



# A Data-Driven Predictive Terrain-Following Autopilot for an Aircraft undergoing Stability-Augmentation-System Failure

Farooq Aslam, \*

*PhD Candidate, Institute of Space Technology, Islamabad, 44000, ICT, Pakistan,*

Syed Aseem UI Islam, †

*Research Fellow, University of Michigan, Ann Arbor, MI, 48109,*

Suhail Akhtar, ‡

*Professor, Institute of Space Technology, Islamabad, 44000, ICT, Pakistan,*

and

Dennis S. Bernstein, §

*Professor, University of Michigan, Ann Arbor, MI, 48109.*

**This paper presents a data-driven terrain-following autopilot that operates within the framework of traditional flight-control systems. In particular, the data-driven controller follows altitude commands, which represent the terrain-following objective, by generating pitch-rate commands, which are followed using a fixed-gain, inner-loop, pitch-rate controller. The paper demonstrates the ability of the terrain-following autopilot to operate reliably when the fixed-gain, inner-loop, pitch-rate loop becomes unstable due to an unknown sensor delay. This failure is mitigated by the data-driven terrain-following autopilot, which stabilizes the vehicle despite the unknown instability in the inner loop. This ability demonstrates the benefit of having a data-driven outer-loop autopilot that can compensate for an unforeseen and unknown failure of the fixed-gain inner loop controller.**

## I. Introduction

Terrain following is a key capability for military aircraft that wish to avoid detection by adversaries while entering contested airspace [1]. In particular, flying as closely as possible to the terrain allows an aircraft to avoid radar detection by flying under the radar's horizon, which is the lowest altitude for which a radar can track targets due to the curvature of the Earth and terrain [2, 3]. A related concept is terrain avoidance, which aims to fly a path that may be three-dimensional. However, terrain following focuses on straight-line flight from one point to another point, while attempting to follow the terrain between those points as closely as possible.

Flying close to the terrain is challenging due to several reasons. The aircraft must maneuver rapidly in response to the terrain in order to follow it as closely as possible. These trajectories usually require excessive pilot workload, and thus are flown using autopilots [4–7]. Furthermore, accurate knowledge of the terrain is required if terrain following is to avoid obstacles. Constraints that are inherent to terrain following include vertical acceleration limits due to the vertical maneuvering ability of the aircraft, the required level of comfort for the pilot, and, because of transient behavior, the need for a safety margin.

Before the advent of extensive satellite scanning, terrain-following autopilots relied primarily on terrain-following radars, which would scan the terrain immediately ahead of the aircraft [2, 8]. However, contemporary terrain-following autopilots can use widely available terrain data [9, 10] to anticipate obstacles and plan terrain-following trajectories. In practice, the two techniques are combined using data-fusion methods to merge radar-scans of the terrain and terrain data to provide an up-to-date map of the terrain [11–13].

Since terrain-following autopilots are employed on non-experimental aircraft, they must utilize the existing flight-control system architecture. Modern aircraft utilize a stability augmentation systems (SAS) as an inner-loop flight-control

\*PhD Candidate, Institute of Space Technology, Islamabad, 44000, ICT, Pakistan.

†Research Fellow, Aerospace Engineering, 1320 Beal Ave, Ann Arbor, MI, 48109.

‡Professor, Institute of Space Technology, Islamabad, 44000, ICT, Pakistan.

§Professor, Aerospace Engineering, 1320 Beal Ave, Ann Arbor, MI, 48109.

system to regulate aircraft body rates  $p$ ,  $q$ , and  $r$  [14]. Since terrain following is restricted to the pitch plane, only the pitch-rate  $q$  SAS is of relevance for terrain following. In particular, terrain-following autopilots generate a commanded pitch rate, which is passed to the pitch-rate SAS, which, in turn, requests actuator deflections that facilitate following of the commanded pitch rate.

Although the inner-/outer-loop architecture is standard in practice, this architecture may be problematic when the pitch-rate SAS fails to stabilize body rates. Failure of the pitch-rate SAS is especially dangerous in flight close to the ground, which leaves little time for remedial measures. This situation motivates the present paper.

A promising approach to terrain following is model predictive control (MPC), which applies receding-horizon optimization to the vehicle model and desired trajectory to determine a future sequence of control inputs subject to constraints on actuator magnitude and rate [15–18]. The ability of predictive control methods to handle nonlinear dynamics, incorporate actuation constraints, and include knowledge of the desired trajectory make them viable candidates for developing terrain following algorithms [7, 19–21]. The ability of MPC to consider a future horizon allows it to anticipate challenging terrain profiles, which makes MPC ideal for terrain following. A key advantage of MPC is its ability to use knowledge of the dynamics of the aircraft and limitations of its actuators when computing an optimal trajectory. A challenge in the embedded implementation of MPC, however, is computational complexity due to the requirement to solve an online optimization problem, which may be nonlinear.

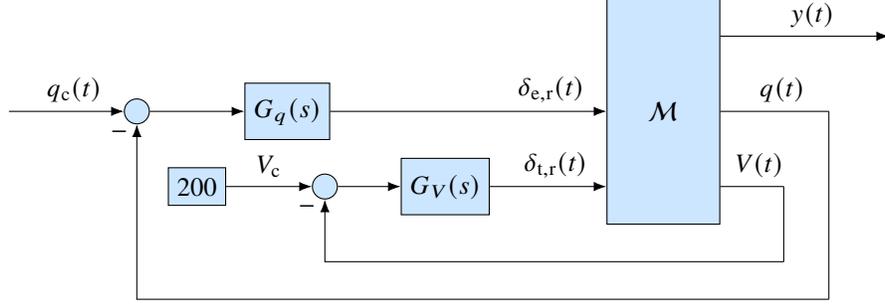
The main contribution of the present paper is the development of a data-driven, MPC-based, terrain-following autopilot that takes into account constraints that are crucial for terrain-following applications. In particular, the terrain-following autopilot specifies control actions that enable the vehicle to fly as closely as possible to the terrain, while not descending below a safety height. Since the terrain-following autopilot in this paper is data-driven, it utilizes no prior modeling information but rather learns the dynamics of the aircraft using online, closed-loop system identification. This allows the terrain-following autopilot to be applied to different aircraft with minimal modification. Another consequence of the data-driven nature of the autopilot is that constraints are satisfied only with respect to the latest identified model, which may not be accurate when the autopilot starts. The terrain-following autopilot presented in this paper may also be used as an offline planning tool to chart feasible routes for particular aircraft, even before they are flown.

In addition to the above benefits of a data-driven autopilot for terrain following, the present paper focuses on the situation where the fixed-gain, inner-loop, pitch-rate SAS becomes unstable during flight. When the outer-loop controller is also fixed gain, this failure mode will almost surely lead to loss of the vehicle, especially since flying close to the terrain presents an unrecoverable situation. The present paper thus investigates the ability of the data-driven outer-loop controller to compensate for the destabilized inner loop. In particular, the data-driven controller follows altitude commands, which represent the terrain-following objective, by generating pitch-rate commands, which are followed using a fixed-gain, inner-loop, pitch-rate controller. During flight, the fixed-gain, inner-loop, pitch-rate loop becomes unstable due to an unknown sensor delay. This failure is mitigated by the data-driven terrain-following autopilot, which stabilizes the vehicle despite the unknown instability in the inner loop. This ability demonstrates the benefit of having a data-driven outer-loop autopilot that can compensate for an unforeseen and unknown failure of the fixed-gain, inner-loop, pitch-rate controller.

