2015 American Control Conference
Palmer House Hilton
July 1-3, 2015. Chicago, IL, USA

An Inner-Loop/Outer-Loop Architecture for an Adaptive Missile Autopilot

Frantisek M. Sobolic, Gerardo Cruz, and Dennis S. Bernstein

Abstract— **We use retrospective cost adaptive control (RCAC) with a constant forgetting factor (CFF), variable forgetting factor (VFF), and Kalman Filter (KF) to control a planar missile with nonlinear dynamics and aerodynamics. RCAC/CFF, RCAC/VFF, and RCAC/KF are used within an inner-loop/outer-loop control architecture, where the normal acceleration command is used to update the pitch-rate command for use by the pitch-rate servo loop. This control architecture is necessitated by the fact that, except for circular arcs with constant velocity, the appropriate pitch-rate command cannot be inferred from the normal acceleration command. Aggressive commands are used to compare RCAC/VFF and RCAC/KF with RCAC/CFF.**

I. INTRODUCTION

Autopilot design for high-performance missiles presents multiple challenges to control technology. Missiles typically fly through a wide range of Mach numbers, high angles of attack, and under high 'g' loading, leading to strongly nonlinear dynamics [10]. The standard approach to addressing these nonlinearities is to schedule the control gains based on the missile's aerodynamics as they vary during flight [1, 10].

Unfortunately, the standard approach to missile autopilot design requires aerodynamic lookup tables based on extensive wind tunnel test data, and thus is expensive and time-consuming. This situation motivates the use of adaptive control laws that compensate online for changing aerodynamics and uncertainty in the aerodynamic model. In order to successfully control a high-performance missile, however, an adaptive control law must adapt sufficiently quickly to rapidly varying commands from the missile guidance system. With these challenges in mind, the present paper extends [3, 5], where retrospective cost adaptive control (RCAC) was used to control the planar missile model described in [8]. This model has been used [13] to study the feasibility of various autopilot control schemes.

RCAC, which was developed in [6, 7, 12], has been shown to be effective on nonminimum-phase systems. This property is relevant to the planar missile model in [8], where the nose-mounted gyro and tail-fin actuation give rise to nonminimum-phase normal acceleration dynamics.

The goal of the present paper is to apply RCAC to the planar missile in [8] under more aggressive commands than were considered in [3, 5]. To address this objective, we extend the formulation in [3, 5] to include additional features, namely, variable forgetting factor (VFF) and Kalman Filter (KF) extensions of the recursive least squares (RLS) update of the controller gains. VFF techniques [2, 4] are used to adjust the forgetting factor λ in RLS based on the information provided by the latest measurement.

An additional novel element of the present paper is an inner-loop/outer-loop control architecture. Specifically, RCAC adaptively adjusts the pitch-rate command based on the normal acceleration command from guidance. This adaptation would not be needed if the missile were commanded to fly along a circular arc with a constant velocity. However, for arbitrary normal acceleration commands, the appropriate pitch-rate command cannot be inferred, and this motivates the use of RCAC/VFF and RCAC/KF to adaptively specify the appropriate pitch-rate command. The pitch-rate command is then used to drive an inner-loop controller that acts on the error between the pitch-rate command and the pitch-rate measurement. The pitch-rate loop architecture is a static gain proportional/integral controller that stabilizes the pitch rate about a single trim point.

II. PROBLEM FORMULATION AND NONLINEAR MISSILE MODEL

To intercept a moving target, the missile is equipped with an active seeker with guidance laws that provides the normal acceleration ('g') command that the missile must follow in order to reach its target. In this section we briefly describe the model used. See [8] for more details.

The target dynamics in an inertial frame are given by

$$\dot{X}_{\rm t} = V_{\rm t}(0)\cos\theta_{\rm t}(0), \quad \dot{Z}_{\rm t} = V_{\rm t}(0)\sin\theta_{\rm t}(0), \tag{1}$$

where $X_{\rm t}$ and $Z_{\rm t}$ are the inertial coordinates of the target and $V_{\rm t}(0)$, $\theta_{\rm t}(0)$, $X_{\rm t}(0)$, $Z_{\rm t}(0)$ are the target's initial conditions.

The nonlinear three-degree-of-freedom missile dynamics described in the body frame are given by

$$m\dot{U} = \sum F_{B_X} - mQW + T_{\rm Thrust}, \tag{2}$$

$$m\dot{W} = \sum F_{B_Z} + mQU, \tag{3}$$

$$I_Y\dot{Q} = \sum M_Y, \dot{\theta} = Q, \tag{4}$$

$$\dot{X} = U\cos\,\theta + W\sin\theta, \tag{5}$$

$$\dot{Z} = -U\sin\,\theta + W\cos\theta, \tag{6}$$

where $U(0) = U_0$, $W(0) = W_0$, $Q(0) = Q_0$, $\theta(0) = \theta_0$, $X(0) = X_0$, $Z(0) = Z_0$. Assuming a flat Earth, the moment

F. M. Sobolic and G.Cruz are graduate students in the Aerospace Engineering Department at the University of Michigan, 1320 Beal Ave., Ann Arbor, MI 48109, USA fsobolic@umich.edu, gecruz@umich.edu

D. S. Bernstein is a Professor with the Aerospace Engineering Department, University of Michigan, dsbaero@umich.edu

978-1-4799-8686-6/$31.00 ©2015 AACC

and forces about the center of gravity are

$$\sum F_{B_X} = \bar{q}SC_A - mg\sin\theta, \tag{7}$$

$$\sum F_{B_Z} = \bar{q}SC_N(\alpha, M, \delta_p) + mg\cos\theta, \tag{8}$$

$$\sum M_Y = \bar{q}SdC_m(\alpha, M, \delta_p, Q), \tag{9}$$

where $\bar{q} = \frac{1}{2}\rho V^2$ is the dynamic pressure.

