Aerodynamic Flow Control Using Active Bleed Actuation with Variable-Channel Width

Harsh Kumar*, Jacob C. Vander Schaaf†,Syed Aseem Ul Islam‡, Krzysztof J. Fidkowski§, and Dennis S. Bernstein¶
University of Michigan, Ann Arbor, Michigan, 48109

Active bleed actuation integrated within an airfoil facilitates aerodynamic control. The present study investigates the ability of predictive cost adaptive control (PCAC) to use active bleed actuation with variable-channel width for flow control and compares its performance with PI control as well as with PCACi, which is an extension of PCAC that incorporates integral action. The fluid dynamics of the airfoil and variable-width channel are modeled using computational fluid dynamics (CFD). As an extension of model predictive control, PCAC leverages recursive least squares (RLS) with variable-rate forgetting (VRF) to facilitate real-time, closed-loop system identification without the need for analytical modeling or offline learning. The proposed approach uses flow control enabled by active bleed actuation to follow pressure setpoint commands at a specified location on the airfoil.

I. Introduction

Bleeding air through a channel in a body causes pressure redistribution, while varying the flow through the channel provides a means for actively controlling the aerodynamics of the body [1–4]. The present paper uses active bleed actuation with variable channel width to regulate pressure distribution around an airfoil. The control technique used in this work is predictive cost adaptive control (PCAC). PCAC was developed in [5, 6], and it was used for active flow control in [7], where the control objective is to achieve flow separation at a specified location on an airfoil.

As an indirect adaptive control extension of model predictive control (MPC), PCAC uses recursive least squares (RLS) with variable-rate forgetting (VRF) for online, closed-loop system identification [8–12]. At each time step, RLS-based system identification updates the coefficients of an input-output model whose order is specified by the user. To determine control inputs within the framework of MPC, PCAC uses quadratic programming for receding-horizon optimization, which enforces magnitude- and rate-saturation constraints as well as output constraints. To enable output-feedback control with physically realistic sensors, PCAC uses the block-observable canonical form (BOCF) realization, which provides an exact state estimation at each step [13]. PCAC can be used with a reference model, and unknown disturbances may be either matched or unmatched. During operation, PCAC uses sampled sensor data (e.g., flow state and altitude) and requests system inputs, such as mesh motion, surface deflections, and fluidic-jet velocities. Other than specification of the model order, optimization horizon, and forgetting parameters, PCAC operates under cold-start conditions without the need for analytical modeling or offline learning. The effectiveness of PCAC is compared to that of PI control and PCACi, an extension of PCAC with integral action.

The contribution of the present paper is a numerical investigation of the ability of PCAC to use active bleed actuation with variable channel width. In particular, PCAC is used to enable active flow control, where the objective is to regulate the pressure around an airfoil under high-Reynolds-number flight conditions. PCAC is distinct from active flow control methods that depend on detailed modeling of the fluid dynamics, either analytically or experimentally, often followed by model reduction. [14–24]. Controller-synthesis techniques for flow control include optimal control [25, 26], robust control [27], and machine-learning methods [23, 28]. In contrast, PCAC relies entirely on a model that is identified online, and thus requires no reduced-order model of any kind, either extracted from offline data or a high-order analytical model.

In the present paper, bleed actuation is realized by a CFD model that captures the fluid dynamics of a variable-width bleed channel through the airfoil. The CFD simulations are implemented in xflow, which is a high-order, mesh-adaptive CFD code [29], efficient for obtaining high-accuracy results for practical aerodynamic simulations. Using this actuation

*Master's Student, Department of Aerospace Engineering, kharsh@umich.edu.
†Ph.D. Candidate, Department of Aerospace Engineering, jacobcvs@umich.edu.
‡Research Fellow, Department of Aerospace Engineering, aseemisl@umich.edu.
§Professor, Department of Aerospace Engineering, kfid@umich.edu.
¶Professor, Department of Aerospace Engineering, dsbaero@umich.edu.

method, PCAC is shown to regulate the pressure measured at a sensor location on the airfoil with no prior model available to the controller.

II. Review of Predictive Cost Adaptive Control

As shown in Figure 1, PCAC combines online identification with output-feedback MPC. The PCAC algorithm is presented in this section. Subsection II.A describes the technique used for online identification, namely, recursive least squares. Subsection II.B reviews the output-feedback MPC technique for receding-horizon optimization based on [30]. Note that the model identified by PCAC is generally only good enough to facilitate control objectives. That is, in general the model identified by PCAC is unsuitable for model-based control design. Subsection II.C reviews PCACi, which is a variation of PCAC with integral action.