The contents of this paper are as follows. Section II presents the model of the aircraft used for simulations in this paper and the pitch-rate stability augmentation system (SAS) used as the inner loop. Section III presents the architecture used for terrain following. Section IV reviews predictive cost adaptive control (PCAC) [22], which is the data-driven model-predictive autopilot used for terrain following in this paper. Section V presents simulations that demonstrate the ability of PCAC to function as a terrain-following autopilot, with no prior modeling information, in the presence of SAS failure, and in the presence of previously undetected obstacles. Finally, section VI provides concluding remarks and directions for future research.

## II. Aircraft Model and Stability Augmentation System

The aircraft model used in this paper is the nonlinear F-16 model described in [14, 23, 24]. In order to simplify autopilot development, we restrict the aircraft's motion to the longitudinal plane. The aircraft model is represented by  $\mathcal{M}$  in Figure 1 and includes actuator models for the elevator and the engine. The inputs to the actuators are the requested elevator deflection  $\delta_{e,r} \in [-25^\circ, 25^\circ] \in \mathbb{R}$  and the requested thrust  $\delta_{t,r} \in [0, 1]$ . In addition to the elevator and throttle, the nonlinear F-16 model takes as input the deflection of the leading edge flap. In this paper, the leading edge flap is set to zero, and is not used as a control input. We assume that measurements of angle-of-attack  $\alpha(t)$ , flight-path angle  $\gamma(t)$ , pitch-rate  $q(t)$ , altitude  $h(t)$ , normal acceleration  $n_z(t)$ , and airspeed  $V(t)$  are available.



**Fig. 1** Continuous-time longitudinal stability augmentation system for a fixed-wing aircraft.  $\mathcal{M}$  represents the nonlinear fixed-wing aircraft dynamics. The pitch-rate controller  $G_q(s)$  is designed to follow pitch-rate commands  $q_c(t)$  by generating requested elevator deflections  $\delta_{e,r}(t)$ . The airspeed controller  $G_V(s)$  is designed to follow a constant airspeed command  $V_c = 200$  by generating the requested throttle settings  $\delta_{t,r}(t)$ .  $y(t)$  includes measurements made from sensors on the aircraft.

In order to design an inner-loop stability augmentation system (SAS), the nonlinear F-16 model is trimmed at an airspeed of  $V \equiv 200$  m/s and an altitude of  $h \equiv 1000$  m. Next, a linear model is obtained for the aircraft's longitudinal dynamics about the trim state. To design the pitch-rate controller, the transfer function from  $\delta_e$  to  $q$  is computed where the poles corresponding to the phugoid mode are removed. A first-order actuator model transfer function, which relates  $\delta_e$  to  $\delta_{e,r}$  is also included. Multiplying these transfer functions yields the transfer function from  $\delta_{e,r}$  to  $q$ , which is given by

$$G_{sp}(s) = -\frac{348.73(s + 1.249)}{(s + 20.2)(s^2 + 2.888s + 2.914)},$$

where the actuator pole is at  $s = -20.2$ . The designed pitch-rate PI controller is given by

$$G_q(s) = -\frac{0.2251(s + 3.5911)}{s},$$

which yields  $60^\circ$  phase margin, infinite gain margin, loop crossover frequency 4.65 rad/s, rise time 0.285 s, overshoot 10.9%, and settling time 1.69 s.

In order to design the speed controller, the closed-loop transfer function from  $\delta_{t,r}$  to  $V$  is obtained, which includes  $G_q(s)$ . In particular, this transfer function is given by

$$G_{as}(s) = \frac{7.8325(s + 1.144)}{(s + 1)(s + 1.141)(s + 0.02224)},$$

where the actuator pole is at  $s = -1$ . The designed speed PI controller is given by

$$G_V(s) = \frac{0.063(s + 0.061087)}{s},$$

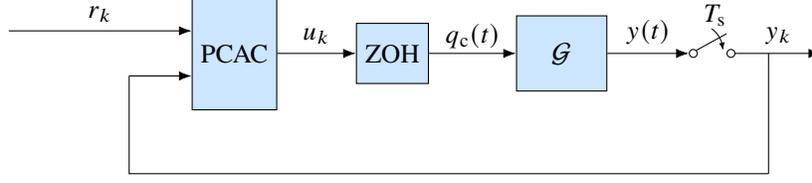
which yields  $60.7^\circ$  phase margin, infinite gain margin, loop crossover frequency 0.454 rad/s, rise time 2.81 s, overshoot 11.5%, and settling time 24.9 s. The commanded airspeed  $V_c$  is set to be constant at 200 m/s.

### III. Architecture used for Terrain Following

This paper utilizes PCAC for terrain-following in the presence of constraints. As shown in Figure 2 we consider sampled-data control of the nonlinear F-16 model, where the commanded pitch-rate  $q_c(t)$  is generated by PCAC. For all  $t \geq 0$ ,  $q_c(t) \in \mathbb{R}$  is the control, and  $y(t)$  is the the output of  $\mathcal{G}$ . The sample operation yields  $y_k \triangleq y(kT_s)$ , where  $T_s$  is the sampling time.

The tracking output  $y_{t,k} \in \mathbb{R}$  is defined by

$$y_{t,k} \triangleq C_t y_k, \quad (1)$$



**Fig. 2** Command following under sampled-data data-driven predictive control.  $q_c(t)$  and  $y(t)$  are the commanded pitch-rate and measurements in Figure 1, respectively. That is, the relationship from  $q_c(t)$  to  $y(t)$  is represented by  $\mathcal{G}$ . PCAC has access to the command  $r_k$  and sampled measurements  $y_k$ . The objective is to have linear combinations of  $y_k$  follow the commands  $r_k$ , while enforcing constraints on additional linear combinations of  $y_k$ , the control  $u_k$ , and the control move-size  $u_k - u_{k-1}$ . A zero-order hold circuit with input  $u_k$  generates the control signal  $q_c(t)$  for the stability augmentation system. All sample-and-hold operations are synchronous.

which specifies the sampled altitude as the tracking output. The performance objective is to have the sampled altitude  $y_{t,k}$  follow a commanded terrain trajectory  $r_k \in \mathbb{R}$ , whose future values are known, and thus command preview is available. It can be assumed that command preview of the terrain is possible through the availability of offline terrain data [10] and online radar-scan information [11–13].

In addition to the performance objective, the constrained output  $y_{c,k} \in \mathbb{R}^{p_c}$  is defined by

$$y_{c,k} \triangleq C_c y_k, \quad (2)$$

where  $C_c \in \mathbb{R}^{p_c \times p}$ . In this paper  $C_c$  is selected to implement soft constraints on altitude and normal acceleration. The objective is to enforce the inequality constraint

$$C y_{c,k} + \mathcal{D} \leq 0_{n_c \times 1}, \quad (3)$$

where  $C \in \mathbb{R}^{n_c \times p_c}$  and  $\mathcal{D} \in \mathbb{R}^{n_c}$ . Note that (3), where “ $\leq$ ” is interpreted component-wise, defines a convex set.

Additionally, the control is constrained in both magnitude and rate. The magnitude control constraint has the form

$$u_{\min} \leq u_k \leq u_{\max}, \quad (4)$$

where  $u_{\min} \in \mathbb{R}$  is the value of the minimum control magnitude and  $u_{\max} \in \mathbb{R}$  is the value of maximum control magnitude. In addition, the move-size control constraint has the form

$$\Delta u_{\min} \leq u_k - u_{k-1} \leq \Delta u_{\max}, \quad (5)$$

where  $\Delta u_{\min} \in \mathbb{R}$  is the value of minimum control move size and  $\Delta u_{\max} \in \mathbb{R}$  is the value of maximum control move sizes. As shown in Figure 2, the inputs to PCAC are the command  $r_k$ , tracking output  $y_{t,k}$ , and constrained output  $y_{c,k}$ . Using these signals, PCAC produces the discrete-time control  $u_k \in \mathbb{R}$  at each step  $k$ , which minimizes the norm of the difference between the command following output and the command over a horizon.