The aerodynamic coefficients are modeled as functions of Mach number M, angle-of-attack α, fin deflection angle δ_p, and pitch rate Q, as

$$\begin{aligned} C_A &= a_a, \\ C_N &= a_n\alpha^3 + b_n\alpha|\alpha| + c_n\left(2 - \frac{M}{3}\right)\alpha + d_n\delta_p, \\ C_m &= a_m\alpha^3 + b_m\alpha|\alpha| + c_m\left(-7 + \frac{8M}{3}\right)\alpha + d_m\delta_p + e_mQ, \end{aligned}$$

where $\alpha \triangleq \tan^{-1}(W/U)$, $V^2 \triangleq U^2 + W^2$, $M \triangleq V/a$, where a is the altitude-dependent speed of sound. The missile model assumes planar flight, and thus only the X, Z coordinates and pitch are modeled, yielding a 6th-order model. Out-of-plane effects, such as roll angle, yaw angle, and sideslip are therefore fixed at zero. Additionally, the fin actuator is modeled as a second-order system. For realism, an actuator rate and magnitude saturations are implemented at 500 deg/sec and 30 deg, respectively. Finally, the normalized normal acceleration is given by $n_z = F_N/(gm) + \cos\theta$.

III. Three-Loop Autopilot (3LA)

The goal of the 3LA [9] is to minimize the error between the commanded normal acceleration, generated by the guidance law, and the normal acceleration measurement provided by the inertial measurement unit (IMU). The states available for measurement are the normal acceleration n_z and the pitch rate Q, which is provided by an on-board accelerometer and gyroscope, respectively. 3LA is implemented as

$$u(s) = K_{\mathrm{Q}}Q(s) + \frac{1}{s}\left(K_\theta Q(s) + K_{\mathrm{I}}\left[K_{\mathrm{SS}}n_{z,\mathrm{cmd}} - n_{z,\mathrm{IMU}}\right]\right),$$

where $K_{\mathrm{Q}}, K_\theta, K_{\mathrm{I}}$, and K_{SS} are the control gains determined by modeling and analysis. Each gain is scheduled on a trim condition based on the missile angle-of-attack and Mach number. In practice, a digital version of the control law is implemented in discrete time.

IV. RCAC Controller

Consider the discrete-time system

$$x(k+1) = Ax(k) + Bu(k) + D_1w(k), \tag{10}$$

$$y(k) = Cx(k) + D_2w(k), \tag{11}$$

$$z(k) = E_1x(k) + E_0w(k), \tag{12}$$

where $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^{l_y}$, $z(k) \in \mathbb{R}^{l_z}$, $u(k) \in \mathbb{R}^{l_u}$, $w(k) \in \mathbb{R}^{l_w}$, and $k \geq 0$. The objective of the adaptive controller is to generate a control signal $u(k)$ that minimizes the performance variable $z(k)$ in the presence of exogenous signals $w(k)$. The exogenous signal $w(k)$ can represent a command signal to be tracked, an external disturbance, or both. The system (10) – (12) can represent a sampled-data system based on continuous-time dynamics with sample and hold operations.

A. The Control Law

Let the control signal be constructed as a strictly proper dynamic compensator of order n_{c} given by

$$u(k) = \sum_{i=1}^{n_c} T_i(k)u(k-i) + \sum_{i=1}^{n_c} S_i(k)y'(k-i), \tag{13}$$

where, for all $i = 1, \ldots, n_c$, $T_i \in \mathbb{R}^{l_u \times l_u}$ and $S_i \in \mathbb{R}^{l_u \times l_{y'}}$ are the gain matrices. The signal $y(k)'$ is usually chosen to be either the output $y(k)$ or the performance $z(k)$. The controller in (13) can be rewritten as

$$u(k) = \Phi(k)\theta(k), \tag{14}$$

where $\Phi(k)$ is the regressor matrix

$$\Phi(k) \triangleq I_{l_u} \otimes \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n_c) \\ y'(k-1) \\ \vdots \\ y'(k-n_c) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{l_u \times l_\theta}, \tag{15}$$

where $\otimes$ is the Kronecker Product and $\theta(k) = \mathrm{vec}\,[T(k)\ S(k)] \in \mathbb{R}^{l_\theta}$ is the vector of the controller gains with size $l_\theta = n_c l_u(l_u + l_{y'})$.

