



Prediction-Based Target Tracking Using Adaptive Input and State Estimation for Real-Time Numerical Differentiation

Shashank Verma*, Dennis S. Bernstein†
University of Michigan, Ann Arbor, Michigan, 48109

This paper presents a novel target tracking algorithm that uses numerical differentiation to estimate the velocity and acceleration of maneuvering targets. By applying adaptive input and state estimation, the first and second derivatives of noisy, sampled position data are computed with minimal latency. The estimated derivatives are then used to predict the target's position. The predicted motion of the target can be used to distinguish between ballistic and maneuvering targets, as well as to facilitate target interception. The performance of the proposed method is validated through detailed, comparative simulations, demonstrating significant improvements in tracking prediction and efficiency compared to existing methodologies.

I. Introduction

Target tracking is a pivotal aspect of guidance, navigation, and control applications, necessitating robust algorithms to handle target dynamics and environmental conditions. Control systems often rely on accurate tracking of targets to make informed decisions and adjustments. Numerous methodologies have been developed, leveraging different mathematical models and estimation techniques to enhance the accuracy and efficiency of target tracking systems.

Prior research has explored various approaches for improving target tracking. [1] employs target acceleration predictions produced by a recurrent neural network for a predictive pursuer guidance algorithm. [2] discusses straight-line target tracking for unmanned surface vehicles using constant-bearing guidance to calculate a desired velocity. [3] presents a tracking scheme based on the Kalman filter, estimating acceleration inputs from residuals and updating the filter accordingly. [4] enhances real-time performance using vision-based state estimators for target tracking.

The extended Kalman filter (EKF) has been used to estimate the kinematic state of reentry ballistic targets, as shown by [5], while [6] uses the Kalman filter to track the movements of ballistic vehicles. Adaptive input estimation techniques for estimating the acceleration of maneuvering targets are demonstrated by [7–9]. [10] compares the performance and accuracy of EKF, unscented Kalman filter (UKF), and particle filter (PF) for tracking ballistic targets and predicting impact points using measurements from 3D radar.

Iterative solutions employing state transition matrices to correct the initial conditions of ballistic vehicles for trajectory calculation have been presented by [11]. To overcome the limitations of conventional constant-level maneuver models, [12, 13] proposes a target-tracking technique using input estimation. [14] introduces a Kalman-filter-based tracking scheme incorporating input estimation for maneuvering targets.

Advanced filter motion models capable of reproducing a wide variety of target motions and tested using EKF, UKF, and PF are developed by [15]. [16] proposes an intelligent Kalman filter for tracking maneuvering targets, using a fuzzy system optimized by genetic algorithms and DNA coding methods to compute time-varying process noise. Machine learning techniques, such as expert prediction, have also been applied to collaborative sensor network target tracking methods by [17].

The performance characterization of α - β - γ filters constant-acceleration targets is discussed in [18], whereas [19] presents a procedure for selecting tracking parameters for all orders of tracking models. [20] provides the analytical expression for the tracking index, which is useful for real time applications. [21] presents an optimal reduced state estimator for the consistent tracking of maneuvering targets, addressing the limitations found in Kalman filters and interacting multiple model (IMM) estimators commonly used for this purpose. The accuracy of these methods depends on the α - β - γ parameter values. To facilitate the selection of these parameters, [22] proposes an adaptive α - β filter, where the α - β parameters are adjusted in real time using genetic algorithms.

*PhD Candidate, University of Michigan, Ann Arbor, shaaero@umich.edu.

†Professor, University of Michigan, Ann Arbor, dsbaero@umich.edu.

Incorporating real-time target orientation data acquired by imaging sensors into EKF for target tracking was demonstrated in [23]. [24] employs EKF to estimate the states of a moving target using position data from unmanned aerial vehicles, using these estimates to predict target trajectories.

