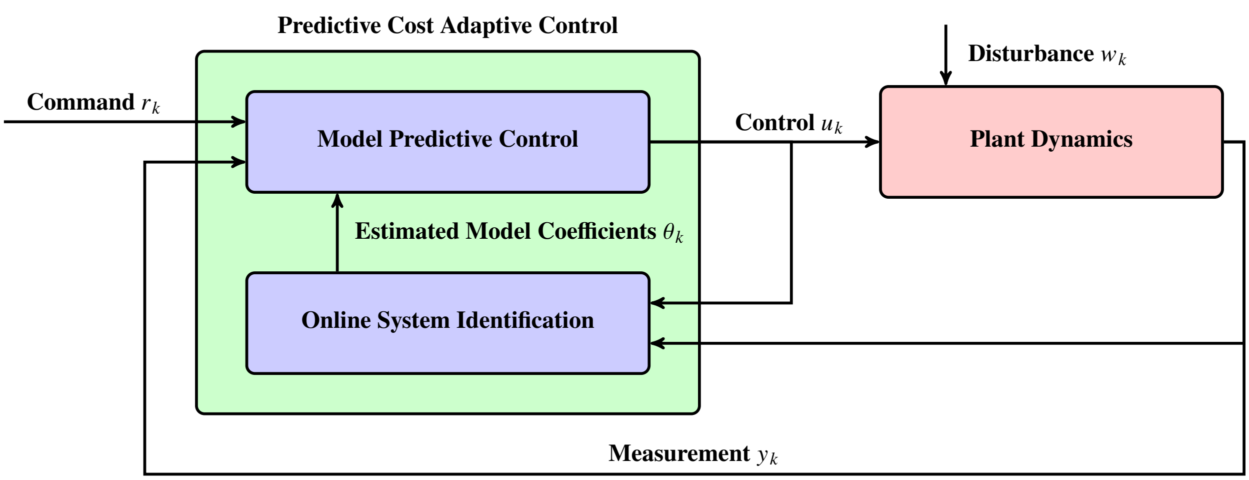

Fig. 2 *PCAC block diagram. The online, closed-loop system identification is based on recursive least squares (RLS) with variable-rate forgetting (VRF). The model predictive control (MPC) algorithm, which is based on quadratic programming (QP), uses the estimated model coefficients θ_k to form a block-observable canonical form (BOCF) state-space model, which is used by QP to determine the control input u_k.*

A. Online Identification Using Recursive Least Squares with Variable-Rate Forgetting Based on the F-Test

Let $\hat{n} \geq 0$ and, for all $k \geq 0$, let $F_{\mathrm{m},1,k}, \ldots, F_{\mathrm{m},\hat{n},k} \in \mathbb{R}^{p\times p}$ and $G_{\mathrm{m},1,k}, \ldots, G_{\mathrm{m},\hat{n},k} \in \mathbb{R}^{p\times m}$ be the coefficient matrices to be estimated using RLS. Furthermore, let $\hat{y}_k \in \mathbb{R}^p$ be an estimate of y_k defined by

$$\hat{y}_k \triangleq -\sum_{i=1}^{\hat{n}} F_{\mathrm{m},i,k} y_{k-i} + \sum_{i=1}^{\hat{n}} G_{\mathrm{m},i,k} u_{k-i}, \tag{8}$$

where

$$y_{-\hat{n}} = \cdots = y_{-1} = 0, \tag{9}$$

$$u_{-\hat{n}} = \cdots = u_{-1} = 0. \tag{10}$$

Using the identity $\mathrm{vec}(XY) = (Y^\mathsf{T} \otimes I)\mathrm{vec}X$, it follows from (8) that, for all $k \geq 0$,

$$\hat{y}_k = \phi_k \theta_k, \tag{11}$$

where

$$\theta_k \triangleq \begin{bmatrix} \theta_{F_\mathrm{m},k}^\mathrm{T} & \theta_{G_\mathrm{m},k}^\mathrm{T} \end{bmatrix}^\mathrm{T} \in \mathbb{R}^{\hat{n}p(m+p)}, \tag{12}$$

$$\theta_{F_\mathrm{m},k} \triangleq \mathrm{vec} \begin{bmatrix} F_{\mathrm{m},1,k} & \cdots & F_{\mathrm{m},\hat{n},k} \end{bmatrix} \in \mathbb{R}^{\hat{n}p^2}, \tag{13}$$

$$\theta_{G_\mathrm{m},k} \triangleq \mathrm{vec} \begin{bmatrix} G_{\mathrm{m},1,k} & \cdots & G_{\mathrm{m},\hat{n},k} \end{bmatrix} \in \mathbb{R}^{\hat{n}pm}, \tag{14}$$

$$\phi_k \triangleq \begin{bmatrix} -y_{k-1}^\mathrm{T} & \cdots & -y_{k-\hat{n}}^\mathrm{T} & u_{k-1}^\mathrm{T} & \cdots & u_{k-\hat{n}}^\mathrm{T} \end{bmatrix} \otimes I_p \in \mathbb{R}^{p\times\hat{n}p(m+p)}. \tag{15}$$

To determine the update equations for θ_k, for all $k \geq 0$, define $e_k \colon \mathbb{R}^{\hat{n}p(m+p)} \to \mathbb{R}^p$ by

$$e_k(\bar{\theta}) \triangleq y_k - \phi_k \bar{\theta}, \tag{16}$$

where $\bar{\theta} \in \mathbb{R}^{\hat{n}p(m+p)}$. Using (11), the *identification error* at step k is defined by

$$e_k(\theta_k) = y_k - \hat{y}_k. \tag{17}$$

For all $k \geq 0$, the RLS cumulative cost $J_k \colon \mathbb{R}^{\hat{n}p(m+p)} \to [0, \infty)$ is defined by [20]

$$J_k(\bar{\theta}) \triangleq \sum_{i=0}^{k} \frac{\rho_i}{\rho_k} e_i^\mathrm{T}(\bar{\theta}) e_i(\bar{\theta}) + \frac{1}{\rho_k}(\bar{\theta} - \theta_0)^\mathrm{T} \psi_0^{-1} (\bar{\theta} - \theta_0), \tag{18}$$

where $\psi_0 \in \mathbb{R}^{\hat{n}p(m+p)\times\hat{n}p(m+p)}$ is positive definite, $\theta_0 \in \mathbb{R}^{\hat{n}p(m+p)}$ is the initial estimate of the coefficient vector, and, for all $i \geq 0$,

$$\rho_i \triangleq \prod_{j=0}^{i} \lambda_j^{-1}. \tag{19}$$

For all $j \geq 0$, the parameter $\lambda_j \in (0, 1]$ is the forgetting factor defined by $\lambda_j \triangleq \beta_j^{-1}$, where

$$\beta_j \triangleq \begin{cases} 1, & j < \tau_\mathrm{d}, \\ 1 + \eta\bar{\beta}_j, & j \geq \tau_\mathrm{d}, \end{cases} \tag{20}$$

$$\bar{\beta}_j \triangleq g(e_{j-\tau_\mathrm{d}}(\theta_{j-\tau_\mathrm{d}}), \ldots, e_j(\theta_j)) \mathbf{1}\big(g(e_{j-\tau_\mathrm{d}}(\theta_{j-\tau_\mathrm{d}}), \ldots, e_j(\theta_j))\big), \tag{21}$$

and $\tau_\mathrm{d} > p$, $\eta > 0$, $\mathbf{1} \colon \mathbb{R} \to \{0, 1\}$ is the unit step function, and g is a function of past RLS identification errors which is given by (10) in [24] in the case where $p = 1$ and (13) in [24] in the case where $p > 1$. Note that g includes forgetting terms based on the inverse cumulative distribution function of the F-distribution and depends on τ_d, $\tau_\mathrm{n} \in [p, \tau_\mathrm{d})$, and *significance level* $\alpha_F \in (0, 1]$.