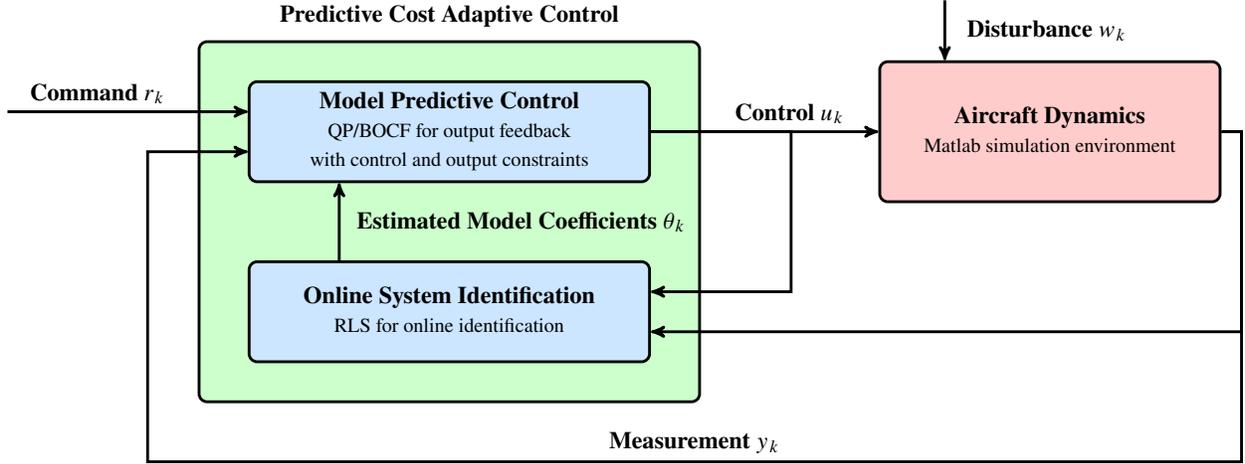
## IV. Review of Predictive Cost Adaptive Control

In this paper, we use PCAC as the data-driven, terrain-following autopilot. PCAC combines online identification with output-feedback MPC, which is presented in this section. Subsection IV.A describes the technique used for online identification, namely, recursive least squares. Subsection IV.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 IV.C reviews the MPC technique for receding-horizon optimization.

### A. Online Identification Using Recursive Least Squares

Let  $\hat{n} \geq 0$  and, for all  $k \geq 0$ , let  $\hat{F}_{1,k}, \dots, \hat{F}_{\hat{n},k} \in \mathbb{R}^{p \times p}$  and  $\hat{G}_{0,k}, \dots, \hat{G}_{\hat{n},k} \in \mathbb{R}^{p \times m}$  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

$$\hat{y}_k = - \sum_{i=1}^{\hat{n}} \hat{F}_{i,k} y_{k-i} + \sum_{i=0}^{\hat{n}} \hat{G}_{i,k} u_{k-i}, \quad (6)$$



**Fig. 3** PCAC block diagram. The online, closed-loop system identification is based on recursive least squares (RLS). The model predictive control (MPC) algorithm, which is based on quadratic programming (QP), uses the estimated model coefficients  $\theta_k$  to form a block-observable canonical form (BOCF) state-space model, which is used by QP to determine the control input  $u_k$ .

where

$$y_{-\hat{n}} = \cdots = y_{-1} = 0, \quad (7)$$

$$u_{-\hat{n}} = \cdots = u_{-1} = u_0 = 0. \quad (8)$$

For online identification, RLS is used to estimate the coefficients of the input-output model (6). To do this, RLS minimizes the cumulative cost

$$J_k(\theta_k) = \sum_{i=0}^k z_i^T(\theta_k) z_i(\theta_k) + (\theta_k - \theta_0)^T P_0^{-1} (\theta_k - \theta_0), \quad (9)$$

where, for all  $k \geq 0$ ,  $P_0 \in \mathbb{R}^{[\hat{n}p(m+p)+mp] \times [\hat{n}p(m+p)+mp]}$  is positive definite, and  $\theta_0 \in \mathbb{R}^{\hat{n}p(m+p)+mp}$  is the initial estimate of the coefficient vector. The performance variable  $z_k(\theta_k) \in \mathbb{R}^p$  is defined by

$$z_k(\theta_k) \triangleq y_k + \sum_{i=1}^{\hat{n}} \hat{F}_{i,k} y_{k-i} - \sum_{i=0}^{\hat{n}} \hat{G}_{i,k} u_{k-i}, \quad (10)$$

where the vector  $\hat{\theta}_k \in \mathbb{R}^{\hat{n}p(m+p)+mp}$  of estimated coefficients is defined by

$$\theta_k \triangleq \text{vec} \begin{bmatrix} \hat{F}_{1,k} & \cdots & \hat{F}_{\hat{n},k} & \hat{G}_{0,k} & \cdots & \hat{G}_{\hat{n},k} \end{bmatrix}. \quad (11)$$

Defining the regressor matrix  $\phi_k \in \mathbb{R}^{p \times [\hat{n}p(m+p)+mp]}$  by

$$\phi_k \triangleq \begin{bmatrix} -y_{k-1}^T & \cdots & -y_{k-\hat{n}}^T & u_k^T & \cdots & u_{k-\hat{n}}^T \end{bmatrix}^T \otimes I_p, \quad (12)$$

it follows that the performance variable (10) can be rewritten as

$$z_k(\theta_k) = y_k - \phi_k \theta_k. \quad (13)$$

Note that, with (13), the cost function (9) is strictly convex and quadratic, and thus has a unique global minimizer. The unique global minimizer is computed by RLS using

$$P_{k+1} = P_k - P_k \phi_k^T (I_p + \phi_k P_k \phi_k^T)^{-1} \phi_k P_k, \quad (14)$$

$$\theta_{k+1} = \theta_k + P_{k+1} \phi_k^T (y_k - \phi_k \theta_k). \quad (15)$$

Note that  $\theta_{k+1}$  computed using (15) is available at step  $k$ , and thus,  $\hat{F}_{1,k+1}, \dots, \hat{F}_{\hat{n},k+1}, \hat{G}_{0,k+1}, \dots, \hat{G}_{\hat{n},k+1}$  are available at step  $k$ .