Building on the “current model” concept for maneuvering targets, [25] introduces a modified Rayleigh density function to describe the probability density of target acceleration and presents an adaptive Kalman-filter algorithm to estimate the mean and variance of the maneuvering acceleration.

The present paper proposes a target tracking algorithm based on adaptive input and state estimation (AISE) for estimating the velocity and acceleration of the target using sampled position data. This approach minimizes the need for prior information about the spectrum of the position signal and its measurement noise. Additionally, the proposed approach avoids any assumptions of target maneuvers and target models. We then use the estimated velocity and acceleration to predict the trajectory of the target. The performance of the proposed method is evaluated through detailed comparative simulations.

The contents of this paper are as follows. Section II presents the problem statement. Section III summarizes the adaptive input and state estimation algorithm. Section IV provides a numerical example illustrating the accuracy of the algorithm for a simulated trajectory. Finally, Section V offers concluding remarks and directions for future research.

II. Problem Statement

We assume the Earth is inertially nonrotating and nonaccelerating. The right-handed frame $F_E = [\hat{i}_E \ \hat{j}_E \ \hat{k}_E]$ is fixed to the Earth, and the origin o_E of F_E is any convenient point fixed on the Earth; hence, o_E has zero inertial acceleration. \hat{k}_E points downward, and \hat{i}_E and \hat{j}_E are horizontal. o_T is any point fixed on the target vehicle.

The location of the target vehicle center of mass o_T relative to o_E at each time instant is given by the physical position vector \vec{r}_{o_T/o_E} as shown in Figure 1. We assume a ground radar sensor is measuring the position of the target vehicle in the frame F_E as

$$\begin{bmatrix} r_{x,k} \\ r_{y,k} \\ r_{z,k} \end{bmatrix} \triangleq \vec{r}_{o_T/o_E}(t_k) \Big|_E, \quad (1)$$

where k is the time step and $t_k \triangleq kT_s$. Here, $r_{q,k} = r_q(kT_s)$ is the position measurements of the target at time step k , with T_s being the sample time and q representing either x , y , or z . The first and second derivatives of \vec{r}_{o_T/o_E} with

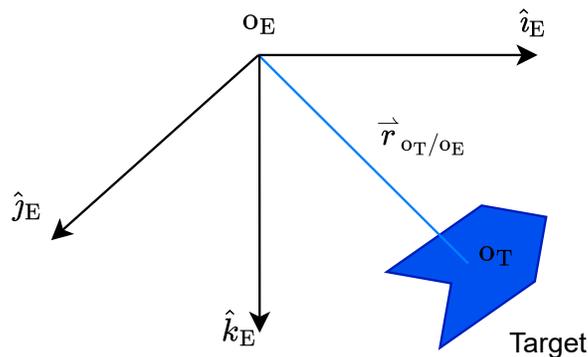


Fig. 1 \vec{r}_{o_T/o_E} is the physical position vector between o_T and o_E .

respect to F_E represent the physical velocity and acceleration vector $\overset{E\bullet}{\vec{r}}_{o_T/o_E}$ and $\overset{E\bullet\bullet}{\vec{r}}_{o_T/o_E}$, respectively. Single and double

differentiating (1) to resolve $\overset{E\bullet}{\vec{r}}_{OT/OE}$ and $\overset{E\bullet\bullet}{\vec{r}}_{OT/OE}$ in frame F_E as

$$\begin{bmatrix} \dot{r}_{x,k} \\ \dot{r}_{y,k} \\ \dot{r}_{z,k} \end{bmatrix} \triangleq \overset{E\bullet}{\vec{r}}_{OT/OE} (t_k) \Big|_E, \quad \begin{bmatrix} \ddot{r}_{x,k} \\ \ddot{r}_{y,k} \\ \ddot{r}_{z,k} \end{bmatrix} \triangleq \overset{E\bullet\bullet}{\vec{r}}_{OT/OE} (t_k) \Big|_E. \quad (2)$$