Finally, for all $k \geq 0$, the unique global minimizer of J_k is given by [20]

$$\theta_{k+1} = \theta_k + \psi_{k+1} \phi_k^\mathrm{T} (y_k - \phi_k \theta_k), \tag{22}$$

where

$$\psi_{k+1} \triangleq \beta_k \psi_k - \beta_k \psi_k \phi_k^\mathrm{T} (\tfrac{1}{\beta_k} I_p + \phi_k \psi_k \phi_k^\mathrm{T})^{-1} \phi_k \psi_k, \tag{23}$$

and ψ_0 is the performance-regularization weighting in (18). Additional details concerning RLS with forgetting based on the F-distribution are given in [24].

B. Input-Output Model and the Block Observable Canonical Form

Considering the estimate $\hat{y}_k$ of y_k given by (8), it follows that, for all $k \geq 0$,

$$y_k \approx -\sum_{i=1}^{\hat{n}} F_{\mathrm{m},i,k} y_{k-i} + \sum_{i=1}^{\hat{n}} G_{\mathrm{m},i,k} u_{k-i}. \tag{24}$$

Viewing (24) as an equality, it follows that, for all $k \geq 0$, the BOCF state-space realization of (24) is given by [26]

$$x_{\mathrm{m},k+1} = A_{\mathrm{m},k} x_{\mathrm{m},k} + B_{\mathrm{m},k} u_k, \tag{25}$$

$$y_k = C_\mathrm{m} x_{\mathrm{m},k}, \tag{26}$$

where

$$A_{\mathrm{m},k} \triangleq \begin{bmatrix} -F_{\mathrm{m},1,k+1} & I_p & \cdots & \cdots & 0_{p\times p} \\ -F_{\mathrm{m},2,k+1} & 0_{p\times p} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0_{p\times p} \\ \vdots & \vdots & & \ddots & I_p \\ -F_{\mathrm{m},\hat{n},k+1} & 0_{p\times p} & \cdots & \cdots & 0_{p\times p} \end{bmatrix} \in \mathbb{R}^{\hat{n}p\times\hat{n}p}, \tag{27}$$

$$B_{\mathrm{m},k} \triangleq \begin{bmatrix} G_{\mathrm{m},1,k+1} \\ G_{\mathrm{m},2,k+1} \\ \vdots \\ G_{\mathrm{m},\hat{n},k+1} \end{bmatrix} \in \mathbb{R}^{\hat{n}p\times m}, \tag{28}$$

$$C_\mathrm{m} \triangleq \begin{bmatrix} I_p & 0_{p\times p} & \cdots & 0_{p\times p} \end{bmatrix} \in \mathbb{R}^{p\times\hat{n}p}, \tag{29}$$

and

$$x_{\mathrm{m},k} \triangleq \begin{bmatrix} x_{\mathrm{m},k(1)} \\ \vdots \\ x_{\mathrm{m},k(\hat{n})} \end{bmatrix} \in \mathbb{R}^{\hat{n}p}, \tag{30}$$

where

$$x_{\mathrm{m},k(1)} \triangleq y_k, \tag{31}$$

and, for all $j = 2, \ldots, \hat{n}$,

$$x_{\mathrm{m},k(j)} \triangleq -\sum_{i=1}^{\hat{n}-j+1} F_{\mathrm{m},i+j-1,k+1} y_{k-i} + \sum_{i=1}^{\hat{n}-j+1} G_{\mathrm{m},i+j-1,k+1} u_{k-i}. \tag{32}$$

Note that multiplying both sides of (25) by C_m and using (26)–(32) implies that, for all $k \geq 0$,

$$\begin{aligned} y_{k+1} &= C_\mathrm{m} x_{\mathrm{m},k+1} \\ &= C_\mathrm{m}(A_{\mathrm{m},k} x_{\mathrm{m},k} + B_{\mathrm{m},k} u_k) \\ &= -F_{\mathrm{m},1,k+1} x_{\mathrm{m},k(1)} + x_{\mathrm{m},k(2)} + G_{\mathrm{m},1,k+1} u_k \\ &= -F_{\mathrm{m},1,k+1} y_k - \sum_{i=1}^{\hat{n}-1} F_{\mathrm{m},i+1,k+1} y_{k-i} + \sum_{i=1}^{\hat{n}-1} G_{\mathrm{m},i+1,k+1} u_{k-i} + G_{\mathrm{m},1,k+1} u_k \\ &= -\sum_{i=1}^{\hat{n}} F_{\mathrm{m},i,k+1} y_{k+1-i} + \sum_{i=1}^{\hat{n}} G_{\mathrm{m},i,k+1} u_{k+1-i}, \end{aligned} \tag{33}$$

which is approximately equivalent to (24) with k in (24) replaced by $k+1$.

C. Model Predictive Control (MPC)

Let $\ell \geq 1$ be the horizon and, for all $k \geq 0$ and all $i = 1, \ldots, \ell$, let $x_{\mathrm{m},k|i} \in \mathbb{R}^{\hat{n}p}$ be the i-step predicted state, $y_{\mathrm{m},k|i} \in \mathbb{R}^{p}$ be the i-step predicted output, and $u_{k|i} \in \mathbb{R}^{m}$ be the i-step predicted control. Then, the ℓ-step predicted output of (26) for a sequence of ℓ future controls is given by

$$Y_{\ell,k|1} = \Gamma_{\ell,k} x_{\mathrm{m},k|1} + T_{\ell,k} U_{\ell,k|1}, \tag{34}$$

where

$$Y_{\ell,k|1} \triangleq \begin{bmatrix} y_{\mathrm{m},k|1} \\ \vdots \\ y_{\mathrm{m},k|\ell} \end{bmatrix} \in \mathbb{R}^{\ell p}, \quad U_{\ell,k|1} \triangleq \begin{bmatrix} u_{k|1} \\ \vdots \\ u_{k|\ell} \end{bmatrix} \in \mathbb{R}^{\ell m}, \tag{35}$$

$$\Gamma_{\ell,k} \triangleq \begin{bmatrix} C_{\mathrm{m}}^{\mathrm{T}} & (C_{\mathrm{m}} A_{\mathrm{m},k})^{\mathrm{T}} & \cdots & (C_{\mathrm{m}} A_{\mathrm{m},k}^{\ell-1})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{\ell p \times \hat{n} p}, \tag{36}$$

$$T_{\ell,k} \triangleq \begin{bmatrix} 0_{p\times m} & \cdots & \cdots & \cdots & \cdots & 0_{p\times m} \\ H_{k,1} & 0_{p\times m} & \cdots & \cdots & \cdots & 0_{p\times m} \\ H_{k,2} & H_{k,1} & 0_{p\times m} & \cdots & \cdots & 0_{p\times m} \\ H_{k,3} & H_{k,2} & H_{k,1} & 0_{p\times m} & \cdots & 0_{p\times m} \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0_{p\times m} \\ H_{k,\ell-1} & H_{k,\ell-2} & H_{k,\ell-3} & \cdots & H_{k,1} & 0_{p\times m} \end{bmatrix} \in \mathbb{R}^{\ell p \times \ell m}, \tag{37}$$

where $H_{k,i} \triangleq C_{\mathrm{m}} A_{\mathrm{m},k}^{i-1} B_{\mathrm{m},k} \in \mathbb{R}^{p\times m}$ for all $i = 1, \ldots, \ell - 1$.