B. Retrospective Performance

We define the retrospective performance variable as

$$\hat{z}(\hat{\theta}(k), k) \triangleq z(k) + G_{\mathrm{f}}(\mathbf{q})[\Phi(k)\hat{\theta}(k) - u(k)], \tag{16}$$

$$G_{\mathrm{f}}(\mathbf{q}) \triangleq \mathcal{D}^{-1}(\mathbf{q})\mathcal{N}(\mathbf{q}) \tag{17}$$

where

$$\mathcal{N}(\mathbf{q}) \triangleq \sum_{i=1}^{n_{\mathrm{f}}} N_i\mathbf{q}^{n_{\mathrm{f}}-i},\ \mathcal{D}(\mathbf{q}) \triangleq I_{l_z}\mathbf{q}^{n_{\mathrm{f}}} - \sum_{i=1}^{n_{\mathrm{f}}} D_i\mathbf{q}^{n_{\mathrm{f}}-i}, \tag{18}$$

with $N_i \in \mathbb{R}^{l_z \times l_u}$ and $D_i \in \mathbb{R}^{l_z \times l_z}$ and $\mathbf{q}$ represents the forward shift operator. The filter G_{f} is of order $n_{\mathrm{f}} \geq 1$ and each polynomial entry of $\mathcal{D}(\mathbf{q})$ is asymptotically stable. Next, we are able to rewrite (16) as

$$\hat{z}(\hat{\theta}(k), k) = z(k) + \Phi_{\mathrm{f}}(k)\hat{\theta}(k) - u_{\mathrm{f}}(k), \tag{19}$$

where the filtered regressor and control are given by

$$\begin{aligned} \Phi_{\mathrm{f}}(k) &\triangleq G_{\mathrm{f}}(\mathbf{q})\Phi(k), \\ &= \sum_{i=1}^{n_{\mathrm{f}}} \left[N_i\Phi(k-i) + D_i\Phi_{\mathrm{f}}(k-i)\right], \end{aligned} \tag{20}$$

$$\begin{aligned} u_{\mathrm{f}}(k) &\triangleq G_{\mathrm{f}}(\mathbf{q})u(k), \\ &= \sum_{i=1}^{n_{\mathrm{f}}} \left[N_iu(k-i) + D_iu_{\mathrm{f}}(k-i)\right], \end{aligned} \tag{21}$$

and $\hat{\theta}(k)$ is determined by the optimization below. In the present paper, G_{f} is chosen to be a finite-impulse-response (FIR) filter, which implies that the coefficients $D_i = 0$. The choice of G_{f} is discussed in [11].

C. Cumulative Cost

Define the cumulative cost function to be minimized as

$$
\begin{aligned}
J(\hat{\theta}(k), k) \triangleq & \sum_{i=k_0}^{k} \hat{z}(i)^{\mathrm{T}} R_z(i) \hat{z}(i) \\
&+ \sum_{i=k_0}^{k} [\Phi(i)\hat{\theta}(k)]^{\mathrm{T}} R_u(i) [\Phi(i)\hat{\theta}(k)] \\
&+ [\hat{\theta}(k) - \hat{\theta}(0)]^{\mathrm{T}} R_\theta(k) [\hat{\theta}(k) - \hat{\theta}(0)], \qquad (22)
\end{aligned}
$$

where $R_z(i), R_u(i)$, and $R_\theta(k)$ are positive definite for all k. Here we assume that the weighting matrices are constant. Since (22) is a strictly convex function, its minimizer can be found by computing the partial derivative of $J(\hat{\theta}(k), k)$ with respect to $\hat{\theta}(k)$ and obtaining

$$
\begin{aligned}
\frac{\partial J(\hat{\theta}(k), k)}{\partial \hat{\theta}(k)} &= 2 \sum_{i=k_0}^{k} [\hat{z}(i)^{\mathrm{T}} R_z \Phi_{\mathrm{f}}(i) + [\Phi(i)\hat{\theta}(k)]^{\mathrm{T}} R_u \Phi(i)] \\
&+ 2[\hat{\theta}(k) - \hat{\theta}(0)]^{\mathrm{T}} R_\theta = 2\mathcal{A}(k)^{\mathrm{T}} + 2\hat{\theta}(k)^{\mathrm{T}} \mathcal{P}(k)^{-1}, \quad (23)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathcal{A}(k) &\triangleq \sum_{i=k_0}^{k} \left(\Phi_{\mathrm{f}}(i)^{\mathrm{T}} R_z \left[z(i) - u_{\mathrm{f}}(i) \right] \right) - R_\theta \hat{\theta}(0), \\
\mathcal{P}(k) &\triangleq [\sum_{i=k_0}^{k} (\Phi_{\mathrm{f}}^{\mathrm{T}}(i) R_z \Phi_{\mathrm{f}}(i) + \Phi(i)^{\mathrm{T}} R_u \Phi(i)) + R_\theta]^{-1}.
\end{aligned}
$$

Therefore, $\hat{\theta}(k) = -\mathcal{P}(k)\mathcal{A}(k)$.

D. RCAC Update Laws

In this section, we design an update law for obtaining the controller parameters $\hat{\theta}(k)$. The recursive least squares (RLS) algorithm is first derived followed by the modifications to RLS which include the constant forgetting factor (CFF), the varable forgetting factor (VFF), and the Kalman Filter (KF).