Using the position measurements of the target vehicle given by (1), along with the velocity and acceleration in (2), we predict the trajectories $\hat{r}_{q,k+l}$ of the target vehicle l steps into the future. In particular, at each step k , we use a second-order approximation to define the predicted position

$$\hat{r}_{q,k+l} \triangleq r_{q,k} + lT_s \hat{r}_{q,k} + \frac{1}{2} l^2 T_s^2 \hat{r}_{q,k}, \quad (3)$$

where q represents x, y , or z , $\hat{r}_{q,k}$ and $\hat{r}_{q,k}$ are estimates of $\dot{r}_{q,k}$ and $\ddot{r}_{q,k}$, respectively, $k+l$ is the future time step, and $l \in \{1, \dots, l_{\max}\}$, where l_{\max} is the prediction horizon. Note that (3) assumes that the velocity and acceleration are constant over the prediction horizon.

We use the position measurements and AISE to obtain the estimates $\hat{r}_{q,k}$ and $\hat{r}_{q,k}$. Table 1 shows the data from the ground radar along with the derivatives that are obtained to compute $\hat{r}_{q,k+l}$ in (3).

Sensors	Data	Processed Data
Ground radar	$r_{q,k}$	$\hat{r}_{q,k}, \hat{r}_{q,k}$

Table 1 Ground radar is used to obtain position data $r_{q,k}$. AISE uses the position data to compute the estimates $\hat{r}_{q,k}$ and $\hat{r}_{q,k}$. Here, q represents x, y , or z .

III. Adaptive Input and State Estimation

We summarize AISE [26–28] for real-time numerical differentiation of position data of the target to estimate its velocity and acceleration.

Consider the linear discrete-time SISO system

$$x_{k+1} = Ax_k + Bd_k, \quad (4)$$

$$y_k = Cx_k + D_{2,k}v_k, \quad (5)$$

where $k \geq 0$ is the step, $x_k \in \mathbb{R}^n$ is the unknown state, $d_k \in \mathbb{R}$ is unknown input, $y_k \in \mathbb{R}$ is a measured output, $v_k \in \mathbb{R}$ is standard white noise, and $D_{2,k}v_k \in \mathbb{R}$ is the sensor noise at time $t = kT_s$, where T_s is the sample time. The matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times 1}$, and $C \in \mathbb{R}^{1 \times n}$, are assumed to be known and $D_{2,k}$ is assumed to be unknown. The sensor-noise covariance is $V_{2,k} \triangleq D_{2,k}D_{2,k}^T$. The goal of adaptive input estimation (AIE) is to estimate d_k and x_k .

In the application of AIE to real-time numerical differentiation, we use (4) and (5) to model a discrete-time integrator. As a result, AIE furnishes an estimate denoted by \hat{d}_k for the derivative of the sampled output y_k . For single discrete-time differentiation, the values are $A = 1$, $B = T_s$, and $C = 1$. However, in the case of double discrete-time differentiation,

$$A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (6)$$

A. Input Estimation

AIE comprises three subsystems, namely, the Kalman-filter forecast subsystem, the input-estimation subsystem, and the Kalman-filter data-assimilation subsystem. First, consider the Kalman-filter forecast step

$$x_{fc,k+1} = Ax_{da,k} + B\hat{d}_k, \quad (7)$$

$$y_{fc,k} = Cx_{fc,k}, \quad (8)$$

$$z_k = y_{fc,k} - y_k, \quad (9)$$

where $x_{\text{da},k} \in \mathbb{R}^n$ is the data-assimilation state, $x_{\text{fc},k} \in \mathbb{R}^n$ is the forecast state, \hat{d}_k is the estimate of d_k , $y_{\text{fc},k} \in \mathbb{R}$ is the forecast output, $z_k \in \mathbb{R}$ is the residual, and $x_{\text{fc},0} = 0$.