Let $\mathcal{R}_{\ell,k} \triangleq \begin{bmatrix} r_{k+1}^{\mathrm{T}} & \cdots & r_{k+\ell}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{\ell p_{\mathrm{t}}}$ be a vector composed of ℓ future commands, let $y_{\mathrm{m,t},k|i} \triangleq C_{\mathrm{t}} y_{\mathrm{m},k|i} \in \mathbb{R}^{p_{\mathrm{t}}}$ be the i-step predicted command-following output, let $Y_{\mathrm{t},\ell,k|1} \triangleq \begin{bmatrix} y_{\mathrm{m,t},k|1}^{\mathrm{T}} & \cdots & y_{\mathrm{m,t},k|\ell}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = C_{\mathrm{t},\ell} Y_{\ell,k|1} \in \mathbb{R}^{\ell p_{\mathrm{t}}}$, where $C_{\mathrm{t},\ell} \triangleq I_{\ell} \otimes C_{\mathrm{t}} \in \mathbb{R}^{\ell p_{\mathrm{t}} \times \ell p}$, and define

$$\Delta U_{\ell,k|1} \triangleq \begin{bmatrix} u_{k|1} - u_k \\ u_{k|2} - u_{k|1} \\ \vdots \\ u_{k|\ell} - u_{k|\ell-1} \end{bmatrix} \in \mathbb{R}^{\ell m}. \tag{38}$$

Then, the receding horizon optimization problem is given by

$$\min_{U_{\ell,k|1}} \left(Y_{\mathrm{t},\ell,k|1} - \mathcal{R}_{\ell,k}\right)^{\mathrm{T}} Q \left(Y_{\mathrm{t},\ell,k|1} - \mathcal{R}_{\ell,k}\right) + \Delta U_{\ell,k|1}^{\mathrm{T}} R \Delta U_{\ell,k|1}, \tag{39}$$

subject to

$$U_{\min} \leq U_{\ell,k|1} \leq U_{\max} \tag{40}$$

$$\Delta U_{\min} \leq \Delta U_{\ell,k|1} \leq \Delta U_{\max}, \tag{41}$$

where $Q \triangleq \begin{bmatrix} \bar{Q} & 0_{(\ell-1)p_{\mathrm{t}}\times p_{\mathrm{t}}} \\ 0_{p_{\mathrm{t}}\times(\ell-1)p_{\mathrm{t}}} & \bar{P} \end{bmatrix} \in \mathbb{R}^{\ell p_{\mathrm{t}} \times \ell p_{\mathrm{t}}}$ is the positive-definite output weighting, $\bar{Q} \in \mathbb{R}^{(\ell-1)p_{\mathrm{t}}\times(\ell-1)p_{\mathrm{t}}}$ is the positive-definite cost-to-go output weighting, $\bar{P} \in \mathbb{R}^{p_{\mathrm{t}}\times p_{\mathrm{t}}}$ is the positive-definite terminal output weighting, $R \in \mathbb{R}^{\ell m \times \ell m}$ is the positive definite control move-size weight, $U_{\min} \triangleq 1_{\ell} \otimes u_{\min} \in \mathbb{R}^{\ell m}$, $U_{\max} \triangleq 1_{\ell} \otimes u_{\max} \in \mathbb{R}^{\ell m}$, $\Delta U_{\min} \triangleq 1_{\ell} \otimes \Delta u_{\min} \in \mathbb{R}^{\ell m}$, and $\Delta U_{\max} \triangleq 1_{\ell} \otimes \Delta u_{\max} \in \mathbb{R}^{\ell m}$.

In summary, at each time step, online identification is performed to find input-output model coefficients θ_{k+1}, which are then used to create a state space realization $\left(A_{\mathrm{m},k}, B_{\mathrm{m},k}, C_{\mathrm{m}}\right)$. Then, the state-space realization is used in a receding horizon optimization problem to solve for the ℓ-step controls $U_{\ell,k|1}$. The control input for the next step is then given by $u_{k|1}$, and the rest of the components of $U_{\ell,k|1}$ are discarded.

IV. Adaptive Control of a Linearized 6DOF Aircraft Model

In this section, the Athena Vortex Lattice (AVL) Supra model, a linearized 6DOF fixed-wing RC aircraft model, is considered for both longitudinal and lateral control. The first example uses altitude, elevation angle, and bank angle measurements for altitude and bank-angle command following using the elevator and aileron. The second example uses altitude, elevation, bank, and azimuth measurements for altitude and azimuth-angle command following using all three control surfaces. Note that all numerical examples in this section use a sampling rate of $T_s = 0.05$ sec/step and the aircraft model is initialized at an altitude $h_0 = 500$ m and airspeed $V_0 = 10$ m/s with all other initial model parameters set to 0.

A. Altitude and Bank-Angle Command Following

For this example, two PCAC loops are used to follow altitude and bank-angle commands. The longitudinal loop uses the altitude h and elevation angle Θ to identify the longitudinal dynamics and to specify the elevator deflection δ_e. Note the longitudinal loop only follows altitude, but utilizes the elevation angle measurement to improve performance. The lateral loop uses the bank angle Φ to identify the lateral dynamics and to specify the aileron deflection δ_a. This control architecture is illustrated in Figure 3.

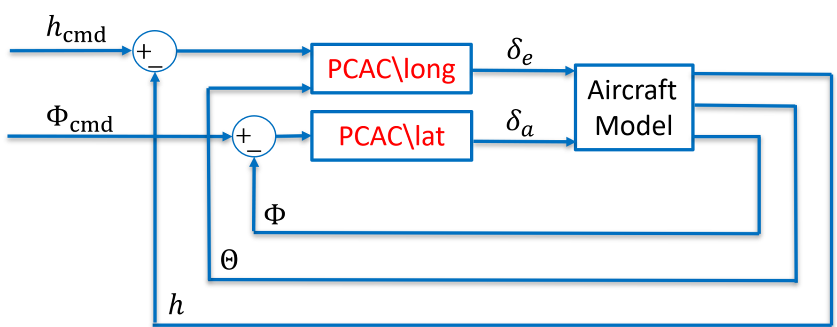

Fig. 3 *Command-following block diagram for altitude h and bank angle Φ. This architecture uses two PCAC controllers, namely, PCAC/long for the longitudinal dynamics and PCAC/lat for the lateral dynamics. Note that PCAC/long uses the elevation angle Θ for system identification.*