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

Considering the estimate  $\hat{y}_k$  of  $y_k$  given by (6), it follows that, for all  $k \geq 0$ ,

$$\hat{y}_k = - \sum_{i=1}^{\hat{n}} \hat{F}_{i,k} \hat{y}_{k-i} + \sum_{i=0}^{\hat{n}} \hat{G}_{i,k} u_{k-i}. \quad (16)$$

It follows that, for all  $k \geq 0$ , the BOCF state-space realization of (16) is given by [25]

$$\hat{x}_{k+1} = \hat{A}_k \hat{x}_k + \hat{B}_k u_k, \quad (17)$$

$$\hat{y}_k = \hat{C} \hat{x}_k + \hat{D}_k u_k, \quad (18)$$

where

$$\hat{A}_k \triangleq \begin{bmatrix} -\hat{F}_{1,k+1} & I_p & \cdots & \cdots & 0_{p \times p} \\ -\hat{F}_{2,k+1} & 0_{p \times p} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0_{p \times p} \\ \vdots & \vdots & & \ddots & I_p \\ -\hat{F}_{\hat{n},k+1} & 0_{p \times p} & \cdots & \cdots & 0_{p \times p} \end{bmatrix} \in \mathbb{R}^{\hat{n}p \times \hat{n}p}, \quad \hat{B}_k \triangleq \begin{bmatrix} \hat{G}_{1,k+1} - \hat{F}_{1,k+1} \hat{G}_{0,k+1} \\ \hat{G}_{2,k+1} - \hat{F}_{2,k+1} \hat{G}_{0,k+1} \\ \vdots \\ \hat{G}_{\hat{n},k+1} - \hat{F}_{\hat{n},k+1} \hat{G}_{0,k+1} \end{bmatrix} \in \mathbb{R}^{\hat{n}p \times m}, \quad (19)$$

$$\hat{C} \triangleq [I_p \quad 0_{p \times p} \quad \cdots \quad 0_{p \times p}] \in \mathbb{R}^{p \times \hat{n}p}, \quad \hat{D}_k \triangleq \hat{G}_{0,k+1} \in \mathbb{R}^{m \times m}, \quad (20)$$

$$\hat{x}_k \triangleq \begin{bmatrix} \hat{x}_{k,1} \\ \vdots \\ \hat{x}_{k,\hat{n}} \end{bmatrix} \in \mathbb{R}^{\hat{n}p}, \quad \hat{x}_{k,1} \triangleq y_k - \hat{G}_{0,k+1} u_k, \quad (21)$$

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

$$\hat{x}_{k,j} \triangleq - \sum_{i=1}^{\hat{n}-j+1} \hat{F}_{i+j-1,k+1} y_{k-i} + \sum_{i=1}^{\hat{n}-j+1} \hat{G}_{i+j-1,k+1} u_{k-i}. \quad (22)$$

## C. Model Predictive Control (MPC)

Let  $\ell \geq 1$  be the horizon and, for all  $k \geq 0$  and all  $i = 1, \dots, \ell$ , let  $\hat{x}_{k|i} \in \mathbb{R}^{\hat{n}p}$  be the  $i$ -step predicted state,  $\hat{y}_{k|i} \in \mathbb{R}^p$  be the  $i$ -step predicted output, and  $u_{k|i} \in \mathbb{R}^m$  be the  $i$ -step predicted control. Then, the  $\ell$ -step predicted output of (18) for a sequence of  $\ell$  future controls is given by

$$Y_{\ell,k|1} = \Gamma_{\ell,k} \hat{x}_{k|1} + T_{\ell,k} U_{\ell,k|1}, \quad (23)$$

where

$$Y_{\ell,k|1} \triangleq \begin{bmatrix} \hat{y}_{k|1} \\ \vdots \\ \hat{y}_{k|\ell} \end{bmatrix} \in \mathbb{R}^{\ell p}, \quad U_{\ell,k|1} \triangleq \begin{bmatrix} u_{k|1} \\ \vdots \\ u_{k|\ell} \end{bmatrix} \in \mathbb{R}^{\ell m}, \quad (24)$$

$$\Gamma_{\ell,k} \triangleq \begin{bmatrix} \hat{C} \\ \hat{C} \hat{A}_k \\ \vdots \\ \hat{C} \hat{A}_k^{\ell-1} \end{bmatrix} \in \mathbb{R}^{\ell p \times \hat{n}p}, \quad T_{\ell,k} \triangleq \begin{bmatrix} \hat{D}_k & \cdots & \cdots & \cdots & \cdots & 0_{p \times m} \\ H_{k,1} & \hat{D}_k & \cdots & \cdots & \cdots & 0_{p \times m} \\ H_{k,2} & H_{k,1} & \hat{D}_k & \cdots & \cdots & 0_{p \times m} \\ H_{k,3} & H_{k,2} & H_{k,1} & \hat{D}_k & \cdots & 0_{p \times m} \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0_{p \times m} \\ H_{k,\ell-1} & H_{k,\ell-2} & H_{k,\ell-3} & \cdots & H_{k,1} & \hat{D}_k \end{bmatrix} \in \mathbb{R}^{\ell p \times \ell m}, \quad (25)$$

where  $H_{k,i} \triangleq \hat{C}\hat{A}_k^{i-1}\hat{B}_k \in \mathbb{R}^{p \times m}$  for all  $i = 1, \dots, \ell - 1$ .

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

$$\Delta U_{\ell,k|1} \triangleq [(u_{k|1} - u_k)^T (u_{k|2} - u_{k|1})^T \dots (u_{k|\ell} - u_{k|\ell-1})^T]^T \in \mathbb{R}^{\ell m}. \quad (26)$$

Then, the receding horizon optimization problem is given by

$$\min_{U_{\ell,k|1}} (Y_{t,\ell,k|1} - \mathcal{R}_{\ell,k})^T Q (Y_{t,\ell,k|1} - \mathcal{R}_{\ell,k}) + \Delta U_{\ell,k|1}^T R \Delta U_{\ell,k|1} + \varepsilon^T S \varepsilon, \quad (27)$$

subject to

$$C_\ell Y_{1|k,\ell} + \mathcal{D}_\ell \leq \varepsilon, \quad (28)$$

$$U_{\min} \leq U_{1|k,\ell} \leq U_{\max}, \quad (29)$$

$$\Delta U_{\min} \leq \Delta U_{1|k,\ell} \leq \Delta U_{\max}, \quad (30)$$

$$0_{\ell n_c \times 1} \leq \varepsilon, \quad (31)$$

where  $Q \in \mathbb{R}^{\ell p_t \times \ell p_t}$  is the positive-definite output weighting,  $R \in \mathbb{R}^{\ell m \times \ell m}$  is the positive definite control move-size weight,  $S \in \mathbb{R}^{\ell n_c \times \ell n_c}$  is the positive-definite constraint relaxation weight,  $U_{\min} \triangleq 1_\ell \otimes u_{\min} \in \mathbb{R}^{\ell m}$ ,  $U_{\max} \triangleq 1_\ell \otimes u_{\max} \in \mathbb{R}^{\ell m}$ ,  $\Delta U_{\min} \triangleq 1_\ell \otimes \Delta u_{\min} \in \mathbb{R}^{\ell m}$ ,  $\Delta U_{\max} \triangleq 1_\ell \otimes \Delta u_{\max} \in \mathbb{R}^{\ell m}$ , and  $C_\ell \in \mathbb{R}^{\ell n_c \times \ell p}$  and  $\mathcal{D}_\ell \in \mathbb{R}^{\ell n_c}$  are matrices that specify the desired inequality constraints. The quadratic program (QP) optimization (27)–(31) is solved using Nesterov's accelerated gradient descent algorithm [26, p. 76] on the dual problem, where for each step  $k$ , the Lagrange multipliers are initialized at zero.

In summary, at each time step, online identification is performed to find input-output model coefficients  $\theta_{k+1}$ , which are then used to create a state space realization  $(\hat{A}_k, \hat{B}_k, \hat{C}, \hat{D}_k)$ . Then, the state-space realization is used in a receding horizon optimization problem to solve for the  $\ell$ -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.

## V. Simulations

For all of the examples in this paper the measured outputs are selected as

$$y = \begin{bmatrix} \alpha & \gamma & q & h & n_z \end{bmatrix}^T \in \mathbb{R}^5, \quad (32)$$

where  $\alpha$  is the angle of attack,  $\gamma$  is the flight path angle,  $q$  is the pitch-rate,  $h$  is the altitude, and  $n_z$  is the normal acceleration defined as

$$n_z \triangleq -\frac{1}{g} \left( \frac{F_z}{m} + g \cos \theta \right), \quad (33)$$

$F_z$  is the net force on the aircraft in the aircraft-frame  $z$  direction,  $m$  is the aircraft mass,  $\theta$  is the aircraft pitch angle, and  $g$  is the acceleration due to gravity. Note that  $n_z$  is the gravity-corrected acceleration of the aircraft in  $g$ 's, positive towards the head of the pilot. Furthermore,  $n_z = 0$  in steady-level flight.