Next, in order to obtain \hat{d}_k , the input-estimation subsystem of order n_e is given by the exactly proper dynamics

$$\hat{d}_k = \sum_{i=1}^{n_e} P_{i,k} \hat{d}_{k-i} + \sum_{i=0}^{n_e} Q_{i,k} z_{k-i}, \quad (10)$$

where $P_{i,k} \in \mathbb{R}$ and $Q_{i,k} \in \mathbb{R}$. AIE minimizes z_k by updating $P_{i,k}$ and $Q_{i,k}$ as shown below. The subsystem (10) can be reformulated as

$$\hat{d}_k = \Phi_k \theta_k, \quad (11)$$

where the estimated coefficient vector $\theta_k \in \mathbb{R}^{l_\theta}$ is defined by

$$\theta_k \triangleq \begin{bmatrix} P_{1,k} & \cdots & P_{n_e,k} & Q_{0,k} & \cdots & Q_{n_e,k} \end{bmatrix}^T, \quad (12)$$

the regressor matrix $\Phi_k \in \mathbb{R}^{1 \times l_\theta}$ is defined by

$$\Phi_k \triangleq \begin{bmatrix} \hat{d}_{k-1} & \cdots & \hat{d}_{k-n_e} & z_k & \cdots & z_{k-n_e} \end{bmatrix}, \quad (13)$$

and $l_\theta \triangleq 2n_e + 1$. The subsystem (10) can be written using backward shift operator \mathbf{q}^{-1} as

$$\hat{d}_k = G_{\hat{d},k}(\mathbf{q}^{-1})z_k, \quad (14)$$

where

$$G_{\hat{d},k} \triangleq D_{\hat{d},k}^{-1} N_{\hat{d},k}, \quad (15)$$

$$D_{\hat{d},k}(\mathbf{q}^{-1}) \triangleq I_{l_d} - P_{1,k} \mathbf{q}^{-1} - \cdots - P_{n_e,k} \mathbf{q}^{-n_e}, \quad (16)$$

$$N_{\hat{d},k}(\mathbf{q}^{-1}) \triangleq Q_{0,k} + Q_{1,k} \mathbf{q}^{-1} + \cdots + Q_{n_e,k} \mathbf{q}^{-n_e}. \quad (17)$$

Next, define the filtered signals

$$\Phi_{f,k} \triangleq G_{f,k}(\mathbf{q}^{-1})\Phi_k, \quad \hat{d}_{f,k} \triangleq G_{f,k}(\mathbf{q}^{-1})\hat{d}_k, \quad (18)$$

where, for all $k \geq 0$,

$$G_{f,k}(\mathbf{q}^{-1}) = \sum_{i=1}^{n_f} \mathbf{q}^{-i} H_{i,k}, \quad (19)$$

$$H_{i,k} \triangleq \begin{cases} CB, & k \geq i = 1, \\ C\bar{A}_{k-1} \cdots \bar{A}_{k-(i-1)} B, & k \geq i \geq 2, \\ 0, & i > k, \end{cases} \quad (20)$$

and $\bar{A}_k \triangleq A(I + K_{\text{da},k}C)$, where $K_{\text{da},k}$ is the Kalman-filter gain given by (32) below. Furthermore, for all $k \geq 0$, define the *retrospective variable* $z_{r,k} : \mathbb{R}^{l_\theta} \rightarrow \mathbb{R}$ by

$$z_{r,k}(\hat{\theta}) \triangleq z_k - (\hat{d}_{f,k} - \Phi_{f,k} \hat{\theta}), \quad (21)$$

and define the *retrospective cost function* $\mathcal{J}_k : \mathbb{R}^{l_\theta} \rightarrow \mathbb{R}$ by

$$\mathcal{J}_k(\hat{\theta}) \triangleq \left(\prod_{j=1}^k \lambda_j \right) (\hat{\theta} - \theta_0)^T R_\theta (\hat{\theta} - \theta_0) + \sum_{i=0}^k \left(\prod_{j=1}^{k-i} \lambda_j \right) [R_z z_{r,i}^2(\hat{\theta}) + R_d (\Phi_i \hat{\theta})^2]$$