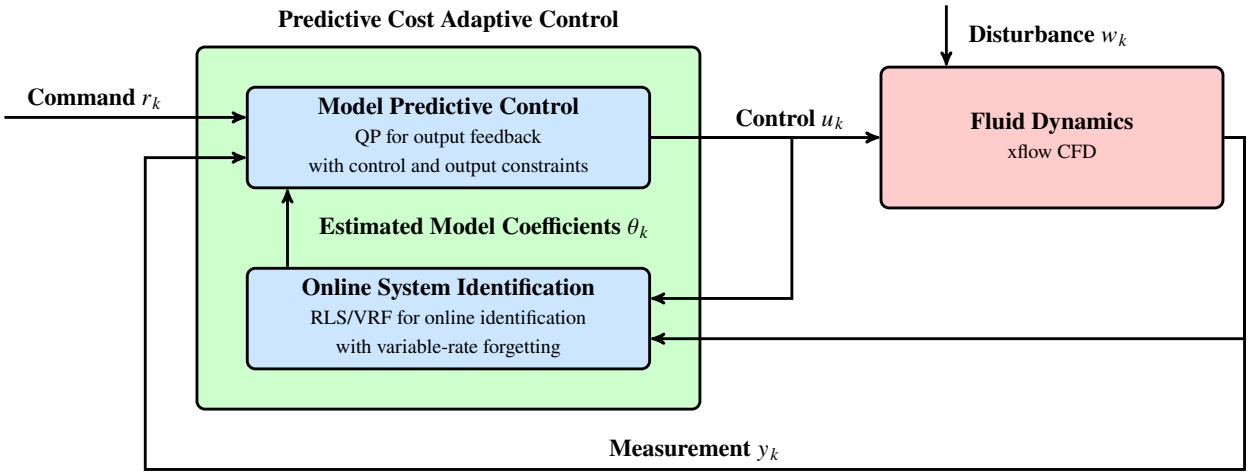

Fig. 1 PCAC block diagram. The online, closed-loop system identification is based on recursive least squares (RLS) with variable-rate forgetting (VRF). The model predictive control (MPC) algorithm, which is based on quadratic programming (QP), uses the estimated model coefficients θ_k to predictive function model, which is used by QP to determine the control input u_k.

A. Online Identification Using Recursive Least Squares

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

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

where

$$y_{-\hat{n}} = \cdots = y_{-1} = 0, \tag{2}$$

$$u_{-\hat{n}} = \cdots = u_{-1} = u_0 = 0. \tag{3}$$

For online identification, RLS is used to estimate the coefficients of the input-output model (1). To do this, RLS minimizes the cumulative cost

$$J_k(\hat{\theta}) = \sum_{i=0}^{k} \frac{\rho_i}{\rho_k} z_i^{\mathrm{T}}(\hat{\theta}) z_i(\hat{\theta}) + \frac{1}{\rho_k}(\hat{\theta} - \theta_0)^{\mathrm{T}} P_0^{-1} (\hat{\theta} - \theta_0), \tag{4}$$

where, for all $k \geq 0$, $\rho_k \triangleq \prod_{j=0}^{k} \lambda_j^{-1} \in \mathbb{R}$, $\lambda_k \in (0, 1]$ is the forgetting factor, $P_0 \in \mathbb{R}^{[\hat{n}p(m+p)+mp]\times[\hat{n}p(m+p)+mp]}$ 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

$$z_k(\theta_k) \triangleq y_k + \sum_{i=1}^{\hat{n}} \hat{F}_{i,k} y_{k-i} - \sum_{i=0}^{\hat{n}} \hat{G}_{i,k} u_{k-i}, \tag{5}$$

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

$$\theta_k \triangleq \operatorname{vec} \begin{bmatrix} \hat{F}_{1,k} & \cdots & \hat{F}_{\hat{n},k} & \hat{G}_{0,k} & \cdots & \hat{G}_{\hat{n},k} \end{bmatrix}. \tag{6}$$

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

$$\phi_k \triangleq \begin{bmatrix} -y_{k-1}^{\mathrm{T}} & \cdots & -y_{k-\hat{n}}^{\mathrm{T}} & u_k^{\mathrm{T}} & \cdots & u_{k-\hat{n}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \otimes I_p, \tag{7}$$

it follows that the performance variable (5) can be rewritten as

$$z_k(\theta_k) = y_k - \phi_k \theta_k. \tag{8}$$

Note that, with (8), the cost function (4) is strictly convex and quadratic, and thus has a unique global minimizer. The unique global minimizer is computed by RLS using

$$L_k = \lambda_k^{-1} P_k, \tag{9}$$

$$P_{k+1} = L_k - L_k \phi_k^{\mathrm{T}} (I_p + \phi_k L_k \phi_k^{\mathrm{T}})^{-1} \phi_k L_k, \tag{10}$$

$$\theta_{k+1} = \theta_k + P_{k+1} \phi_k^{\mathrm{T}} (y_k - \phi_k \theta_k). \tag{11}$$

Note that θ_{k+1} computed using (11) is available at step k, and thus, $\hat{F}_{1,k+1}, \ldots, \hat{F}_{\hat{n},k+1}, \hat{G}_{0,k+1}, \ldots, \hat{G}_{\hat{n},k+1}$ are available at step k.

The step-dependent parameter λ_k is the *forgetting factor.* In the case where λ_k is constant, RLS uses *constant-rate forgetting* (CRF); otherwise, RLS uses *variable-rate forgetting* (VRF) [31]. For VRF, λ_k is given by

$$\lambda_k = \frac{1}{1 + \eta g(z_{k-\tau_{\mathrm{d}}}, \ldots, z_k) \mathbf{1}[g(z_{k-\tau_{\mathrm{d}}}, \ldots, z_k)]}, \tag{12}$$

where $\mathbf{1}\colon \mathbb{R} \to \{0, 1\}$ is the unit step function, where $\mathbf{1}(x) = 0$ for all $x < 0$ and $\mathbf{1}(x) = 1$ for $x \geq 0$, and

$$g(z_{k-\tau_{\mathrm{d}}}, \ldots, z_k) \triangleq \frac{\sqrt{\frac{1}{\tau_{\mathrm{n}}} \sum_{i=k-\tau_{\mathrm{n}}}^{k} z_i^{\mathrm{T}} z_i}}{\sqrt{\frac{1}{\tau_{\mathrm{d}}} \sum_{i=k-\tau_{\mathrm{d}}}^{k} z_i^{\mathrm{T}} z_i}} - 1. \tag{13}$$