1) Recursive Least Squares (RLS): To find a recursive solution for the controller parameters, we start with

$$
\begin{aligned}
\mathcal{A}(k) &= \mathcal{A}(k-1) + \Phi_{\mathrm{f}}(k)^{\mathrm{T}} R_z \left[z(k) - u_{\mathrm{f}}(k) \right] \\
&= \mathcal{A}(k-1) + X(k)^{\mathrm{T}} \bar{R}\, \bar{z}(k), \qquad (24)
\end{aligned}
$$

where $\mathcal{A}(0) = -R_\theta \hat{\theta}(0)$ and

$$
\begin{aligned}
\mathcal{P}(k) &= \left[\mathcal{P}(k-1)^{-1} + \Phi_{\mathrm{f}}^{\mathrm{T}}(k) R_z \Phi_{\mathrm{f}}(k) + \Phi(k)^{\mathrm{T}} R_u \Phi(k) \right]^{-1} \\
&= \mathcal{P}(k-1) - \mathcal{P}(k-1) X(k)^{\mathrm{T}} \Gamma(k)^{-1} X(k) \mathcal{P}(k-1), \qquad (25)
\end{aligned}
$$

$$
X(k) \triangleq \begin{bmatrix} \Phi_{\mathrm{f}}(k) \\ \Phi(k) \end{bmatrix} \in \mathbb{R}^{(l_z + l_u) \times l_\theta}, \qquad (26)
$$

$$
\bar{R} \triangleq \begin{bmatrix} R_z & 0 \\ 0 & R_u \end{bmatrix} \in \mathbb{R}^{(l_z + l_u) \times (l_z + l_u)}, \qquad (27)
$$

$$
\bar{z}(k) \triangleq \begin{bmatrix} z(k) - u_{\mathrm{f}}(k) \\ 0 \end{bmatrix} \in \mathbb{R}^{l_z + l_u}, \qquad (28)
$$

$$
\Gamma(k) \triangleq \bar{R}^{-1} + X(k) \mathcal{P}(k-1) X(k)^{\mathrm{T}}, \qquad (29)
$$

and $\mathcal{P}(0) \triangleq R_\theta^{-1}$. Combining (24) and (25) yields

$$
\begin{aligned}
\hat{\theta}(k) &= -\mathcal{P}(k)\mathcal{A}(k), \\
&= \hat{\theta}(k-1) - \mathcal{P}(k-1) X(k)^{\mathrm{T}} \Gamma(k)^{-1} \left[X(k) \theta(k-1) \right. \\
&\quad \left. - X(k) \mathcal{P}(k-1) X(k)^{\mathrm{T}} \bar{R} \bar{z}(k) + \Gamma(k) \bar{R} \bar{z}(k) \right], \\
&= \hat{\theta}(k-1) + \mathcal{P}(k-1) X(k)^{\mathrm{T}} \Gamma(k)^{-1} \epsilon(k), \qquad (30)
\end{aligned}
$$

where $\hat{\theta}(0) = \hat{\theta}_0$ and $\epsilon(k) \triangleq -\bar{z}(k) - X(k)\hat{\theta}(k-1)$.

2) RCAC with Constant Forgetting Factor (CFF): The recursive algorithm based on RLS is modified to include a constant forgetting factor by redefining the weight matrices

$$
R_z(k,i) = \lambda^{k-i} R_z, \quad R_u(k,i) = \lambda^{k-i} R_u, \quad R_\theta(k) = \lambda^k R_\theta,
$$

where $\lambda \in (0, 1]$. The recursion on (24) and (25) is then

$$
\begin{aligned}
\mathcal{A}(k) &= \lambda \mathcal{A}(k-1) + \Phi_{\mathrm{f}}(k)^{\mathrm{T}} R_z(k) \left[z(k) - u_{\mathrm{f}}(k) \right], \quad (31) \\
\mathcal{P}(k) &= \left[\lambda \mathcal{P}(k-1)^{-1} + X(k)^{\mathrm{T}} \bar{R}(k) X(k) \right]^{-1}, \qquad (32)
\end{aligned}
$$

which yields the recursive update

$$
\hat{\theta}(k) = \hat{\theta}(k-1) + \mathcal{P}(k-1) X(k)^{\mathrm{T}} \tilde{\Gamma}(k)^{-1} \epsilon(k), \qquad (33)
$$

where $\tilde{\Gamma}(k) \triangleq \lambda \bar{R}(k)^{-1} + X(k) \mathcal{P}(k) X(k)^{\mathrm{T}}$.

3) Variable Forgetting Factor (VFF): Following [4] and [2], we define

$$
\begin{aligned}
K(k) &\triangleq \mathcal{P}(k) X(k)^{\mathrm{T}} \Gamma(k)^{-1}, \qquad (34) \\
\mathcal{E}(k) &\triangleq \epsilon(k)^{\mathrm{T}} \left(I - X(k) K(k) \right) \epsilon(k), \qquad (35) \\
W(k) &\triangleq \mathcal{P}(k) - K(k) X(k) \mathcal{P}(k). \qquad (36)
\end{aligned}
$$

The VFF variable $\lambda(k)$ is then defined as $\lambda(k) \triangleq 1 - \mathcal{E}(k)/\Sigma_0$, where $\Sigma_0 = \sigma_0^2 N_0$, σ_0^2 is the expected measurement noise variance, and N_0 determines the speed of adaptation, which corresponds to the nominal asymptotic memory length [4]. Note that $\lambda(k)$ is bounded by the rule

```
**if** $\lambda(k) < \lambda_{min}$, **then** $\lambda(k) = \lambda_{min}$,
**elseif** $1/\lambda(k)\mathrm{tr}(W(k)) > c$, **then** $\lambda(k) = 1$,
**endif**.
```

The covariance matrix is then updated as

$$
\mathcal{P}(k) = \frac{1}{\lambda(k)} W(k), \qquad (37)
$$

with $\mathcal{P}(0) = \delta I, c > \delta$, and the update for $\hat{\theta}(k)$ is (30).