where $R_\theta \in \mathbb{R}^{l_\theta \times l_\theta}$ is positive definite, $R_z \in (0, \infty)$, $R_d \in (0, \infty)$, and $\lambda_k \in (0, 1]$ is the forgetting factor. Then, for all $k \geq 0$, the unique global minimizer $\theta_{k+1} \triangleq \arg \min_{\hat{\theta} \in \mathbb{R}^{l_\theta}} \mathcal{J}_k(\hat{\theta})$ is given recursively by the RLS update equations [29] as

$$P_{k+1}^{-1} = \lambda_k P_k^{-1} + (1 - \lambda_k) R_\infty + \tilde{\Phi}_k^T \tilde{R} \tilde{\Phi}_k, \quad (22)$$

$$\theta_{k+1} = \theta_k - P_{k+1} \tilde{\Phi}_k^T \tilde{R} (\tilde{z}_k + \tilde{\Phi}_k \theta_k), \quad (23)$$

where $P_0 \triangleq R_\theta^{-1}$, for all $k \geq 0$, positive-definite $P_k \in \mathbb{R}^{l_\theta \times l_\theta}$ is the covariance matrix, the positive-definite matrix $R_\infty \in \mathbb{R}^{l_\theta \times l_\theta}$ is the user-selected *resetting matrix*, and where, for all $k \geq 0$,

$$\tilde{\Phi}_k \triangleq \begin{bmatrix} \Phi_{f,k} \\ \Phi_k \end{bmatrix}, \quad \tilde{z}_k \triangleq \begin{bmatrix} z_k - \hat{d}_{f,k} \\ 0 \end{bmatrix}, \quad \tilde{R} \triangleq \begin{bmatrix} R_z & 0 \\ 0 & R_d \end{bmatrix}.$$

Hence, (22) and (23) recursively update the input-estimation subsystem (10).

A forgetting factor $\lambda_k < 1$ in (22) enables the eigenvalues of P_k to decrease, facilitating ongoing adaptation of the input-estimation subsystem (10), even after extensive data collection [30]. Conversely, the resetting matrix R_∞ in (22) prevents the eigenvalues of P_k from becoming excessively large under conditions of poor excitation [31], a phenomenon known as covariance windup [32].

Next, variable-rate forgetting based on the F -test [33] is used for all $k \geq 0$ to select the forgetting factor $\lambda_k \in (0, 1]$. For all $k \geq 0$, we define the *residual error* at step k as

$$\varepsilon_k \triangleq \tilde{z}_k + \tilde{\Phi}_k \theta_k \in \mathbb{R}^2. \quad (24)$$

Note that the residual error is a metric of how well the input-estimation subsystem (10) predicts the input one step into the future. Furthermore, for all $k \geq 0$, the sample mean of the residual errors over the past $\tau \geq 1$ steps is defined as

$$\bar{\varepsilon}_{\tau,k} \triangleq \frac{1}{\tau} \sum_{i=k-\tau+1}^k \varepsilon_i \in \mathbb{R}^2, \quad (25)$$

and the sample variance of the residual errors over the past τ steps is defined as

$$\Sigma_{\tau,k} \triangleq \frac{1}{\tau} \sum_{i=k-\tau+1}^k (\varepsilon_i - \bar{\varepsilon}_{\tau,k})(\varepsilon_i - \bar{\varepsilon}_{\tau,k})^T \in \mathbb{R}^{2 \times 2}. \quad (26)$$