B. Model Predictive Control (MPC)

We define the tracking output $y_{\mathrm{t},k} \in \mathbb{R}$ a

$$y_{\mathrm{t},k} \triangleq C_{\mathrm{t}} y_k. \tag{14}$$

The performance objective is to have $y_{\mathrm{t},k} \in \mathbb{R}^{p_{\mathrm{t}}}$ follow a commanded sequence of $r_k \in \mathbb{R}^{p_{\mathrm{t}}}$, whose future values may or may not be known. In addition to the performance objective, the constrained output $y_{\mathrm{c},k} \in \mathbb{R}^{p_{\mathrm{c}}}$ is defined by

$$y_{\mathrm{c},k} \triangleq C_{\mathrm{c}} y_k, \tag{15}$$

where $C_{\mathrm{c}} \in \mathbb{R}^{p_{\mathrm{c}} \times p}$. The objective is to enforce the inequality constraint

$$C y_{\mathrm{c},k} + \mathcal{D} \leq 0_{n_{\mathrm{c}} \times 1}, \tag{16}$$

where $C \in \mathbb{R}^{n_{\mathrm{c}} \times p_{\mathrm{c}}}$ and $\mathcal{D} \in \mathbb{R}^{n_{\mathrm{c}}}$. Note that (16), where "$\leq$" is interpreted component-wise, defines a convex set.

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

$$u_{\min} \leq u_k \leq u_{\max}, \tag{17}$$

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

$$\Delta u_{\min} \leq u_k - u_{k-1} \leq \Delta u_{\max}, \tag{18}$$

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.

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

$$Y_{k,\ell} = \Gamma_k + T_k U_{k,\ell}. \tag{19}$$

where

$$Y_{k,\ell} \triangleq \begin{bmatrix} y_{k|1} \\ \vdots \\ y_{k|\ell} \end{bmatrix} \in \mathbb{R}^{\ell p}, \quad U_{k,\ell} \triangleq \begin{bmatrix} u_{k|1} \\ \vdots \\ u_{k|\ell} \end{bmatrix} \in \mathbb{R}^{\ell m}, \quad D_{\hat{n},k} \triangleq \begin{bmatrix} y_{k-\hat{n}+1} \\ \vdots \\ y_k \\ u_{k-\hat{n}+1} \\ \vdots \\ u_k \end{bmatrix} \in \mathbb{R}^{\hat{n}(p+m)}, \tag{20}$$

$$\Gamma_k \triangleq \begin{bmatrix} -F_{\mathrm{p},k}^{-1} F_{\mathrm{d},k} & F_{\mathrm{p},k}^{-1} G_{\mathrm{d},k} \end{bmatrix} D_{\hat{n},k}, \quad T_k \triangleq F_{\mathrm{p},k}^{-1} G_{\mathrm{p},k}, \tag{21}$$

$$F_{\mathrm{d},k} \triangleq \begin{bmatrix} \hat{F}_{\hat{n},k} & \cdots & \hat{F}_{1,k} \\ \vdots & \ddots & \vdots \\ 0_{p\times p} & \cdots & \hat{F}_{\hat{n},k} \\ 0_{p\times p} & \cdots & 0_{p\times p} \\ \vdots & \ddots & \vdots \\ 0_{p\times p} & \cdots & 0_{p\times p} \end{bmatrix} \in \mathbb{R}^{\ell p \times \hat{n} p}, \quad G_{\mathrm{d},k} \triangleq \begin{bmatrix} \hat{F}_{\hat{n},k} & \cdots & \hat{G}_{1,k} \\ \vdots & \ddots & \vdots \\ 0_{p\times m} & \cdots & \hat{G}_{\hat{n},k} \\ 0_{p\times m} & \cdots & 0_{p\times m} \\ \vdots & \ddots & \vdots \\ 0_{p\times m} & \cdots & 0_{p\times m} \end{bmatrix} \in \mathbb{R}^{\ell p \times \hat{n} m}, \tag{22}$$

$$F_{\mathrm{p},k} \triangleq \begin{bmatrix} I_p & \cdots & 0_{p\times p} & 0_{p\times p} & \cdots & 0_{p\times p} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \hat{F}_{\hat{n}-1,k} & \cdots & I_p & 0_{p\times p} & \cdots & 0_{p\times p} \\ \hat{F}_{\hat{n},k} & \cdots & \hat{F}_{1,k} & I_p & \cdots & 0_{p\times p} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0_{p\times p} & \cdots & \hat{F}_{\hat{n},k} & \hat{F}_{\hat{n}-1,k} & \cdots & I_p \end{bmatrix} \in \mathbb{R}^{\ell p \times \ell p}, \tag{23}$$

$$G_{\mathrm{p},k} \triangleq \begin{bmatrix} \hat{G}_{0,k} & \cdots & 0_{p\times m} & 0_{p\times m} & \cdots & 0_{p\times m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \hat{G}_{\hat{n}-1,k} & \cdots & \hat{G}_{0,k} & 0_{p\times m} & \cdots & 0_{p\times m} \\ \hat{G}_{\hat{n},k} & \cdots & \hat{G}_{1,k} & \hat{G}_{0,k} & \cdots & 0_{p\times m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0_{p\times m} & \cdots & \hat{G}_{\hat{n},k} & \hat{G}_{\hat{n}-1,k} & \cdots & \hat{G}_{0,k} \end{bmatrix} \in \mathbb{R}^{\ell p \times \ell m}. \tag{24}$$