4) Kalman Filter Update: The Kalman Filter update, where $A = I$, is of the form

$$
\hat{\theta}(k) = \hat{\theta}(k-1) + w_v(k), \qquad (38)
$$

where $w_v(k)$ is a white sequence with covariance $Q_{\mathrm{KF}}(k)$. The Kalman Filter can then be implemented by computing the Kalman gain as

$$
K(k) = \mathcal{P}(k) X(k)^{\mathrm{T}} \Gamma(k)^{-1}. \qquad (39)
$$

Note that the structure of $K(k)$ is the same as (34), and the matrix $\bar{R}^{-1}$ can be viewed as the measurement noise covariance. The next step is to update the estimate and compute the error covariance as

$$\hat{\theta}(k) = \hat{\theta}(k|k-1) + K(k)[-\bar{z}(k) - X(k)\hat{\theta}(k|k-1)],$$
$$\mathcal{P}(k) = [I - K(k)X(k)]\mathcal{P}(k|k-1).$$

Finally, we project ahead as

$$\hat{\theta}(k+1|k) = \hat{\theta}(k), \tag{40}$$
$$\mathcal{P}(k+1|k) = \mathcal{P}(k) + Q_{\text{KF}}(k). \tag{41}$$

V. Adaptive Autopilot Architecture

RCAC generates a pitch-rate command by adapting on the normal acceleration error and by using the normal acceleration command in the controller regressor Φ. As shown in Fig. 1, this pitch-rate command is then used as the reference to a fixed-gain proportional-integral (PI) controller. The PI loop is a pitch-rate stabilizing loop based on a single trim condition. This differs from [3], where the adaptive control law acts directly on the normal acceleration error to generate fin deflection commands based on that error. The steady-state pitch-rate command for the missile flying in a circular arc with a constant velocity is $Q_{\text{CMD,SS}} = g(\sin\theta - n_{z,\text{CMD}})/V_{\text{T}}$, where V_{T} is the tangential velocity. Since thrust is not regulated (assuming post burn-out), maintaining a constant velocity is not feasible. Other than this maneuver, deriving the equations for a pitch-rate command involves information that is not directly measured, thus motivating the need for an adaptive pitch-rate command.

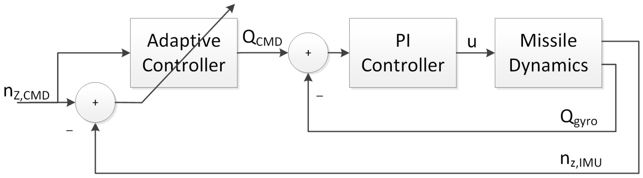

Fig. 1: Inner-Loop/Outer-Loop Adaptive Autopilot Architecture

To apply RCAC. the missile dynamics are linearized about a single trim point, and these dynamics are augmented by the PI controller. The resulting matricies A, B, and C from (10) and (11) are used to create the filter G_{f} [11] in (17). D_1 and D_2 are not used since external disturbances are not considered. Note that the linearization is used only for constructing an FIR filter and goes into the design of filter numerator $\mathcal{N}$; all simulations are performed using the full nonlinear dynamics. For RCAC, the performance signal $z(k)$ in (12) is used in cost minimization (22), where $z(k)$ is the difference between the normal acceleration command $n_{\text{z,cmd}}$, which is represented by $w(k)$ as in (12), and the normal acceleration measurement $n_{\text{z,IMU}}$, while $n_{\text{z,cmd}}$ is used in the controller regressor, Φ. Finally, for notational convenience, the time step k, represents the actual sample time of the controller, $k\text{T}_s$.

VI. Example of RCAC with standard RLS

We present an example that illustrates the need for modifying the RLS update law. Consider the nonlinear missile model presented in Section II. The missile is initialized at $M_0 = 3$, $\theta_0 = 25$ deg, $Z_0 = 3$ km, $Q_0 = X_0 = \alpha_0 = 0$, and the target is initialized at $X_{\text{t}}(0) = 4$ km, $Z_{\text{t}}(0) = 2.8$ km, $M_{\text{t}}(0) = 0.3$, and $\theta_{\text{t}}(0) = 180$ deg. RCAC has the initial values of $n_{\text{c}} = 4$, $n_{\text{f}} = 1$, $N_1 = -0.6$, $R_z = 1$, $R_u = 3$, and $\mathcal{P}(0) = 1e^{10}I_{2n_{\text{c}}}$. In order to intercept the target, the missile is commanded to pull an aggressive maneuver and consequently high angles of attack. This maneuver ensures that the missile's nonlinearities are exposed. Fig. 2 compares the 3LA to the pitch-rate commanded RCAC architecture. As shown in the figure, the adaptive controller is initially able to track the acceleration command but, as the flight progresses, tracking becomes worse. This inability to track is depicted by the controller gains and covariance in Fig.2. After 0.6 sec, the controller gains converge and the eigenvalues in the covariance matrix $\mathcal{P}$ approach zero.