The approach in [33] compares $\Sigma_{\tau_n,k}$ to $\Sigma_{\tau_d,k}$, where $\tau_n \geq 1$ is the short-term sample size, and $\tau_d > \tau_n$ is the long-term sample size. If the short-term variance $\Sigma_{\tau_n,k}$ is found to be statistically more significant than the long-term variance $\Sigma_{\tau_d,k}$, according to the Lawley-Hotelling trace approximation [34], then $\lambda_k < 1$ is chosen, inversely proportional to its statistical significance. Otherwise, λ_k is set to 1. In particular, for all $k \geq 0$, the forgetting factor is selected as

$$\lambda_k \triangleq \frac{1}{1 + \eta g_k \mathbf{1}[g_k]}, \quad (27)$$

where $\eta \geq 0$ is a tuning parameter, $\mathbf{1}: \mathbb{R} \rightarrow \{0, 1\}$ is the unit step function, and, for all $k \geq 0$,

$$g_k \triangleq \sqrt{\frac{\tau_n}{\tau_d} \frac{\text{tr}(\Sigma_{\tau_n,k} \Sigma_{\tau_d,k}^{-1})}{c}} - \sqrt{F_{2\tau_n, b}^{-1}(1 - \alpha)}, \quad (28)$$

$$a \triangleq \frac{(\tau_n + \tau_d - 3)(\tau_d - 1)}{(\tau_d - 5)(\tau_d - 2)}, \quad (29)$$

$$b \triangleq 4 + \frac{2(\tau_n + 1)}{a - 1}, \quad c \triangleq \frac{2\tau_n(b - 2)}{b(\tau_d - 3)}, \quad (30)$$

and where $\alpha \in [0, 1]$ is the significance level and $F_{2\tau_n, b}^{-1}: [0, 1] \rightarrow \mathbb{R}$ is the inverse cumulative distribution function of the F -distribution with degrees of freedom $2\tau_n$ and b . For further details, see [33] and [34].

B. State Estimation

The forecast variable $x_{fc,k}$, given by (7), is used to obtain the estimate $x_{da,k}$ of x_k given, for all $k \geq 0$, by the Kalman-filter data-assimilation step

$$x_{da,k} = x_{fc,k} + K_{da,k} z_k, \quad (31)$$

where the Kalman-filter gain $K_{\text{da},k} \in \mathbb{R}^n$, the data-assimilation error covariance $P_{\text{da},k} \in \mathbb{R}^{n \times n}$, and the forecast error covariance $P_{\text{f},k+1} \in \mathbb{R}^{n \times n}$ are given by

$$K_{\text{da},k} = -P_{\text{f},k} C^T (C P_{\text{f},k} C^T + V_{2,k})^{-1}, \quad (32)$$

$$P_{\text{da},k} = (I_n + K_{\text{da},k} C) P_{\text{f},k}, \quad (33)$$

$$P_{\text{f},k+1} = A P_{\text{da},k} A^T + V_{1,k}, \quad (34)$$

where $V_{2,k} \in \mathbb{R}$ is the measurement covariance matrix and

$$V_{1,k} \triangleq B \text{var} (d_k - \hat{d}_k) B^T + A \text{cov} (x_k - x_{\text{da},k}, d_k - \hat{d}_k) B^T + B \text{cov} (d_k - \hat{d}_k, x_k - x_{\text{da},k}) A^T \quad (35)$$

and $P_{\text{f},0} = 0$.

C. Adaptive State Estimation

This section summarizes adaptive state estimation of AISE. Assuming that, for all $k \geq 0$, $V_{1,k}$ and $V_{2,k}$ are unknown in (34) and (32), the goal is to adapt $V_{1,\text{adapt},k}$ and $V_{2,\text{adapt},k}$ at each step k to estimate $V_{1,k}$ and $V_{2,k}$, respectively. To do this, we define, for all $k \geq 0$, the performance metric $J_k: \mathbb{R}^{n \times n} \times \mathbb{R} \rightarrow \mathbb{R}$ as