We implement PCAC as shown in the architecture in Figure 2. In particular, PCAC generates a sequence  $u_k$ , which is processed through a zero-order hold to generate a pitch-rate command  $q_c(t)$  signal. An SAS operates to achieve this commanded pitch-rate, in addition to a speed command of 200 m/s, as shown in Figure 1. The controllers in Figure 1 are designed through the process explained in Section III.

For PCAC, we set  $p = 5$  and  $m = 1$ . For online ID we set  $\hat{n} = 5$ ,  $P_0 = 10^6$ , and  $\theta_0 = 0.1 \cdot \mathbf{1}_{(\hat{n}p(m+p)+mp) \times 1}$ . Note that this choice of  $\theta_0$  corresponds to an initialization of PCAC without prior modeling information. We choose the PCAC step time as  $T_s = 0.1$  s/step, and the prediction horizon as  $\ell = 100$ . This corresponds to a prediction time of 10 s, and prediction over a flown distance of approximately 2 km, assuming that the aircraft flies at an airspeed close to 200 m/s.

Terrain data is obtained from the online databases provided by the U.S. Geological Survey [10]. In particular, the terrain profile starts at 39.5°N, 32°E, extends directly eastwards for a downrange distance of approximately 45 km, and has a downrange resolution of 1-arc-second, that is, 30 m. An offset of 300 m is added to the terrain profile to give an initial altitude of 1000 m, which is the altitude at which the aircraft is trimmed and linearized for controller design. The terrain profile is extended with a flat region to give an initial distance of approximately 2 km for warm start of the terrain-following algorithm. We denote the terrain data as  $\mathcal{T} = [T_1 \ \cdots \ T_N] \in \mathbb{R}^{1 \times N}$  where  $N$  is the number of terrain data points. At each step  $k$ , we set  $\mathcal{R}_{\ell,k} = [T_\tau + 50 \ \cdots \ T_{\tau+\ell} + 50]^T \in \mathbb{R}^\ell$ , where  $T_\tau$  is the element of  $\mathcal{T}$  that corresponds to the terrain height immediately below the aircraft. That is, PCAC has data of the  $\ell$ -step future terrain. Note that  $N$  is selected so that at the end of the simulation  $\ell$  additional data points of  $\mathcal{T}$  remain.

The desired constraints on the commanded pitch-rate are  $q_c \in [-20, 30]^\circ/\text{s}$  and  $\dot{q}_c \in [-100, 100]^\circ/\text{s}^2$ . These are specified as  $u_{\min} = -20^\circ/\text{s}$ ,  $u_{\max} = 30^\circ/\text{s}$ ,  $\Delta u_{\min} = -100 \cdot T_s^\circ/\text{s}^2$ , and  $\Delta u_{\max} = 100 \cdot T_s^\circ/\text{s}^2$ . Furthermore, we select  $Q = 100I$ ,  $R = 10^4I$ ,  $S = 10^{12}I$ ,  $p_t = 1$ , and  $h$  as the tracking output, that is,

$$C_t = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{1 \times 5}.$$

Lastly, we choose  $n_z$  and  $h$  as the constrained output by setting  $p_c = 2$ ,  $n_c = 3$ ,

$$C_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{2 \times 5},$$

where the desired constraints are  $-1 \leq n_z \leq 2$  and  $[T_\tau + 37.5 \ \cdots \ T_{\tau+\ell} + 37.5] \leq [h_{k|1} \ \cdots \ h_{k|\ell}]$ , where the inequality is interpreted element-wise, and  $h_{k|i}$  is the predicted altitude at step  $k+i$  using the data at step  $k$ . That is, we desire a one-sided inequality constraint on the predicted altitude, which is based on the terrain preview that is available to PCAC. The terrain following command  $\mathcal{R}_{\ell,k}$  is a trajectory that is the terrain shifted up vertically by 50 m and the desired one-sided constraint on the altitude is a trajectory that is the terrain shifted up vertically by 37.5 m, which the aircraft should remain above of for all time.

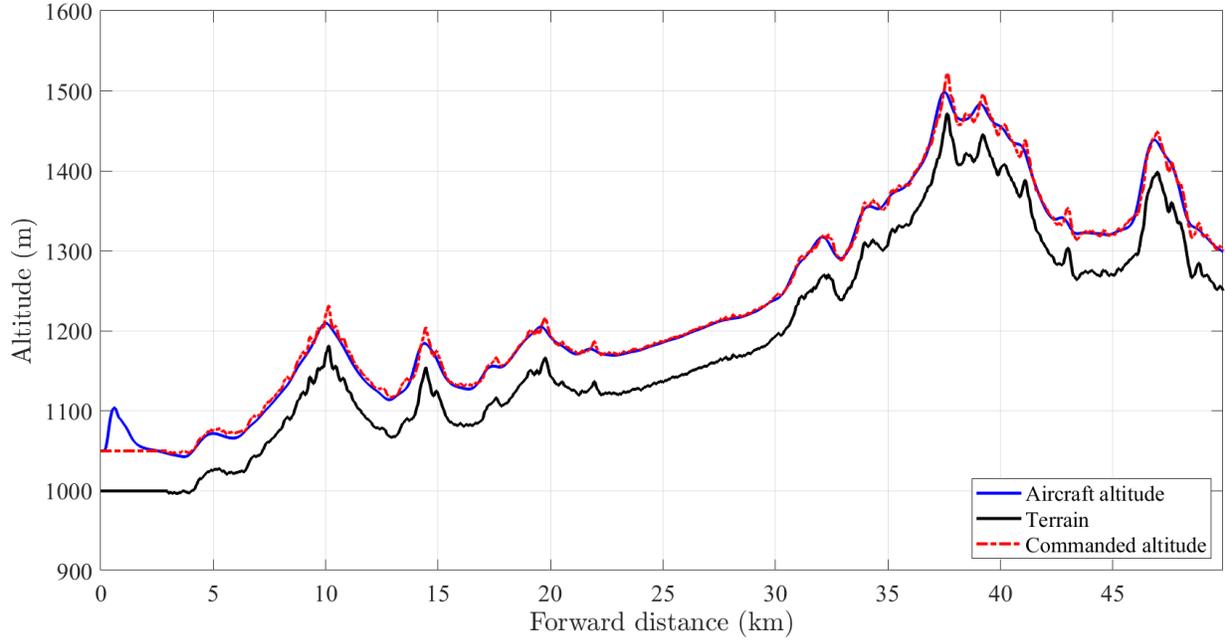
Note that the constraints in PCAC are soft due to the inclusion of the slack variable and slack weight in (27). This implies that the amount of command-following and constraint-satisfaction performance achieved depends on the relative magnitudes of the weighting matrices  $Q$  and  $S$ . That is the user may choose the right balance of command following versus constraint satisfaction through the selection of the weights in (27).

The total simulation length is set to 250 s and a discrete-time solver with sample time 0.01 s is used in Simulink.