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

$$\Delta U_{k,\ell} \triangleq [(u_{k|1} - u_k)^{\mathrm{T}} \ (u_{k|2} - u_{k|1})^{\mathrm{T}} \ \cdots \ (u_{k|\ell} - u_{k|\ell-1})^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{\ell m}. \tag{25}$$

Then, the receding horizon optimization problem is given by

$$\min_{U_{k,\ell}} \left(Y_{\mathrm{t},k,\ell} - \mathcal{R}_{k,\ell}\right)^{\mathrm{T}} Q \left(Y_{\mathrm{t},k,\ell} - \mathcal{R}_{k,\ell}\right) + \Delta U_{k,\ell}^{\mathrm{T}} R \Delta U_{k,\ell} + \varepsilon^{\mathrm{T}} S \varepsilon, \tag{26}$$

subject to

$$C_\ell Y_{k,\ell} + \mathcal{D}_\ell \le \varepsilon, \tag{27}$$

$$U_{\min} \le U_{k,\ell} \le U_{\max}, \tag{28}$$

$$\Delta U_{\min} \le \Delta U_{k,\ell} \le \Delta U_{\max}, \tag{29}$$

$$0_{\ell n_{\mathrm{c}} \times 1} \le \varepsilon, \tag{30}$$

where $Q \in \mathbb{R}^{\ell p_{\mathrm{t}} \times \ell p_{\mathrm{t}}}$ is the positive-definite output weighting, $R \in \mathbb{R}^{\ell m \times \ell m}$ is the positive definite control move-size weight, $S \in \mathbb{R}^{\ell n_{\mathrm{c}} \times \ell n_{\mathrm{c}}}$ 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 $C_\ell \triangleq I_\ell \otimes (CC_{\mathrm{c}}) \in \mathbb{R}^{\ell n_{\mathrm{c}} \times \ell p}$ and $\mathcal{D}_\ell \triangleq 1_{\ell \times 1} \otimes \mathcal{D} \in \mathbb{R}^{\ell n_{\mathrm{c}}}$. The quadratic program (QP) optimization (26)–(30) is solved using MATLAB's quadprog algorithm.

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 the matrices Γ_k and T_k. Then, the matrices Γ_k and T_k are used in a receding horizon optimization problem to solve for the ℓ-step controls $U_{k,\ell}$. The control input for the next step is then given by $u_{k|1}$, and the rest of the components of $U_{k,\ell}$ are discarded.

C. PCACi: PCAC with Integral Action

In order to include integrators in the control loop using PCAC, we define the sequence v_k to be the the backward difference of u_k. That is

$$v_k \triangleq u_k - u_{k-1}, \tag{31}$$

which implies

$$u_k = u_{k-1} + v_k. \tag{32}$$

That is, u_k is the integral of v_k. We reformulate the model predictive problem in terms of predicted values of v_k without modifying the procedure for online identification, which is from u_k to y_k.

We define

$$V_{k,\ell} \triangleq \begin{bmatrix} v_{k|1} \\ v_{k|2} \\ \vdots \\ v_{k|\ell} \end{bmatrix} \triangleq \begin{bmatrix} u_{k|1} - u_k \\ u_{k|2} - u_{k|1} \\ \vdots \\ u_{k|\ell} - u_{k|\ell-1} \end{bmatrix} \in \mathbb{R}^{\ell m}, \quad y_{\mathrm{a},k} \triangleq \begin{bmatrix} y_k \\ u_{k-1} \end{bmatrix} \in \mathbb{R}^{p+m}, \quad \hat{F}_{\mathrm{a},i,k} \triangleq \begin{bmatrix} \hat{F}_{i,k} & -\hat{G}_{i,k} \\ 0_{m \times p} & -I_i \end{bmatrix} \in \mathbb{R}^{(p+m)\times(p+m)}, \tag{33}$$

$$\hat{G}_{\mathrm{a},i,k} \triangleq \begin{bmatrix} \hat{G}_{i,k} \\ I_i \end{bmatrix} \in \mathbb{R}^{(p+m)\times m}, \quad \hat{F}_{\mathrm{a},0,k} \triangleq \begin{bmatrix} I_p & -\hat{G}_{0,k} \\ 0_{m \times p} & I_m \end{bmatrix} \in \mathbb{R}^{(p+m)\times(p+m)}, \quad I_i \triangleq \begin{cases} I_m, & i = 1, \\ 0_{m \times m} & \text{otherwise.} \end{cases} \tag{34}$$

Using these definitions the linear model (1) can be replaced with

$$\hat{F}_{\mathrm{a},0,k} \hat{y}_{\mathrm{a},k} = -\sum_{i=1}^{\hat{n}} \hat{F}_{\mathrm{a},i,k} y_{\mathrm{a},k-i} + \sum_{i=0}^{\hat{n}} \hat{G}_{\mathrm{a},i,k} v_{k-i}, \tag{35}$$

where $\hat{y}_{\mathrm{a},k} = \begin{bmatrix} \hat{y}_k^{\mathrm{T}} & u_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ is the estimated model output. Furthermore, it can be shown that (19) can be replaced with

$$Y_{\mathrm{a},k,\ell} = \Gamma_k + T_k V_{k,\ell}, \tag{36}$$

where

$$Y_{\mathrm{a},k,\ell} \triangleq \begin{bmatrix} y_{\mathrm{a},k|1}^{\mathrm{T}} & \cdots & y_{\mathrm{a},k|\ell}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{\ell(p+m)}, \tag{37}$$