$$J_k(V_1, V_2) \triangleq |\widehat{S}_k - S_k|, \quad (36)$$

where \widehat{S}_k is the sample variance of z_k over $[0, k]$ given by

$$\widehat{S}_k \triangleq \frac{1}{k} \sum_{i=0}^k (z_i - \bar{z}_k)^2, \quad \bar{z}_k \triangleq \frac{1}{k+1} \sum_{i=0}^k z_i, \quad (37)$$

and S_k is the variance of the residual z_k given by the Kalman filter, defined as

$$S_k \triangleq C (A P_{\text{da},k-1} A^T + V_1) C^T + V_2. \quad (38)$$

For all $k \geq 0$, we assume that $V_{1,\text{adapt},k} \triangleq \eta_k I_n$ and we define $\eta_k \in \mathbb{R}$ and $V_{2,\text{adapt},k}$ as

$$\eta_k, V_{2,\text{adapt},k} \triangleq \arg \min_{\eta \in [\eta_L, \eta_U], V_2 \geq 0} J_k(\eta I_n, V_2), \quad (39)$$

where $0 \leq \eta_L \leq \eta_U$. Next, defining $J_{\text{f},k}: \mathbb{R} \rightarrow \mathbb{R}$ as

$$J_{\text{f},k}(V_1) \triangleq \widehat{S}_k - C (A P_{\text{da},k-1} A^T + V_1) C^T, \quad (40)$$

and using (38), (36) can be rewritten as

$$J_k(V_1, V_2) = |J_{\text{f},k}(V_1) - V_2|. \quad (41)$$

We construct a set of positive values of $J_{\text{f},k}$ as

$$\mathcal{J}_{\text{f},k} \triangleq \{J_{\text{f},k}(\eta I_n) : J_{\text{f},k}(\eta I_n) > 0, \eta_L \leq \eta \leq \eta_U\} \subseteq \mathbb{R}. \quad (42)$$

Finally, Proposition 1 gives a method to compute η_k and $V_{2,\text{adapt},k}$, defined in (39).

Proposition 1 *Let $k \geq 0$ and let $\eta_k \in [\eta_L, \eta_U]$ and $V_{2,k} \geq 0$ be given by (39). If $\mathcal{J}_{\text{f},k}$, defined in (42), is nonempty, then, for any $\beta \in [0, 1]$, η_k and $V_{2,k}$ are given by*

$$\eta_k = \arg \min_{\eta \in [\eta_L, \eta_U]} |J_{\text{f},k}(\eta I_n) - \widehat{J}_{\text{f},k}(\beta)|, \quad (43)$$

$$V_{2,\text{adapt},k} = J_{\text{f},k}(\eta_k I_n), \quad (44)$$

where

$$\widehat{J}_{\text{f},k}(\beta) \triangleq \beta \min \mathcal{J}_{\text{f},k} + (1 - \beta) \max \mathcal{J}_{\text{f},k}, \quad (45)$$

If $\mathcal{J}_{\text{f},k}$ is empty, then η_k and $V_{2,k}$ are given by

$$\eta_k = \arg \min_{\eta \in [\eta_L, \eta_U]} |J_{\text{f},k}(\eta I_n)|, \quad (46)$$

$$V_{2,\text{adapt},k} = 0. \quad (47)$$

Proof: See Section 5.2 of [26].

IV. Numerical Examples

In this section, a numerical example is provided to compare the accuracy of the predicted trajectory of the target vehicle (3) using velocity and acceleration estimates. For comparison, the estimates are obtained from two different numerical differentiation methods and using α - β - γ filter (ABG). The first differentiation method is the backward difference with Butterworth filter (BDB), and the second is AISE. In BDB, the noisy position measurements are first refined using a Butterworth filter, and then the backward difference is applied. We assume a planar x - y model in this section, hence ignoring the z component. The tracking index Γ is a well-known parameter in the α - β - γ filter [19]. Here, we also compare the performance of the α - β - γ filter for a range of values of Γ .

We further assess the value of acceleration estimates for trajectory prediction by presenting six versions of BDB, ABG, and AISE prediction using either only the velocity estimate or both the velocity and acceleration estimates in (3), as shown in Table 2.