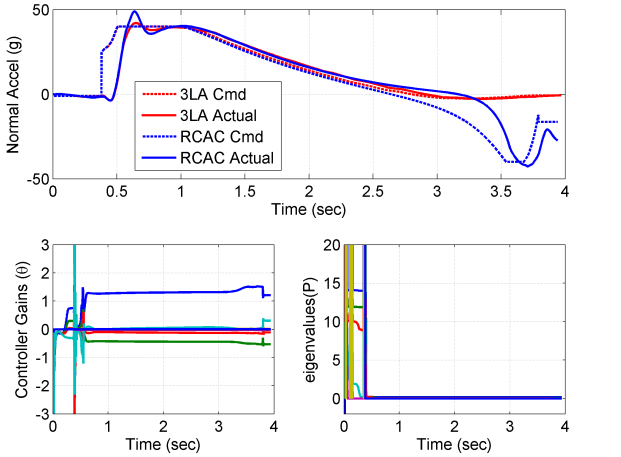

Fig. 2: Missile 'g' command tracking of the gain-scheduled three-loop autopilot vs. RCAC without a forgetting factor. [Top] compares the gain-scheduled 'g' command/response versus the RCAC 'g' command/response, where the tracking error increases significantly toward the end of flight, [Bottom] shows that the RCAC gains remain almost constant after about 0.6 sec due to the eigenvalues of the covariance matrix tending toward zero.

VII. COMPARISON OF 3LA, RCAC/CFF, RCAC/VFF

We revisit the example in Section VI and compare RCAC/CFF and RCAC/VFF with the three-loop autopilot (3LA). In this example, we vary the constant forgetting factor from 0.975 to 1.0 and compare the calculated miss distance (MD), which, after the target has been acquired, is the distance between the missile and the target at the instant the seeker loses sight of the target. The variable forgetting factor parameters are set to $\Sigma_0 = 1000$, $\lambda_{\text{min}} = 0.98$, and $c = 10\delta$. Fig. 3 shows the miss distance compared to a range of constant forgetting factors. Included in the plot is both the miss distance with the 3LA and the miss distance with RCAC/VFF. As the CFF is increased past 0.995, the miss distance becomes increasingly large (above the 2 meter plot limit). Note that the RCAC/VFF as well as some RCAC/CFF miss distances are below that of the 3LA.

To compare the CFF with VFF, note that with $\lambda = 0.979$, the CFF miss distance is the lowest of all the FF. The bottom of Fig. 3 depicts the adaptive gains through the missile flight. Notice how the gains are more aggressive and oscillatory during the latter half of the flight time for the CFF compared to VFF. This type of behavior is undesirable since it may cause the adaptive controller to become unstable leading to erroneous pitch-rate commands. The top right of Fig. 3 shows how the variable forgetting factor λ varies throughout the flight. This aids in producing adaptation that is smoother then the constant forgetting factor. From this we conclude that the VFF is the more desirable algorithm in comparison to the CFF for this application.

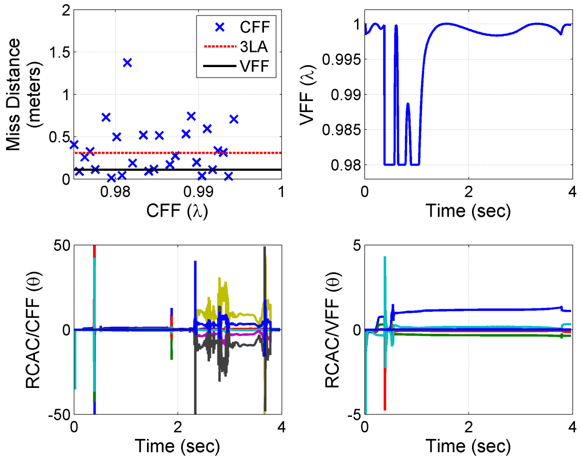

Fig. 3: Comparison of 3LA, RCAC/CFF, and RCAC/VFF. [Top Left] shows the calculated miss distance for a CFF that is varied from 0.975 to 1.0. The additional horizontal lines represent the miss distance of 3LA and RCAC/VFF with $\Sigma_0 = 1000$, $\lambda_{\min} = 0.98$, and $c = 10\delta$. [Bottom] shows the gains for RCAC/VFF and RCAC/CFF with CFF set to 0.9795, which has the best miss distance performance. [Top Right] shows how the VFF $\lambda(k)$ varies throughout the flight.

VIII. Comparison of RCAC/VFF and RCAC/KF

We revisit the example in Section VI to compare RCAC/VFF with RCAC/KF. The VFF tunings are the same as in Section VII, and the KF tuning is $Q_{\mathrm{KF}}(k) = 0.06$. Fig. 4 compares these algorithms. The calculated miss distance (MD) is similar, but unlike the 'g' response, the controller gains are significantly different as shown in the second row of the figures. Notice that the magnitude of the VFF controller gains are smaller than the magnitude of the KF gains and also vary less during flight.The bottom of Fig. 4 compares the eigenvalues of the covariance matrix. At around 0.4 sec, covariance values for both algorithms suddenly drop to zero. After this time, the trends look similar but the KF has a larger number of gains that are increasing.

IX. Robustness Comparison of 3LA, RCAC/VFF, and RCAC/KF

In this section, we present a final comparison between 3LA, RCAC/VFF, and RCAC/KF by considering four examples to test the robustness of the adaptive algorithms. The RCAC parameter G_{f} is modified to optimize performance through a range of flight Mach numbers. To do this, N_1 is a function of Mach number where $N_1(M) = -0.02M^2 + 0.25M - 0.87$. This modification is a gain-schedule in missile Mach number for increased performance. Unlike the gain-scheduled 3LA, we do not schedule based on angle of attack.