PCAC/long is initialized with $\psi_{\rm long,0} = 10^5 I_{36}$, $\theta_{\rm long,0} = 0.01\mathbf{1}_{36\times1}$, $\hat{n} = 6$, $\ell = 40$, $\bar{Q} = 0.1 I_{39}$, $\bar{P} = 0.1$, $R = 0.1$, $|\Delta\delta_e| \le \Delta\delta_{e,\rm max} = 0.5$ deg/step, and $|\delta_e| \le \delta_{e,\rm max} = 10$ deg with no forgetting. PCAC/lat is initialized with $\psi_{\rm lat,0} = 10^5 I_{36}$, $\theta_{\rm lat,0} = 0.01\mathbf{1}_{36\times1}$, $\hat{n} = 6$, $\ell = 40$, $\bar{Q} = 0.1 I_{39}$, $\bar{P} = 0.1$, $R = 0.01$, $|\Delta\delta a| \le \Delta\delta_{a,\rm max} = 0.5$ deg/step, and $|\delta_a| \le \delta_{a,\rm max} = 2$ deg with no forgetting. The resulting flight path data is shown in Figure 4. An altitude ramp command is given for $t \in [25, 50]$ s, followed by a bank-angle ramp command for $t \in [50, 60]$ s. Furthermore, altitude and bank-angle ramp commands are given simultaneously for $t \in [75, 85]$ s. As shown in Figure 4, PCAC follows the altitude and bank-angle commands with a slight overshoot in the altitude command following while satisfying the control constraints.

Figure 5 shows the estimated coefficient vectors $\theta_{\rm long}$ and $\theta_{\rm lat}$, where $\theta_{\rm long}$ is the estimated coefficient vector of PCAC/long and $\theta_{\rm lat}$ is the estimated coefficient vector of PCAC/lat. ◇

B. Altitude and Azimuth-Angle Command Following

For this example, two PCAC loops are used to follow altitude and azimuth-angle commands. The longitudinal loop uses the altitude h and elevation angle Θ to identify the longitudinal dynamics and to specify the elevator deflection δ_e. The lateral loop uses the bank angle Φ and azimuth angle Ψ to identify the lateral dynamics and to specify the aileron and rudder deflections δ_a and δ_r. Note the longitudinal and lateral loops measure elevation and bank angles, respectively, to improve performance and are not part of the command following. This control architecture is illustrated in Figure 6.

The same PCAC/long parameters from Section IV.A are used to initialize the PCAC/long loop in this example. PCAC/lat is initialized with $\psi_{\rm lat,0} = 2 \times 10^7 I_{48}$, $\theta_{\rm lat,0} = 0.01\mathbf{1}_{48\times1}$, $\hat{n} = 6$, $\ell = 50$, $\bar{Q} = 60 I_{49}$, $\bar{P} = 60$, $R = 1.5 I_{2\times2}$, $[|\Delta\delta_a|,\ |\Delta\delta_r|]^{\rm T} \le [\Delta\delta_{a,\rm max},\ \Delta\delta_{r,\rm max}]^{\rm T} = [0.2,\ 0.2]^{\rm T}$ deg/step, and $[|\delta_a|,\ |\delta_r|]^{\rm T} \le [\delta_{a,\rm max},\ \delta_{r,\rm max}]^{\rm T} = [5,\ 5]^{\rm T}$ deg with no forgetting. The resulting flight path data is shown in Figure 7. Altitude ramp commands are given for $t \in [20, 40]$ s and $t \in [60, 80]$ s while azimuth-angle ramp commands are given for $t \in [20, 45]$ s and $t \in [55, 80]$ s. As shown

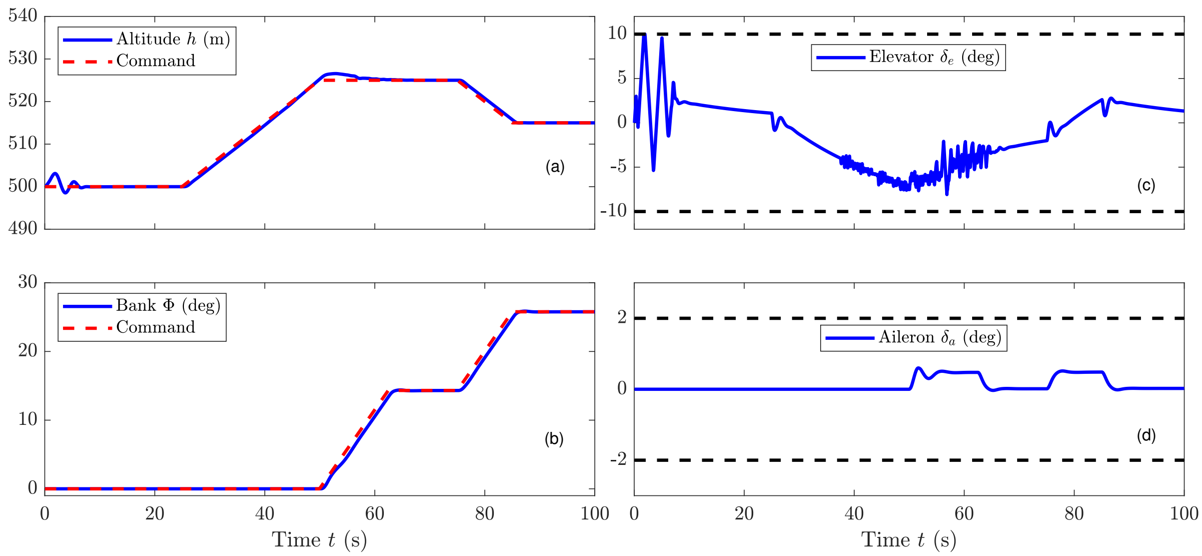

Fig. 4 *Altitude and bank-angle command following and control surface deflection. (a) compares the command* h_{cmd} *and the simulated* h*. (b) compares the command* Φ_{cmd} *and the simulated* Φ*. Note that learning takes place for the longitudinal dynamics for* $t \in [0, 5]$ *s causing the excitation seen in (a). (c) shows the elevator deflection* δ_e *(blue) with the constraints on* $|\delta_e|$ *(dashed black). (d) shows the aileron deflection* δ_a *(blue) with the constraints on* $|\delta_a|$ *(dashed black).*