by replacing I_p in (23) with $\hat{F}_{\mathrm{a},0,k}$, $\hat{G}_{0,k}$ in (24) with $\hat{G}_{\mathrm{a},0,k}$, for all $i = 1, \ldots, \hat{n}$, $\hat{F}_{i,k}$ and $\hat{G}_{i,k}$ in (22)–(24) with $\hat{F}_{\mathrm{a},i,k}$ and $\hat{G}_{\mathrm{a},i,k}$, respectively, and for all $i = 0, \ldots, \hat{n}-1$, y_{k-i} and u_{k-1} in (20) with $y_{\mathrm{a},k-i}$ and v_{k-i}. Finally, using (36) and $Y_{\mathrm{t},k,\ell} = \begin{bmatrix} C_{\mathrm{t},\ell} & 0_{p_{\mathrm{t}} \times m} \end{bmatrix} Y_{\mathrm{a},k,\ell}$, a modified receding horizon optimization problem can be stated as

$$\min_{V_{k,\ell}} \left(Y_{\mathrm{t},k,\ell} - \mathcal{R}_{k,\ell}\right)^{\mathrm{T}} Q \left(Y_{\mathrm{t},k,\ell} - \mathcal{R}_{k,\ell}\right) + \Delta V_{k,\ell}^{\mathrm{T}} R \Delta V_{k,\ell} + \varepsilon^{\mathrm{T}} S \varepsilon, \tag{38}$$

subject to

$$C_\ell Y_{k,\ell} + \mathcal{D}_\ell \leq \varepsilon, \tag{39}$$

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

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

$$0_{\ell n_{\mathrm{c}} \times 1} \leq \varepsilon, \tag{42}$$

where $R \in \mathbb{R}^{\ell m \times \ell m}$ is now the positive definite control-move, move-size weight. Note that the optimization variable for the cost (38) is $V_{k,\ell}$. Thus, the control input for the next step is then given by $u_{k|1} = u_k + v_{k|1}$, where $v_{k|1}$ is the first component of $V_{k,\ell}$ and the rest of the components of $V_{k,\ell}$ are discarded.

III. Simulation Setup

We consider a 2D NACA 4412 airfoil at a 15 degree angle of attack with a variable-width bleed channel through the airfoil geometry. The bleed channel allows the flow to pass through the airfoil from the leading edge to the upper surface, where the flow exits the channel approximately normal to the upper surface. The left and right boundaries of the domain are velocity inlets and outlets, respectively. The top and bottom boundaries are freestream conditions, and the airfoil boundaries are no-slip walls. The freestream velocity is non-dimensionalized, and the flow Reynolds number based on the chord length and viscosity is 10^5. The units for this problem are convenient $O(1)$ quantities in which one time unit corresponds to the time taken by flow moving at freestream speed to traverse the airfoil chord. Figure 2 shows the mesh and boundary conditions along with the location of the bleed channel, and Figure 3 shows the bleed channel partially constricted.

The fluid dynamics are simulated in xflow, which uses a discontinuous finite-element method to discretize the governing equation in space, and implicit multi-stage time integration for unsteady problems. The flow state is advanced using a time step of 0.03 s. The physical model consists of the Reynolds-Averaged Navier-Stokes equations with the Spalart-Allmaras closure. To vary the width of the bleed channel, an arbitrary Lagrangian-Eulerian (ALE) formulation is used [32, 33]. The relevant motion for this case is a blended rigid-body plunge, in which which the plunging region is centered at a point next to the channel, and in which a cubic radial blending function attenuates the plunge to zero deformation at a certain distance away from the center. The channel is in the blending region, and the disparity in deformation of the two sides of the channel allows for a variations in the channel width.

xflow is directly interfaced with MATLAB-based control codes using the MATLAB Engine API. At each controller timestep k, the interface obtains pressure measurements y_k from a location the flow field upstream of the bleed channel as indicated in Figure 3. This measurement is provided to the controller, which computes the control inputs u_{k+1}. The interface then sends u_{k+1} to xflow, which applies the control input by specifying the mesh motion of the channel to vary its width.

IV. Numerical Results

In this section, the flow control objective is to vary the bleed-channel width to achieve a pressure setpoint at the sensor location. Three controllers are tested: PI, PCAC, and PCACi. In each example, the controller specifies the width

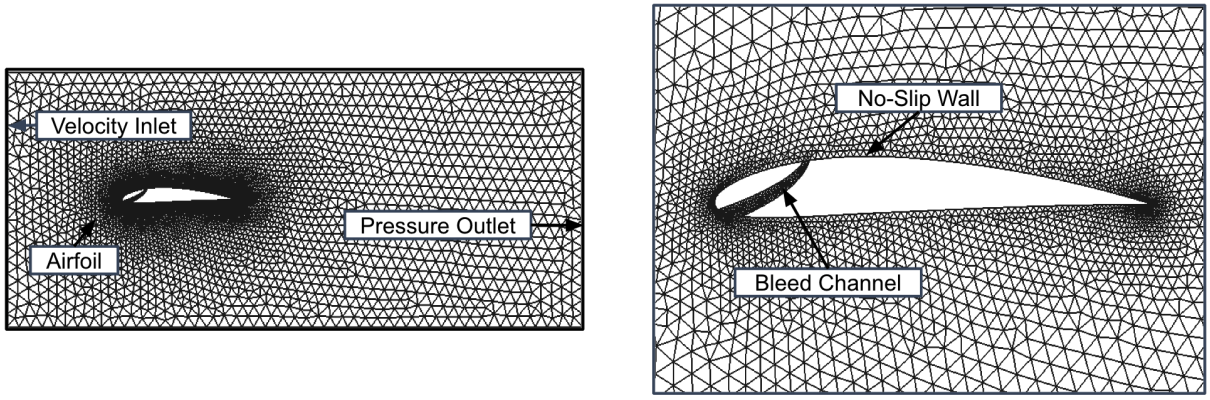

Fig. 2 Full case mesh (left) and mesh zoomed to airfoil region showing the bleed channel (right). PCAC specifies the width of this channel to vary the amount of bleed flow.