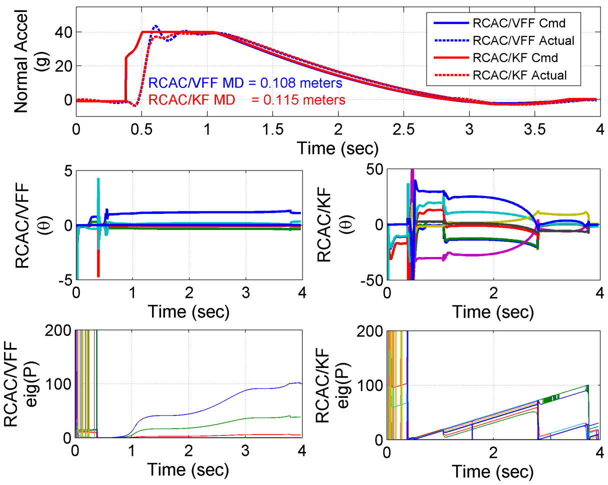

Fig. 4: Comparison of RCAC/VFF and RCAC/KF. Results shown use the parameters presented in the previous section with the addition of the KF variable $Q_{\mathrm{FK}}(k) = 0.06$. [Top] shows the similarity of the command and response of the two algorithms. The calculated miss distances are also similar, unlike the controller gains shown [Middle]. The magnitudes of the gains in RCAC/KF are much larger and vary more through the flight. [Bottom] shows how both eigenvalues of the covariance matrix drop to zero at time 0.4 sec and steadily increase as the flight progresses to optimize gains.

Example 9.1: In this example, the initial missile Mach number is increased to $M_0 = 4$, and the initial rotation angle is decreased to $\theta_0 = 15$ deg with the remaining initial conditions left as before. The top of Fig. 5 shows the normal acceleration error of each of the three controllers along with the miss distances. As shown, RCAC/VFF has the least miss distance but also has the largest amount of overshoot error around 0.5 sec when the maximum 'g' command is given. The overshoot is attributed to this method having the best miss distance value. The remaining two methods have similar responses with the gain-scheduled 3LA having a better miss distance value compared to RCAC/KF.

Example 9.2: In this example, the initial missile Mach number is $M_0 = 3$, $\theta_0 = 15$ deg, and the aerodynamic coefficient C_{z_α} in the body Z direction due to the angle-of-attack is scaled by 3, that is $C_{z_{\alpha_n}} = 3C_{z_\alpha}$ with the remaining initial conditions left as before. The middile of Fig. 5 shows the normal acceleration error of each of the three controllers as well as their calculated miss distances. In this example, RCAC/KF has the lowest miss distance and provides the best tracking despite this aerodynamic modification. RCAC/VFF is able to adapt to the modified aerodynamic coefficient but yields the worst miss distance. 3LA struggles with the aerodynamic coefficient modification and response similar to that of a lightly damped system.

Example 9.3: In this example, the initial missile Mach number is $M_0 = 2.5$, $\theta_0 = 25$ deg, and the aerodynamic moment coefficient due to the deflection of the control surface C_{M_δ} is scaled by 3, that is, $C_{M_{\delta_n}} = 3C_{M_\delta}$ with the remaining initial conditions left as before. The bottom of Fig. 5 shows the normal acceleration error of each controller as well as their calculated miss distances.

In this example, RCAC/KF has the lowest miss distance followed by RCAC/VFF. Both of these controllers quickly adapted to the modification in the aerodynamic coefficient and are able to track the normal acceleration command. 3LA struggled with this aerodynamic modification and continually oscillated throughout the entire flight.

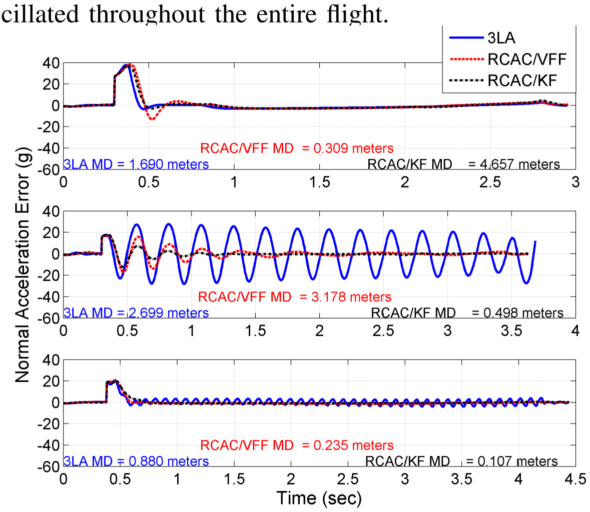

Fig. 5: Comparison of the normal acceleration error between 3LA, RCAC/VFF, and RCAC/KF on three representative cases. [Top] shows the case where the initial missile Mach number is increased to 4 and θ_0 decreased to 15 deg. [Middle] shows the case where the initial missile Mach number of 3, $\theta_0 = 15$ deg, and the aerodynamic coefficent $C_{z_{\alpha_n}} = 3C_{z_\alpha}$. [Bottom] shows the case where the initial missile Mach number is 2.5, $\theta_0 = 25$ deg, and the aerodynamic coefficent $C_{M_{\delta_n}} = 3C_{M_\delta}$.

Example 9.4: In this example, we revisit the scenario presented in Section VI with an unmodeled decrease in the actuator bandwidth from 150 Hz to 40 Hz throughout the flight. In doing this, the optimized design of 3LA will suffer due to the increase in the missiles' time constant. Fig. 6 shows the normal acceleration error of each of the three controllers as well as the actuator position and rate, respectively. 3LA struggles with the decrease in actuator performance whereas both the RCAC/VFF and the RCAC/KF are able to dampen out the response and adapt to the lower bandwidth actuator.