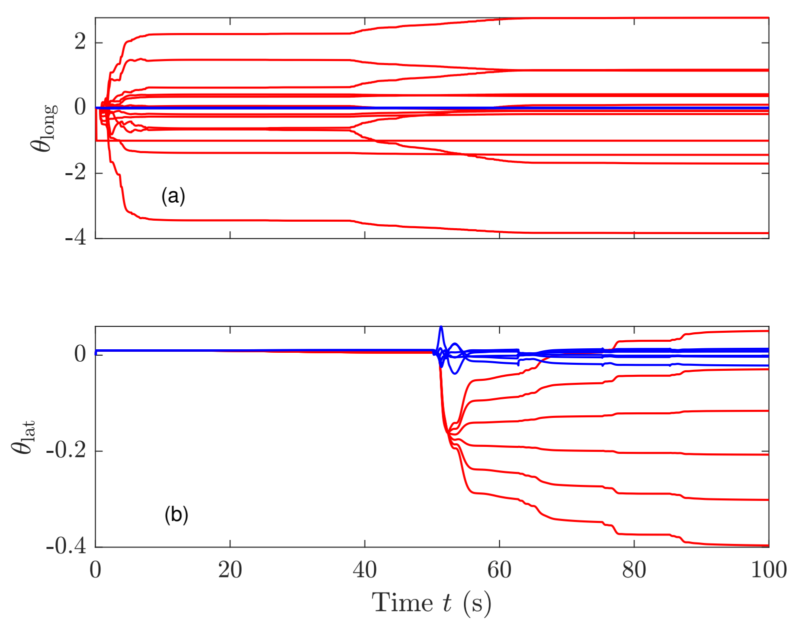

Fig. 5 *Estimated coefficient vectors for the altitude command-following loop* θ_{long} *(a) and the bank-angle command-following loop* θ_{lat} *(b). Note* $\theta_{F_{\text{m}},k}$ *for both loops is denoted by red lines while* $\theta_{G_{\text{m}},k}$ *for both loops is denoted by blue lines. No prior modeling information is assumed, where* $\theta_{\text{long},0} = \theta_{\text{lat},0} = 0.01\boldsymbol{I}_{36\times 1}$.

in Figure 7, PCAC follows the altitude and azimuth-angle commands with slight overshoot in the altitude command following while satisfying the control constraints.

Figure 8 shows the estimated coefficient vectors θ_{long} and θ_{lat}, where θ_{long} is the estimated coefficient vector of PCAC/long and θ_{lat} is the estimated coefficient vector of PCAC/lat. ◇

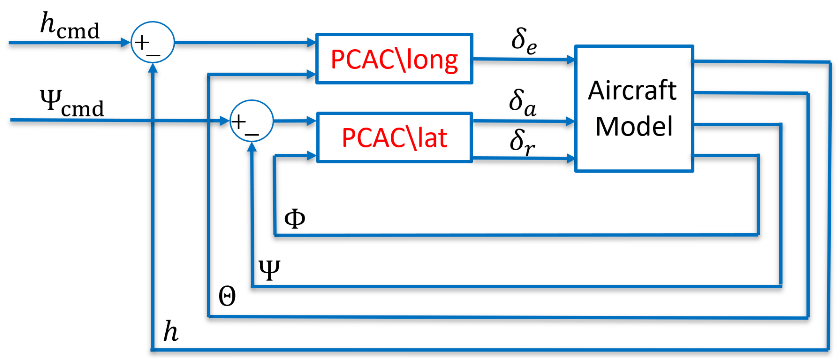

Fig. 6 *Command-following block diagram for altitude h and azimuth angle Ψ. This architecture uses two PCAC controllers, namely, PCAC/long for the longitudinal dynamics and PCAC/lat for the lateral dynamics. Note that PCAC/long uses the elevation angle Θ for system identification and PCAC/lat uses the bank angle Φ for system identification.*

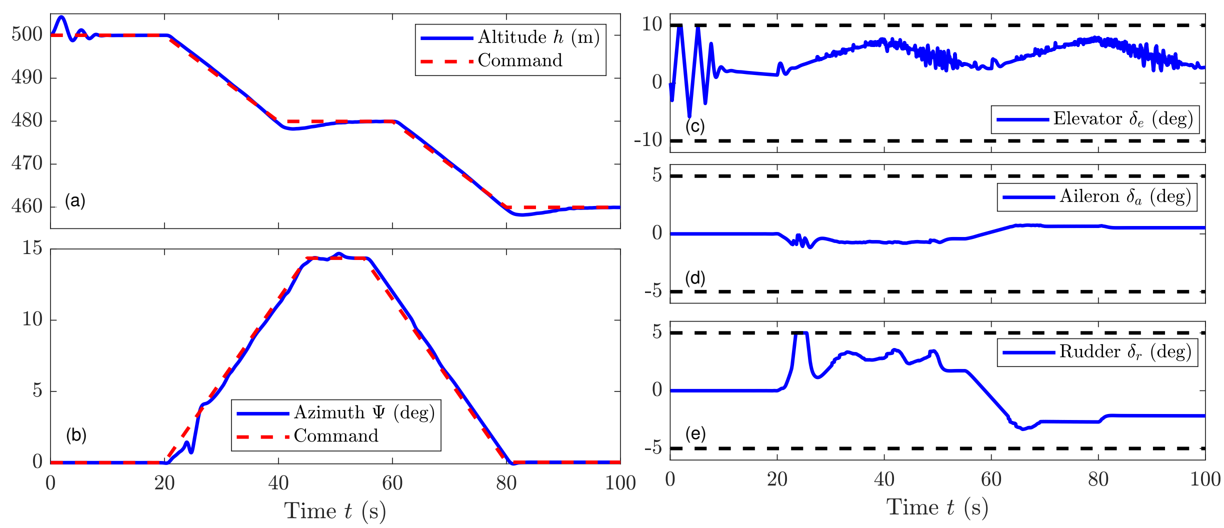

Fig. 7 *Altitude and azimuth-angle command following and control surface deflection. (a) compares the command $h_{\rm cmd}$ and the simulated h. (b) compares the command $\Psi_{\rm cmd}$ and the simulated Ψ. Note that learning takes place for the longitudinal dynamics for $t \in [0, 5]$ s causing the excitation seen in (a), while learning for the lateral dynamics takes place for $t \in [20, 25]$ s, causing the excitation seen in (b). (c) shows the elevator deflection δ_e (blue) with the constraint on $|\delta_e|$ (dashed black). (d) shows the aileron deflection δ_a (blue) with the constraint on $|\delta_a|$ (dashed black). (e) shows the rudder deflection δ_r (blue) with the constraint on $|\delta_r|$ (dashed black).*

V. Adaptive Control of a Nonlinear 3DOF Aircraft Model

In this section, a nonlinear 3DOF fixed-wing passenger aircraft model provided by Matlab's aerospace toolbox and featured in [27] is considered for longitudinal control. The first example uses altitude and angle of attack measurements for altitude command following using the elevator. The second example uses altitude, angle of attack, and airspeed measurements for altitude and airspeed command following using the elevator and throttle. Note that all simulations in this section use a sampling rate of $T_s = 0.05$ sec/step and the aircraft model is initialized at an altitude $h_0 = 2000$ m and airspeed $V_0 = 85$ m/s with all other initial model parameters set to 0. Both examples use forgetting for altitude command following with $\eta = 0.025$, $\tau_{\rm n} = 40$, $\tau_{\rm d} = 200$, and $\alpha_F = 0.001$.

A. Altitude Command Following

For this example, a single PCAC loop is used to follow altitude commands. The loop uses the altitude h and angle of attack α to identify the longitudinal dynamics and generate an elevator deflection δe. Note the loop only follows altitude, but utilizes the angle of attack measurement to improve performance. This control architecture is illustrated in Figure 9.