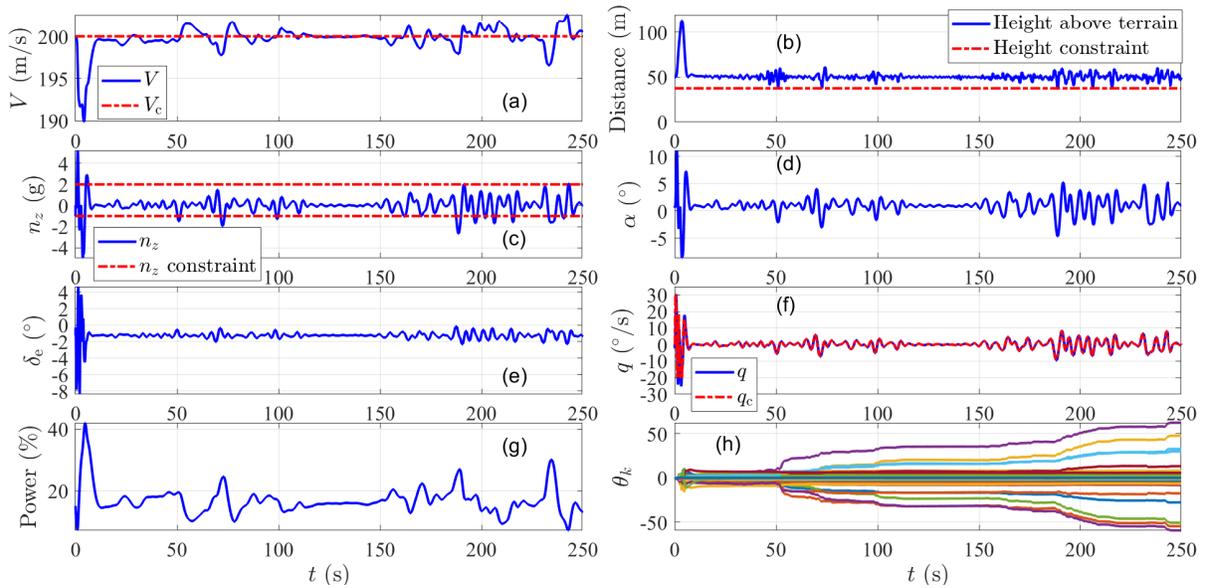
**Example 1.** *Terrain following with nominal SAS.* We consider the case where the SAS operates without failure. Figure 4 shows that terrain following is achieved while PCAC is initialized with no prior modeling information, as shown in Figure 5(h). Furthermore, Figure 5(a) shows that the soft constraints on  $n_z$  and  $h$  are respected most of the time and the airspeed is maintained around 200 m/s.  $\diamond$

**Example 2.** *Terrain following with SAS failure due to unknown delay.* We consider the case where the SAS fails due to unknown delay in the measurement  $q_k$ . In particular, an increasing amount of delays are added and the SAS fails with 23 steps of delay and as shown in Figure 7(c), and the delay for  $t > 140$  s is 39 steps. This shows that PCAC is compensating for the failure of the inner-loop SAS. Figure 6 shows that terrain following is achieved while PCAC is initialized with no prior modeling information, as shown in Figure 6(h). Furthermore, Figure 7(a) shows that the soft constraints on  $n_z$  and  $h$  are respected most of the time and the airspeed is maintained around 200 m/s.  $\diamond$

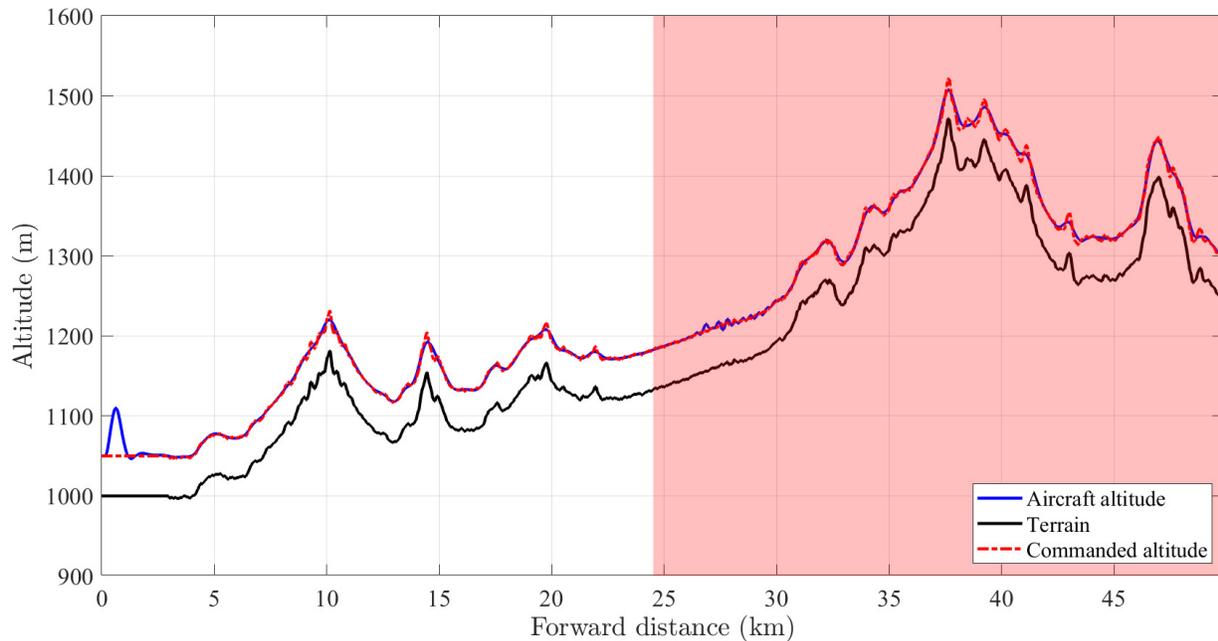
**Example 3.** *Terrain following with previously undetected obstacle.* We consider the case where due to outdated terrain data, a previously undetected obstacle appears within the preview horizon of PCAC. This represents a situation where through data-fusion of radar measurements with pre-stored terrain maps, a terrain element is detected very close to the aircraft that was previously not present in the terrain data. Recall that the PCAC horizon has the horizontal-flown distance of approximately 2 km. When the aircraft is at the 20 km downrange position, the terrain element 1 km ahead of the aircraft is raised in altitude by 60 m. This represents an obstacle that is detected only 1 km ahead due to previously being undetected. As shown in Figure 8 and its inset, PCAC maneuvers the aircraft to avoid the previously undetected obstacle that appears approximately half-way along its horizon. Figure 8 shows that terrain following is achieved while PCAC is initialized with no prior modeling information, as shown in Figure 8(h). Furthermore, Figure 9(a) shows that the soft constraints on  $n_z$  and  $h$  are respected most of the time and the airspeed is maintained around 200 m/s.  $\diamond$



**Fig. 4** Example 1: Terrain following with nominal SAS. The terrain is shown in black, the commanded altitude in red dashed, and the aircraft altitude in blue. Note that the commanded altitude is the terrain altitude increased by a safety height of 50 m.



**Fig. 5** Example 1: Terrain following with nominal SAS. (a) shows the achieved and commanded airspeeds; (b) shows the aircraft altitude above ground level and the soft constraint on  $h$ , which is 37.5 m above the terrain; (c) shows normal acceleration and its soft constraints; (d) shows the angle of attack  $\alpha$ ; (e) shows the achieved elevator deflection; (f) shows the achieved and commanded pitch rates; (g) shows the engine thrust as a percentage of the total thrust, which is equal to  $100\delta_t$ ; (h) shows the PCAC estimated model coefficients  $\theta_k$ , where PCAC is initialized with no prior modeling information.