X. CONCLUSIONS

We extended [3, 5] in the RCAC formulation by including a variable forgetting factor and Kalman Filter. This extension allowed for continual parameter estimation in the adaptive controller update, a feature that is essential to controlling a system with nonlinear dynamics. Additionally, an inner-loop/outer-loop control architecture is used to adaptively adjust the pitch-rate command based on the normal acceleration command. The adaptive pitch-rate command loop is appropriate due to the inability to infer such a command when given an arbitrary normal acceleration command. Results show that the CFF, although effective, leads to aggressive adaptation that may lead to instabilities. We also showed that both RCAC/VFF and RCAC/KF allow for gain adapting through the entire flight and were able to track the normal acceleration command as well as the 3LA. The adaptive controller excelled when an aerodynamic coefficient modification was introduced to the system as well as having a lower bandwidth actuator.

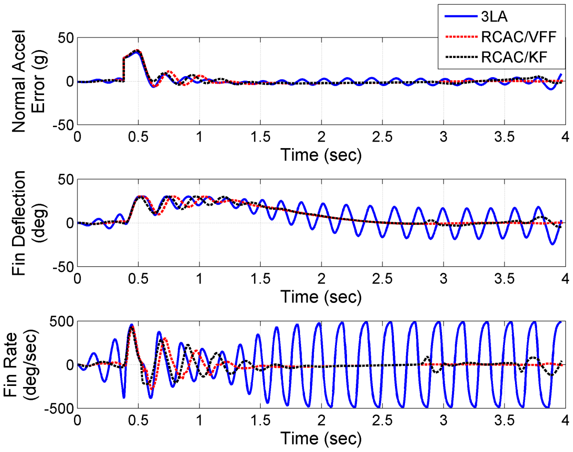

Fig. 6: Comparison of 3LA, RCAC/VFF, and RCAC/KF with an unmodeled decrease in actuator bandwidth. Nominally the actuator has a bandwidth of 150 Hz, but the results above assume an actuator with a bandwidth of 40 Hz. [Top] The normal acceleration error is shown, while the actuator deflection angle [Middile] and rate [Bottom] are shown.

XI. ACKNOWLEDGMENTS

We thank R. Fuentes and M. Unger for helpful discussions.

REFERENCES

[1] H. Buschek. Full envelope missile autopilot design using gain scheduled robust control. *Journal of Guidance, Control, and Dynamics*, 22(1):115–122, 1999.

[2] A.O. Cordero and D.Q. Mayne. Deterministic convergence of a self-tuning regulator with variable forgetting factor. *Control Theory and Applications, IEE Proceedings D*, 128(1):19–23, January 1981.

[3] A. M. D'Amato and D. S. Bernstein. "Adaptive Control of a Seeker-Guided 2D Missile with Unmodeled Aerodynamics," AIAA Guid. Nav. Contr. Conf., Minneapolis, MN,. August 2012, AIAA-2012-4615-516.

[4] T.R. Fortescue, L.S. Kershenbaum, and B.E. Ydstie. Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17(6):831 – 835, 1981.

[5] R.J. Fuentes, J. B. Hoagg, B. J. Anderton, A. M. D'Amato, and D. S. Bernstein. "Investigation of Cumulative Retrospective Cost Adaptive Control for Missile Application," AIAA Guid. Nav. Contr. Conf., Toronto,. August 2010, AIAA 2010–7577.

[6] J. B. Hoagg and D. S. Bernstein. Retrospective cost model reference adaptive control for nonminimum-phase systems. *Journal of Guidance, Control, and Dynamics*, 35(6):1767–1786, Oct 2012.

[7] J. B. Hoagg, A. M. Santillo, and D. S. Bernstein. Discrete-time adaptive command following and disturbance rejection with unknown exogenous dynamics. *Automatic Control, IEEE Transactions on*, 53(4):912 –928, May 2008.

[8] C. P. Mracek and J. R. Cloutier. "Full Envelope Missile Longitudinal Autopilot Design Using the State-Dependent Riccati Equation Method". In *AIAA GNC*, pages 1697–1705, 1997.

[9] C. P. Mracek and D. B. Ridgely. "Missile Longitudinal Autopilots: Connections Between Optimal Control and Classical Topologies," AIAA GNC Conf., San Francisco, CA. August 2005, AIAA 2005-6381.

[10] C. Schumacher and P. P. Khargonekar. Missile Autopilot Designs Using H_∞ Control with Gain Scheduling and Dynamic Inversion. *Journal of guidance, control, and dynamics*, 21(2):234–243, 1998.

[11] E.D. Sumer, M.H. Holzel, A.M. D'Amato, and D.S. Bernstein. "FIR-Based Phase Matching for Robust Retrospective-Cost Adaptive Control," Proc. ACC. pp 2707-2712, Montreal, Canada, June 2012.

[12] R. Venugopal and D. S. Bernstein. Adaptive disturbance rejection using ARMARKOV system representations. In *IEEE Trans. Contr. Sys. Tech.*, pages 257 –269 Vol. 8, Dec 2000.

[13] M. Xin and S. N. Balakrishnan. Missile autopilot design using a new suboptimal nonlinear control method. In *IEEE Proc. of 41st Aerospace Sciences Meeting and Exhibit, Reno*, pages 577–584, 2003.