PCAC/long is initialized with $\psi_{\rm long,0} = 10^4 I_{60}$, $\theta_{\rm long,0} = 0.01\mathbf{1}_{60\times1}$, $\hat{n} = 10$, $\ell = 40$, $\bar{Q} = 0.01 I_{39}$, $\bar{P} = 500$, $R = 0.01$,

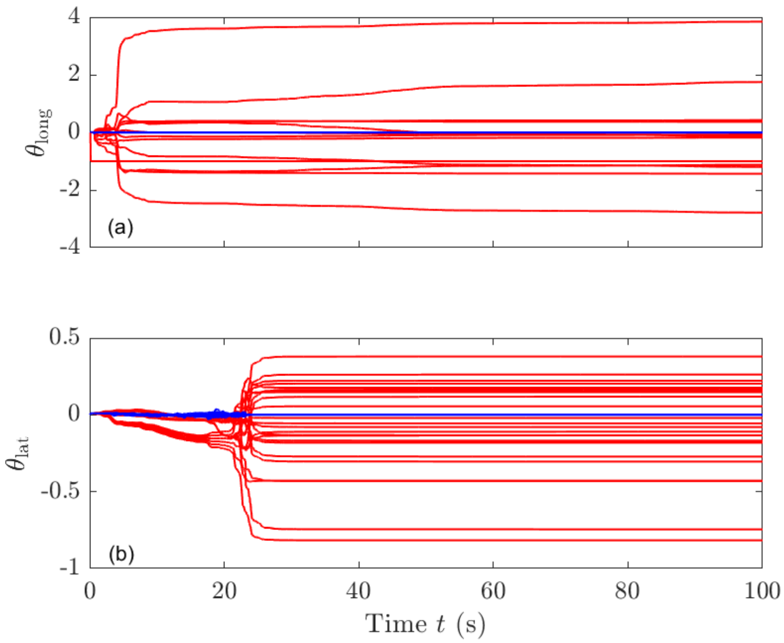

Fig. 8 *Estimated coefficient vectors for the altitude command-following loop θ_{long} (a) and the azimuth-angle command-following loop θ_{lat} (b). Note $\theta_{F_{\text{m}},k}$ for both loops is denoted by red lines while $\theta_{G_{\text{m}},k}$ for both loops is denoted by blue lines. No prior modeling information is assumed, where $\theta_{\text{long},0} = 0.01\boldsymbol{1}_{39\times1}$ and $\theta_{\text{lat},0} = 0.01\boldsymbol{1}_{49\times1}$.*

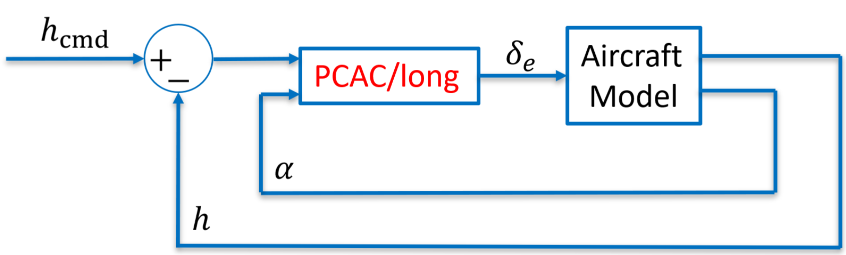

Fig. 9 *Command-following block diagram for altitude h. Note that the loop uses the angle of attack α for system identification.*

$|\Delta\delta_e| \leq \Delta\delta_{e,\max} = 1$ deg/step, and $|\delta_e| \leq \delta_{e,\max} = 5$ deg. The resulting flight path data is shown in Figure 10. Altitude ramp commands are given for $t \in [30, 50]$ s and $t \in [75, 90]$ s. As shown in Figure 10, PCAC follows the altitude commands with slight overshoot and undershoot while satisfying the control constraints. Figure 11 shows the estimated

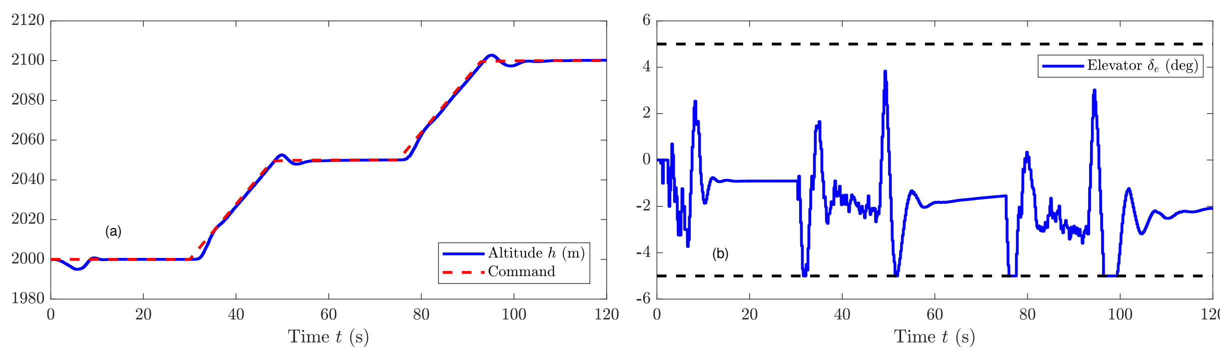

Fig. 10 *Altitude command following and control surface deflection. (a) compares the command h_{cmd} and the simulated h. Note that learning takes place for $t \in [0, 5]$ s causing the initial excitation seen in the plot. (b) shows the elevator deflection δ_e (blue) with the constraint on $|\delta_e|$ (dashed black).*

coefficient vector $\theta_{\rm long}$. ◇

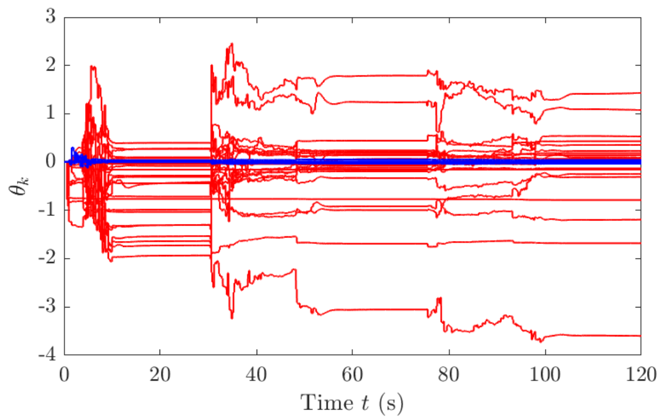

Fig. 11 *Estimated coefficient vector $\theta_{\rm long}$ for altitude command following. No prior modeling information is assumed, where $\theta_{\rm long,0} = 0.01\mathbf{1}_{60\times1}$. Note $\theta_{F_{\rm m},k}$ is denoted by red lines while $\theta_{G_{\rm m},k}$ is denoted by blue lines.*

B. Altitude and Airspeed Command Following

For this example, two PCAC loops are used to follow altitude and airspeed commands. The longitudinal loop uses the altitude h and angle of attack α to identify the longitudinal dynamics and to specify the elevator deflection δe. Note the longitudinal loop only follows altitude, but utilizes the angle of attack measurement to improve performance. The airspeed loop uses the airspeed V to identify the throttle dynamics and generate a change in throttle ΔT. Note that the throttle parameter $T \in [0, 1]$ has the constant input 0.5. This control architecture is illustrated in Figure 12.