**Fig. 6 Example 2: Terrain following with SAS failure due to delayed pitch-rate measurements. The terrain is shown in black, the commanded altitude in red dashed, and the aircraft altitude in blue. The commanded altitude is the terrain altitude increased by a safety height of 50 m. The shaded red region indicates the time interval during which the delay in the pitch-rate measurements exceeds 0.23 s, which is the time-delay margin of the inner-loop pitch-rate controller, and thus the SAS is destabilizing.**

## VI. Conclusions and Future Work

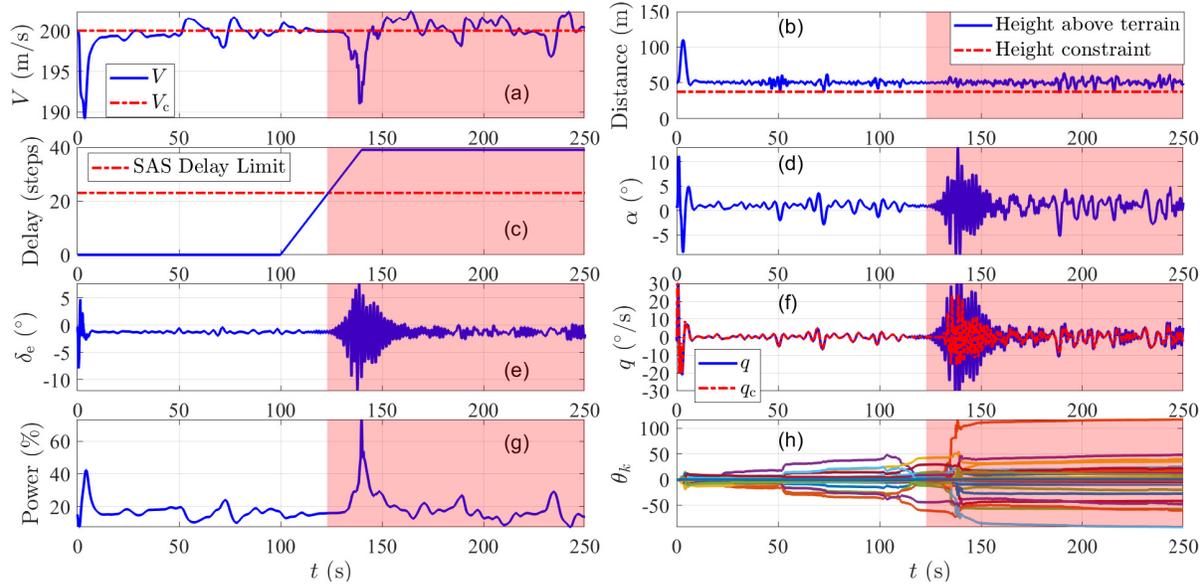
This paper presented a digital, data-driven, terrain-following algorithm based on predictive cost adaptive control (PCAC). PCAC combines concurrent online identification and output-feedback receding horizon control, within a discrete-time control framework. Magnitude and move-size limits on the actuation were also incorporated. Most importantly, PCAC's ability to consider a future horizon allows it to anticipate challenging terrain profiles, which makes PCAC ideal for terrain following, thus allowing the soft constraints on altitude and normal acceleration to be satisfied. A key advantage of PCAC is its ability to use knowledge of the dynamics of the aircraft and limitations of its actuators when computing an optimal trajectory.

PCAC-based terrain-following was applied in the framework of traditional inner-outer loop flight control where the inner-loop control was made up of fixed-design pitch-rate and speed controllers, and PCAC generated the commanded pitch-rate values, acting as the outer-loop autopilot. We demonstrated that when the inner-loop pitch-rate controller was destabilized due to delays beyond the allowable delay margin, PCAC-based terrain-following was able to continue to perform terrain following despite the failed inner-loop, pitch-rate controller. Additionally, we applied PCAC-based terrain-following to a case where a previously undetected obstacle was detected half-way within PCAC's horizon. PCAC-based terrain-following successfully maneuvered the aircraft to avoid this obstacle.

The next step for this research is to extend PCAC-based terrain following to explicitly include sensor fusion of radar measurements of the terrain with pre-stored terrain data. Being a data-driven algorithm, PCAC may be able to do sensor fusion without any modification. Additionally, we would extend PCAC-based terrain following to PCAC-based terrain avoidance, which is flying close to the ground in three-dimensions as opposed to being restricted to the pitch plane.

## References

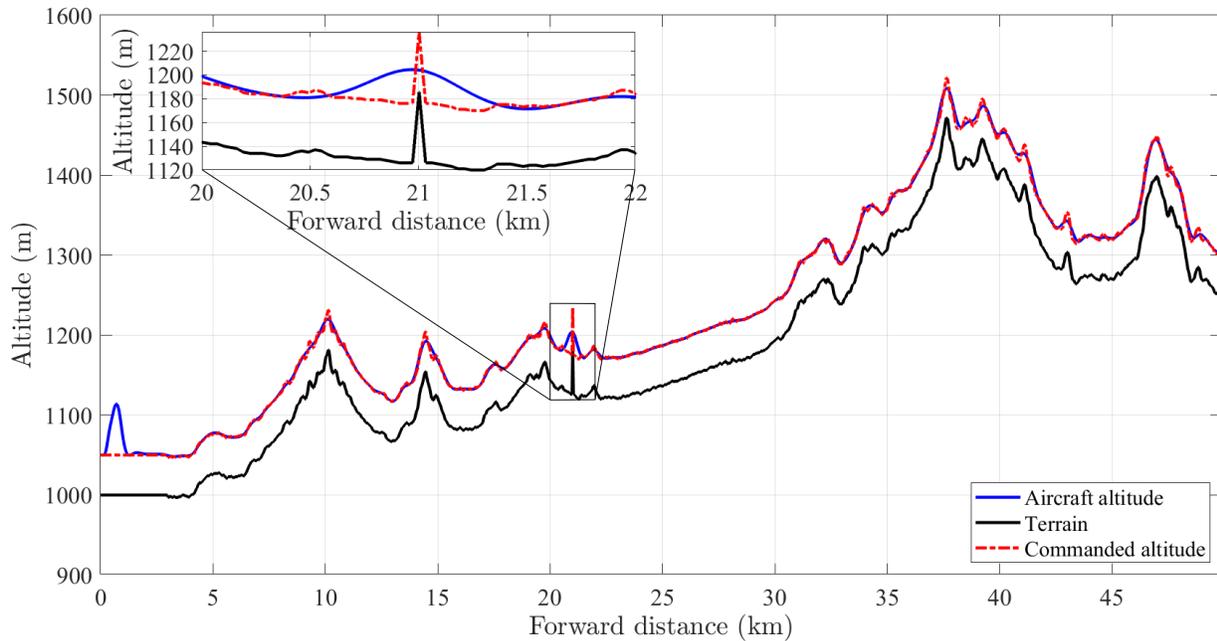
- [1] Barfield, A., Probert, J., and Browning, D., "All terrain ground collision avoidance and maneuvering terrain following for automated low level night attack," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 8, No. 3, 1993, pp. 40–47.
- [2] Starling, R., and Stewart, C., "The development of terrain following radar: an account of the progress made with an airborne



**Fig. 7 Example 2: Terrain following with SAS failure due to delayed pitch-rate measurements.** (a) shows the achieved and commanded airspeeds; (b) shows the aircraft altitude above ground level and the soft constraint on  $h$ , which is 37.5 m above the terrain; (c) shows the delay profile used to delay the pitch-rate measurements; (d) shows the angle of attack  $\alpha$ ; (e) shows the achieved elevator deflection; (f) shows the achieved and commanded pitch rates; (g) shows the engine thrust as a percentage of the total thrust, which is equal to  $100\delta_t$ ; (h) shows the PCAC estimated model coefficients  $\theta_k$ , where PCAC is initialized with no prior modeling information. In each subplot, the shaded red region indicates the time interval during which the delay in the pitch-rate measurements exceeds 0.23 s, which is the time-delay margin of the inner-loop pitch-rate controller, and thus the SAS is under failure.