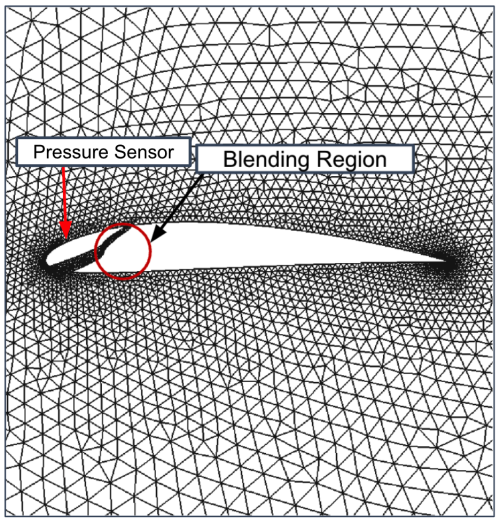

Fig. 3 Mesh zoomed to airfoil region. The mesh motion constricts the width of the bleed channel in the region indicated by the red circle. The red arrow indicates the location of the pressure sensor.

of the bleed-channel to vary the amount of bleed flow. Since the flow is injected normally to the wall, the bleed induces flow separation and redistributes the pressure around the airfoil. By varying this bleed, we can control the pressure at the sensor location. A control-input constraint $0 \leq u_k \leq 0.5$ is enforced, where $u_k = 0$ represents the bleed channel fully open and $u_k = 0.5$ represents the bleed channel partially closed. We consider the pressure setpoint commands at a forward location on the airfoil. The bleed channel is initially fully open, and the resulting non-dimensionalized pressure measured at the sensor location is 17.205, and a pressure setpoint of 17.195 is commanded.

Example 1. *PI control of pressure.* In this example, a discrete-time PI controller uses samples of the pressure measurement provided at the rate of 0.06 s/sample. Figure 4 shows the pressure measurements y_k, the requested channel width u_k, and the error e_k. After approximately 20 s, the pressure approaches the setpoint.

Example 2. *PCAC control of pressure.* We reconsider Example 1 using PCAC. Figure 5 shows the sensor measurements, error, requested channel width, model coefficients estimated by RLSID, and RLS forgetting factor.

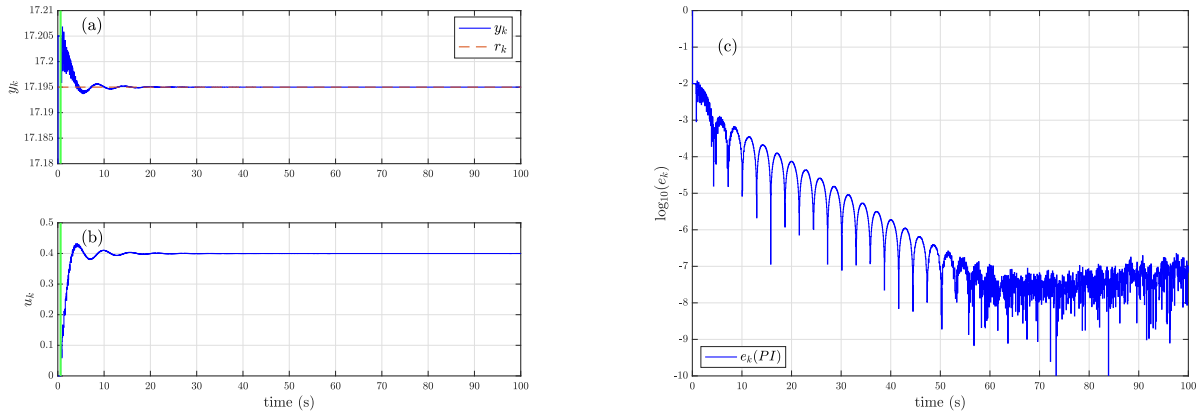

Fig. 4 Example 1: PI control is enabled at 0.6 s. (a) shows that pressure measurement approaches the setpoint and that the initial transient damps out in approximately 20 s. (b) shows the PI control requested channel width and (c) shows the error.

Although the pressure approaches the setpoint, Figure 5 (b) shows that the rate of convergence with PCAC is slower than with PI control. Additionally, a large transient response is present over the first 10 s as PCAC identifies the system dynamics.