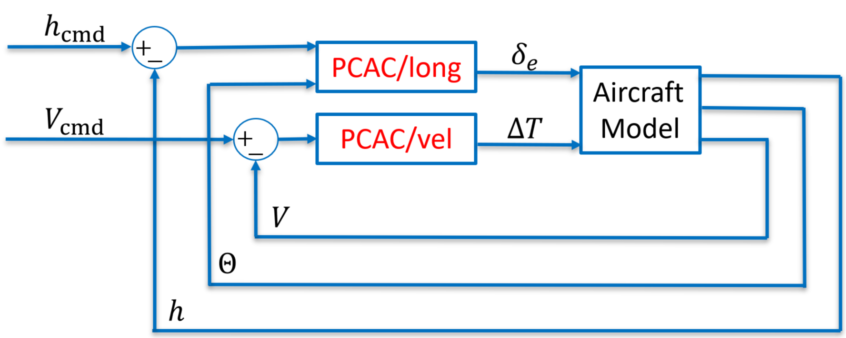

Fig. 12 *Command-following block diagram for altitude h and airspeed V. This architecture uses two PCAC controllers, namely, PCAC/long for the longitudinal dynamics and PCAC/vel for the throttle dynamics. Note that PCAC/long uses the angle of attack α for system identification.*

The same PCAC/long parameters from Section V.A are used to initialize the PCAC/long altitude command-following loop. PCAC/vel is initialized with $\psi_{\rm vel,0} = 10^5 I_8$, $\theta_{\rm vel,0} = 0.01\mathbf{1}_{8\times1}$, $\hat{n} = 4$, $\ell = 30$, $\bar{Q} = 0.0001 I_{29}$, $\bar{P} = 0.01$, $R = 0.01$, $|\Delta(\Delta T)| \le \Delta(\Delta T_{\rm max}) = 0.1$ 1/step, and $|\Delta T| \le \Delta T_{\rm max} = 0.25$ with no forgetting. The resulting flight path data is shown in Figure 13. Airspeed ramp commands are given for $t \in [10, 45]$ s followed by an altitude ramp command for $t \in [70, 95]$ s. Furthermore, altitude and airspeed commands are given simultaneously for $t \in [125, 145]$ s. As shown in Figure 13, PCAC follows the altitude and airspeed commands with slight overshoot in the altitude command following.

Figure 14 shows the estimated coefficient vectors $\theta_{\rm long}$ and $\theta_{\rm vel}$, where $\theta_{\rm long}$ is the estimated coefficient vector of PCAC/long and $\theta_{\rm vel}$ is the estimated coefficient vector of PCAC/vel. ◇

VI. Conclusions and future work

For autonomous control of aircraft with unmodeled aerodynamics, this paper proposed an autopilot based on predictive cost adaptive control (PCAC), which is an indirect adaptive control extension of model predictive control. PCAC uses recursive least squares (RLS) with variable-rate forgetting for online, closed-loop system identification, and receding-horizon optimization based on quadratic programming. Output feedback is facilitated by using the

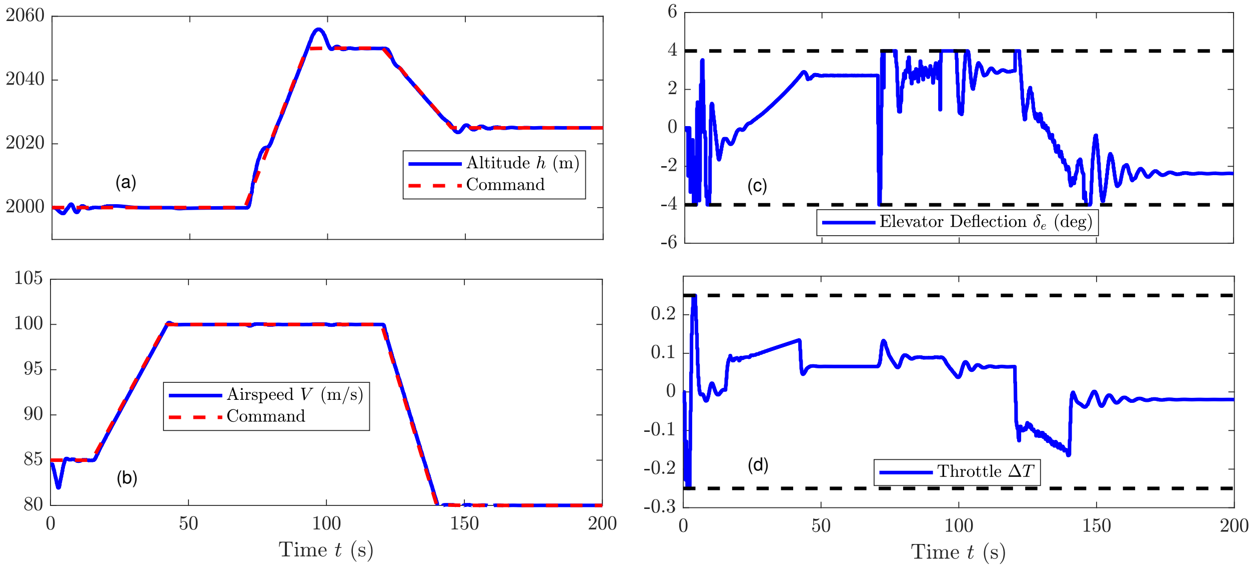

Fig. 13 *Altitude and airspeed command following and control surface deflection. (a) compares the command* $h_{\rm cmd}$ *and the simulated* h*. (b) compares the command* $V_{\rm cmd}$ *and the simulated data* V*. Note that learning takes place for both the longitudinal dynamics and throttle dynamics for* $t \in [0, 5]$ *s causing the initial excitations seen in the plots. (c) shows the elevator deflection* δ_e *(blue) with the constraint on* $|\delta_e|$ *(dashed black). (d) shows the throttle change* ΔT *(blue) with the constraint on* $|\Delta T|$ *(dashed black).*

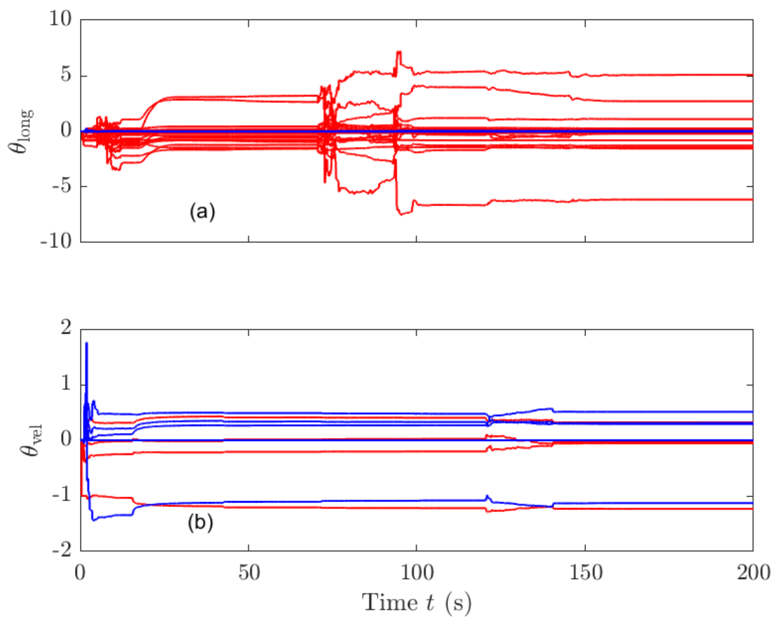

Fig. 14 *Estimated coefficient vectors for the altitude command-following loop* $\theta_{\rm long}$ *(a) and the airspeed command-following loop* $\theta_{\rm vel}$ *(b). Note* $\theta_{F_{\rm m},k}$ *for both loops is denoted by red lines while* $\theta_{G_{\rm m},k}$ *for both loops is denoted by blue lines. No prior modeling information is assumed, where* $\theta_{\rm long,0} = 0.01\boldsymbol{1}_{60\times 1}$ *and* $\theta_{\rm vel,0} = 0.01\boldsymbol{1}_{8\times 1}$.