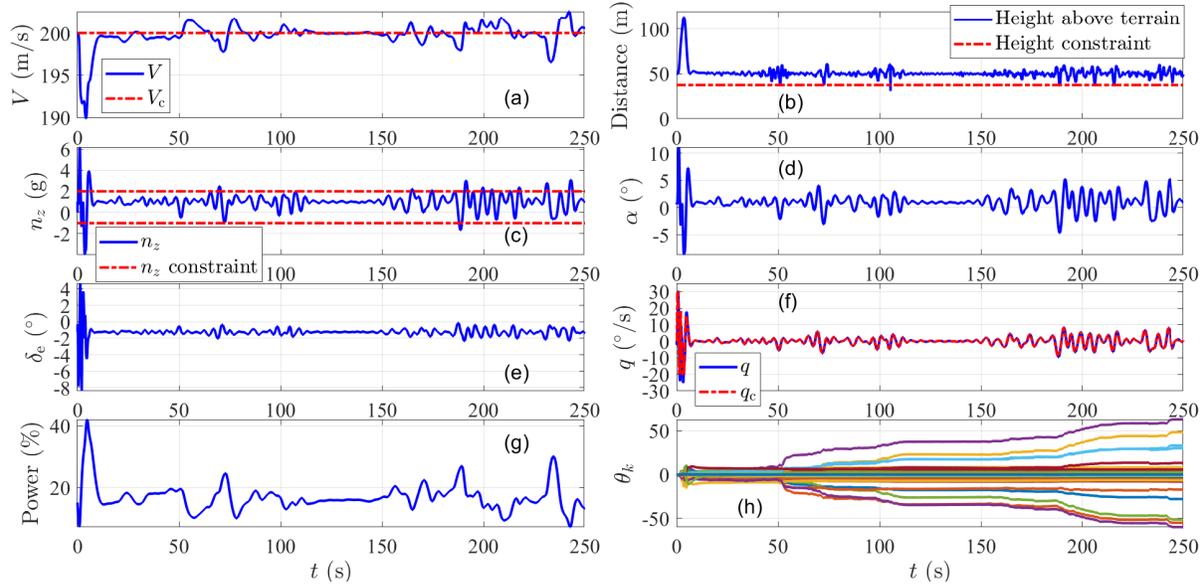
guidance system for low flying military aircraft,” *Aircraft Engineering and Aerospace Technology*, Vol. 43, No. 4, 1971, pp. 13–15.

- [3] Kamyar, R., and Taheri, E., “Aircraft optimal terrain/threat-based trajectory planning and control,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 2, 2014, pp. 466–483.
- [4] Funk, J. E., “Optimal-path precision terrain-following system,” *Journal of Aircraft*, Vol. 14, No. 2, 1977, pp. 128–134.
- [5] Asseo, S. J., “Terrain following/terrain avoidance path optimization using the method of steepest descent,” *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference*, IEEE, 1988, pp. 1128–1136.
- [6] Menon, P., Kim, E., and Cheng, V., “Optimal trajectory synthesis for terrain-following flight,” *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 4, 1991, pp. 807–813.
- [7] Williams, P., “Three-dimensional aircraft terrain-following via real-time optimal control,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, 2007, pp. 1201–1206.
- [8] Livshitz, A., and Idan, M., “Low-cost laser range-measurement-based terrain-following concept and error analysis,” *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 4, 2018, pp. 1006–1014.
- [9] National Imagery and Mapping Agency (NIMA), *Performance specification digital terrain elevation data (DTED)*, 2000. MIL-PRF-89020B.
- [10] United States Geological Survey (USGS), *USGS EarthExplorer*, (webpage accessed on February 10, 2024). URL <https://earthexplorer.usgs.gov/>.
- [11] Raymer, K., and Weingartner, T., “Advanced terrain data processor,” *AIAA/IEEE Digital Avionics Systems Conference. 13th DASC*, IEEE, 1994, pp. 636–639.



**Fig. 8 Example 3: Terrain following with previously undetected obstacle. The terrain is shown in black, the commanded altitude in red dashed, and the aircraft altitude in blue. Note that the commanded altitude is the terrain altitude increased by a safety height of 50 m. The inset shows a zoomed-in portion of the plot where the previously undetected obstacle appears.**

- [12] Choi, D.-Y., Hahn, S., Jang, D.-S., Lee, H., Lee, S., and Park, H., "Development of a terrain scan data generation algorithm using DTED," *Asia-Pacific International Symposium on Aerospace Technology*, Springer, 2021, pp. 1111–1124.
- [13] Yoo, M., Kang, S., Lee, H., Jang, D.-S., and Lee, S., "A study on a terrain scan data generation algorithm using digital terrain elevation data (DTED)," *International Journal of Aeronautical and Space Sciences*, Vol. 24, No. 5, 2023, pp. 1418–1429.
- [14] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, John Wiley & Sons, 2015.
- [15] Kwon, W., and Han, S., *Receding Horizon Control: Model Predictive Control for State Models*, Springer, 2006.
- [16] Camacho, E. F., and Bordons, C., *Model Predictive Control*, 2<sup>nd</sup> ed., Springer, 2007. doi:10.1007/978-0-85729-398-5.
- [17] Di Cairano, S., and Kolmanovsky, I. V., "Real-time optimization and model predictive control for aerospace and automotive applications," *Proc. Amer. Contr. Conf.*, 2018, pp. 2392–2409. doi:10.23919/ACC.2018.8431585.
- [18] 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. doi:10.2514/1.G002507.
- [19] Lu, P., and Pierson, B. L., "Aircraft terrain following based on a nonlinear continuous predictive control approach," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 4, 1995, pp. 817–823.
- [20] Williams, P., "Aircraft trajectory planning for terrain following incorporating actuator constraints," *Journal of Aircraft*, Vol. 42, No. 5, 2005, pp. 1358–1361.
- [21] Williams, P., "Real-time computation of optimal three-dimensional aircraft trajectories including terrain-following," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6603.
- [22] 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.



**Fig. 9 Example 3: Terrain following with previously undetected obstacle.** (a) shows the achieved and commanded airspeeds; (b) shows the aircraft altitude above ground level and the soft constraint on  $h$ , which is 37.5 m above the terrain; (c) shows normal acceleration and its soft constraints; (d) shows the angle of attack  $\alpha$ ; (e) shows the achieved elevator deflection; (f) shows the achieved and commanded pitch rates; (g) shows the engine thrust as a percentage of the total thrust, which is equal to  $100\delta_t$ ; (h) shows  $\theta_k$ , the PCAC estimated coefficients, where PCAC is initialized with no prior modeling information.

- [23] Nguyen, L. T., Ogburn, M. E., Gilbert, W. P., Kibler, K. S., Brown, P. W., and Deal, P. L., "Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability," Tech. rep., NASA, Washington, D.C., December 1979. NASA Tech. Paper 1538.
- [24] Sonneveldt, L., "Nonlinear F-16 model description," Tech. rep., Control and Simulation Div., Faculty of Aerospace Engineering, Delft Univ. of Technology, The Netherlands, June 2006. Ver. 3.0.
- [25] 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. doi:10.1007/BF02551362.
- [26] Nesterov, Y., *Introductory Lectures on Convex Optimization: A Basic Course*, Springer, New York, 2004.