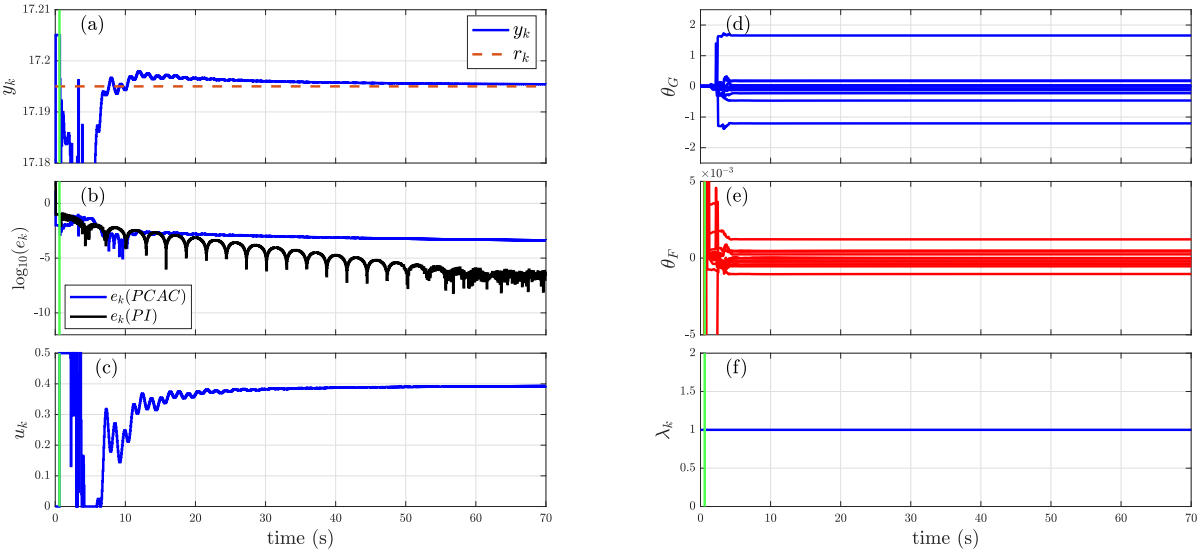

Fig. 5 Example 2: PCAC is enabled at 0.6 s (indicated by the green line). After the initial transient, (a) shows that the pressure measurement approaches the setpoint. (b) shows the error between the measurement and setpoint for both PCAC and PI, (c) shows the PCAC requested bleed channel-width, (d) shows the time histories of the numerator of the identified model, (e) shows the time histories of the denominator of the identified model, and (f) shows the variable rate forgetting factor, which is inactive.

Example 3. *PCACi control of pressure.* We reconsider Example 2 using PCACi. Figure 6 shows the pressure measurements, error, requested channel width, model coefficients estimated by RLSID, and RLS forgetting factor. The pressure measurements approaches the setpoint and have a smaller transient over the first 10 s than in Example 2.

Furthermore, Figure 6 b shows the error between the pressure measurement and setpoint along with the error from Examples 1 and 2. PCACi has a convergence rate between that of PI and PCAC.

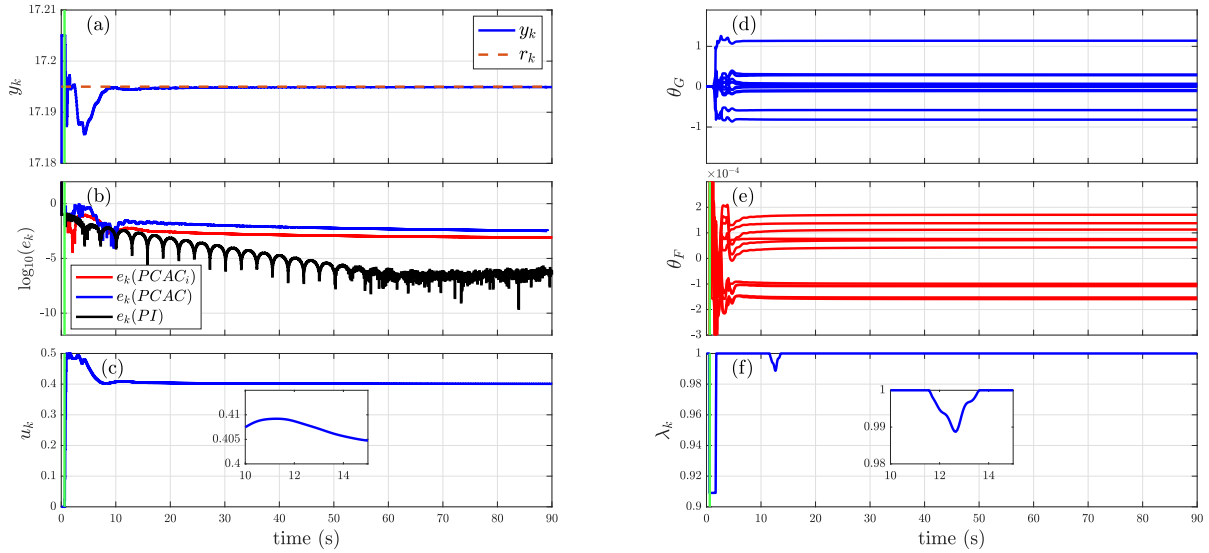

Fig. 6 Example 3: PCACi is enabled at 0.6 s (indicated by the green line). After the initial transient, (a) shows that the pressure measurement approaches the setpoint. (b) shows the error of PCACi with PCAC and PI controller, (c) shows the PCACi requested channel width, the inset shows the effect on channel width due to the forgetting being active, (d) shows the time histories of the numerator of the identified model, (e) shows the time histories of the denominator of the identified model , and (f) shows the variable rate forgetting factor, which is active between approximately 11 and 14 s.

V. Conclusions

This paper used predictive cost adaptive control (PCAC) for bleed- actuated control of the pressure on an airfoil. A CFD simulation of a variable-width channel was use to assess the effect of active flow control. Without prior modeling, PCAC followed the commanded pressure setpoints at the sensor location by modifying the bleed channel thickness. Furthermore, the inclusion of integral action within PCAC provided better transient response and faster convergence to the pressure setpoint. Though PI control resulted in a better rate of convergence, it required more effort to to tune since the gains must be adjusted for the case. Additionally, the PI gains must be retuned if the airfoil angle of attack or flow Reynolds number were to change. PCAC requires the selections of weights and can adapt to changing angles of attack and Reynolds numbers. Future work will explore the effect of external disturbances as well as the ability of PCAC to adapt to changing angles of attack and Reynolds numbers as compared to PI control. Finally, rigid-body dynamics will be included to simulate the use of PCAC and bleed actuation to control the plunge, pitch, and surge motion of the airfoil.