block-observable canonical form realization, whose state is known exactly at each step. This technique is demonstrated numerically on a 6DOF linearized aircraft model and a 3DOF nonlinear aircraft model. In both cases, the adaptive autopilot is able to follow attitude, bank-angle, azimuth-angle, and velocity commands over a range of operation.

The ability of PCAC to operate as an adaptive autopilot without aerodynamic modeling has useful implications in practice. First and foremost, this technique can mitigate the need for wind tunnel testing. Second, it eliminates the need for gain scheduling, loop shaping, and dynamic inversion. Finally, PCAC can be used to accelerate the expensive and time-consuming aircraft/autopilot design cycle. This potential will be investigated in future research through flight testing of autonomous, fixed-wing aircraft.

References

[1] Tewari, A., *Advanced Control of Aircraft, Spacecraft and Rockets*, Wiley, 2011.

[2] Stengel, R. F., *Flight Dynamics*, 2nd ed., Princeton, 2022.

[3] Hovakimyan, N., and Cao, C., L_1 *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*, SIAM, 2010.

[4] Lavretsky, E., and Wise, K., *Robust and Adaptive Control: With Aerospace Applications*, Springer, 2012.

[5] Ansari, A., and Bernstein, D. S., "Retrospective Cost Adaptive Control of the Generic Transport Model Under Uncertainty and Failure," *J. Aerospace Information Systems*, Vol. 14, No. 3, 2017, pp. 123–174.

[6] Kwon, W., and Han, S., *Receding Horizon Control: Model Predictive Control for State Models*, Springer, 2006.

[7] Camacho, E. F., and Bordons, C., *Model Predictive Control*, 2nd ed., Springer, 2007. https://doi.org/10.1007/978-0-85729-398-5.

[8] Eren, U., Prach, A., Koçer, B. B., Raković, S. V., Kayacan, E., and Açıkmeşe, B., "Model predictive control in aerospace systems: Current state and opportunities," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 7, 2017, pp. 1541–1566. https://doi.org/10.2514/1.G002507.

[9] Rawlings, J. B., Mayne, D. Q., and Diehl, M. M., *Model Predictive Control: Theory, Computation, and Design*, Nob Hill, 2020.

[10] van Waarde, H. J., Eising, J., Trentelman, H. L., and Camlibel, M. K., "Data informativity: a new perspective on data-driven analysis and control," *IEEE Trans. Autom. Contr.*, Vol. 65, No. 11, 2020, pp. 4753–4768.

[11] Islam, S. A. U., Nguyen, T., Kolmanovsky, I., and Bernstein, D. S., "Data-Driven Retrospective Cost Adaptive Control for Flight Control Applications," *J. Guid. Contr. Dyn.*, Vol. 44, 2021, pp. 1732–1758.

[12] Soloperto, R., Müller, M. A., and Allgöwer, F., "Guaranteed closed-loop learning in model predictive control," *IEEE Trans. Autom. Contr.*, Vol. 68, 2022, pp. 991–1006.

[13] Feldbaum, A. A., "Dual Control Theory, Parts I and II," *Autom. Rem. Contr.*, Vol. 21, 1961, pp. 874–880, 1033–1039.

[14] Hjalmarsson, H., Gevers, M., De Bruyne, F., and Leblond, J., "Identification for Control: Closing the Loop Gives More Accurate Controllers," *Proc. IEEE Conf. Dec. Contr.*, Lake Buena Vista, FL, 1994, pp. 4150–4155.

[15] Forssell, U., and Ljung, L., "Closed-Loop Identification Revisited," *Automatica*, Vol. 35, No. 7, 1999, pp. 1215–1241.

[16] Filatov, N. M., and Unbehauen, H., *Adaptive Dual Control: Theory and Applications*, Springer, 2004.

[17] Mesbah, A., "Stochastic model predictive control with active uncertainty learning: A Survey on dual control," *Ann. Rev. Contr.*, Vol. 45, 2018, pp. 107–117.

[18] Nguyen, T. W., Islam, S. A. U., Bernstein, D. S., and Kolmanovsky, I. V., "Predictive Cost Adaptive Control: A Numerical Investigation of Persistency, Consistency, and Exigency," *IEEE Contr. Sys. Mag.*, Vol. 41, 2021, pp. 64–96.

[19] Nguyen, T. W., Kolmanovsky, I. V., and Bernstein, D. S., "Sampled-Data Output-Feedback Model Predictive Control of Nonlinear Plants Using Online Linear System Identification," *Proc. Amer. Contr. Conf.*, 2021, pp. 4682–4687.

[20] Islam, S. A. U., and Bernstein, D. S., "Recursive Least Squares for Real-Time Implementation," *IEEE Contr. Syst. Mag.*, Vol. 39, No. 3, 2019, pp. 82–85.

[21] Bruce, A. L., Goel, A., and Bernstein, D. S., "Convergence and consistency of recursive least squares with variable-rate forgetting," *Automatica*, Vol. 119, 2020, p. 109052.

[22] Goel, A., Bruce, A. L., and Bernstein, D. S., "Recursive Least Squares With Variable-Direction Forgetting: Compensating for the Loss of Persistency," *IEEE Contr. Sys. Mag.*, Vol. 40, No. 4, 2020, pp. 80–102.

[23] Bruce, A. L., Goel, A., and Bernstein, D. S., "Necessary and Sufficient Regressor Conditions for the Global Asymptotic Stability of Recursive Least Squares," *Sys. Contr. Lett.*, Vol. 157, 2021, pp. 1–7. Article 105005.

[24] Mohseni, N., and Bernstein, D. S., "Recursive least squares with variable-rate forgetting based on the F-test," *Proc. Amer. Contr. Conf.*, 2022, pp. 3937–3942.

[25] Kwon, W. H., and Pearson, A. E., "On Feedback Stabilization of Time-Varying Discrete Linear Systems," *IEEE Trans. Autom. Contr.*, Vol. AC-23, No. 3, 1978, pp. 479–481.

[26] Polderman, J. W., "A State Space Approach to the Problem of Adaptive Pole Assignment," *Mathematics of Control, Signals and Systems*, Vol. 2, No. 1, 1989, pp. 71–94.

[27] MathWorks Inc., "Lightweight Airplane Design," , 2023. URL https://www.mathworks.com/help/aeroblks/lightweight-airplane-design.html.