References

[1] Kearney, J., and Glezer, A., "Aerodynamic Control Using Distributed Bleed," *6th AIAA Flow Control Conference*, New Orleans, LA, 2012, pp. 1–17. AIAA 2012-3246.

[2] Kearney, J., and Glezer, A., "Aero-Effected Control of a Pitching Airfoil by Bleed Actuation," *31st AIAA Applied Aerodynamics Conference*, San Diego, CA, 2013, pp. 1–10. AIAA 2013-2519.

[3] Kearney, J., and Glezer, A., "Aerodynamic control of a Pitching Airfoil by Active Bleed," *32nd AIAA Applied Aerodynamics Conference*, Atlanta, GA, 2014, pp. 1–11. AIAA 2014-2045.

[4] Kestel, K., Ramazanlı, B., and Yavuz, M. M., "Control of flow structure over a non-slender delta wing using passive bleeding," *Aerospace Science and Technology*, Vol. 106, 2020, pp. 1–12.

[5] 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.

[6] 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.

[7] Vander Schaaf, J., Lu, Q., Fidkowski, K. J., and Bernstein, D. S., "Data-Driven Model Predictive Control of Airfoil Flow Separation," *Proc. Amer. Contr. Conf.,*, Toronto, Canada, 2024.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] Bewley, T. R., "Flow control: new challenges for a new renaissance," *Progress in Aerospace sciences*, Vol. 37, No. 1, 2001, pp. 21–58.

[15] Aamo, O. M., and Krstic, M., *Flow Control by Beedback: Stabilization and Mixing*, Springer, 2003.

[16] Gunzburger, M. D., *Perspectives in Flow Control and Optimization*, SIAM, 2002.

[17] Gad-el Hak, M., *Flow Control: Passive, Active, and Reactive Flow Management*, Cambridge University Press, 2000.

[18] Joslin, R. D., and Miller, D. N. (eds.), *Fundamentals and Applications of Modern Flow Control*, AIAA, 2009.

[19] Barbu, V., *Stabilization of Navier–Stokes Flows*, Springer, 2011.

[20] Cattafesta III, L. N., and Sheplak, M., "Actuators for active flow control," *Ann. Rev. Fluid Mech.*, Vol. 43, 2011, pp. 247–272.

[21] Luhar, M., Sharma, A. S., and McKeon, B. J., "Opposition control within the resolvent analysis framework," *J. Fluid Mech.*, Vol. 749, 2014, p. 597–626.

[22] Brunton, S. L., and Noack, B. R., "Closed-loop turbulence control: Progress and challenges," *Appl. Mech. Rev.*, Vol. 67, No. 5, 2015.

[23] Duriez, T., Brunton, S. L., and Noack, B. R., *Machine Learning Control—Taming Nonlinear Dynamics and Turbulence*, Springer, 2017.

[24] Wang, J., and Feng, L., *Flow Control Techniques and Applications*, Cambridge University Press, 2019.

[25] Joshi, S. S., Speyer, J. L., and Kim, J., "A Systems Theory Approach to the Feedback Stabilization of Infinitesimal and Finite-Amplitude Disturbances in Plane Poiseuille Flow," *J. Fluid Mech.*, Vol. 332, 1997, pp. 157–184.

[26] Joshi, S. S., Speyer, J. L., and Kim, J., "Finite Dimensional Optimal Control of Poiseuille Flow," *J. Guid. Contr. Dyn.*, Vol. 22, 1999, pp. 340–348.

[27] Bewley, T. R., and Liu, S., "Optimal and robust control and estimation of linear paths to transition," *J. Fluid Mech.*, Vol. 365, 2001, pp. 305–349.

[28] Gautier, N., Aider, J.-L., Duriez, T., Noack, B., Segond, M., and Abel, M., "Closed-loop separation control using machine learning," *J. Fluid Mech.*, Vol. 770, No. 5, 2015, pp. 442–457.

[29] Fidkowski, K. J., "Output error estimation strategies for discontinuous Galerkin discretizations of unsteady convection-dominated flows," *International Journal for Numerical Methods in Engineering*, Vol. 88, No. 12, 2011, pp. 1297–1322. https://doi.org/10.1002/nme.3224.

[30] Clarke, D. W., Mohtadi, C., and Tuffs, P. S., "Generalized predictive control—Part I. The basic algorithm," *Automatica*, Vol. 23, No. 2, 1987, pp. 137–148.

[31] Bruce, A., Goel, A., and Bernstein, D. S., "Convergence and Consistency of Recursive Least Squares with Variable-Rate Forgetting," *Automatica*, Vol. 119, 2020, pp. 1–6. Article 109052.

[32] Persson, P.-O., Bonet, J., and Peraire, J., "Discontinuous Galerkin Solution of the Navier-Stokes Equations on Deformable Domains," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, 2009, pp. 1585–1595.

[33] Kast, S. M., and Fidkowski, K. J., "Output-based Mesh Adaptation for High Order Navier-Stokes Simulations on Deformable Domains," *Journal of Computational Physics*, Vol. 252, No. 1, 2013, pp. 468–494. https://doi.org/10.1016/j.jcp.2013.06.007.