Prediction Methods	$\hat{r}_{q,k}$	$\hat{i}_{q,k}$
BDB/v	Used	Not used
BDB/va	Used	Used
ABG/v	Used	Not used
ABG/va	Used	Used
AISE/v	Used	Not used
AISE/va	Used	Used

Table 2 Definitions of the prediction methods using BDB, ABG, and AISE. Each version includes either only the velocity estimate $\hat{r}_{q,k}$ or both the velocity estimate $\hat{r}_{q,k}$ and the acceleration estimate $\hat{i}_{q,k}$ in the trajectory prediction equation (3). Here q represents x or y .

To quantify the accuracy of the predicted trajectory, we define the error metric for the prediction horizon of l_{\max} steps

$$\text{RMSE}_{q,l_{\max}} \triangleq \frac{1}{N - l_{\max} - 499} \sqrt{\sum_{k=500}^{N-l_{\max}} (r_{q,k+l_{\max}} - \hat{r}_{q,k+l_{\max}})^2}, \quad (48)$$

where q represents x or y , and to avoid the transient phase for the adaptation of AISE, k starts from 500 in (48).

Example 1 *Trajectory Prediction for a Parabolic Trajectory.* In this simulation scenario, the target follows a parabolic trajectory in the planer model under constant gravity of 9.8 m/s^2 in the negative y direction. The discrete-time equations governing the trajectory are defined as

$$r_{x,k} = 100kT_s, \quad (49)$$

$$r_{y,k} = 200kT_s - 9.8 \frac{(kT_s)^2}{2}, \quad (50)$$

where $T_s = 0.01 \text{ s}$ and for all $k \geq 0$. To simulate the noisy measurement of the target position $r_{x,k}$ and $r_{y,k}$, white Gaussian noise is added to each position measurement, with standard deviation $\sigma = 1$. We perform Monte-Carlo simulations with 100 trials.

For single differentiation using AISE, we set $n_e = 25$, $n_f = 50$, $R_z = 1$, $R_d = 10^{-6.7}$, $R_\theta = 10^{-1} I_{51}$, $\eta = 0.008$, $\tau_n = 20$, $\tau_d = 160$, $\alpha = 0.0008$, and $R_\infty = 10^2$. The parameters $V_{1,k}$ and $V_{2,k}$ are adapted, with $\eta_L = 10^{-6}$ and $\eta_U = 1$ as described in Section III.C. For double differentiation using AISE, we set $n_e = 25$, $n_f = 20$, $R_z = 1$, $R_d = 10^{-4}$, $R_\theta = 10^{-2} I_{51}$, $\eta = 0.008$, $\tau_n = 20$, $\tau_d = 160$, $\alpha = 0.0008$, and $R_\infty = 10^1$. Similarly, $V_{1,k}$ and $V_{2,k}$ are adapted, with $\eta_L = 10^{-6}$ and $\eta_U = 10^{-2}$ in Section III.C. For BDB, the Butterworth filter is 10^{th} order with a cutoff frequency of $0.8\pi \text{ rad/step}$. For ABG, the tracking index $\Gamma = 0.6$.

Figure 2a compares the predicted trajectory of $l_{\max} = 100$ -steps prediction horizon using BDB/v, ABG/v, and AISE/v to the measured position of the target. Figure 2b, a zoomed-in view of Figure 2a, demonstrates that the predicted trajectory by AISE/v is closer to the measured trajectory than that of BDB/v and ABG/v. AISE/va in Figure 3c, outperforms BDB/va in Figure 3a and ABG/va in Figure 3b. Figure 4a compares AISE/v and AISE/va, showing that

the prediction by AISE/va is initially less accurate, as evidenced in the lower-left corner, due to the transient phase of AISE adaptation, particularly for double differentiation. Figure 4b, a zoomed-in view of Figure 4a, indicates that the performances of AISE/va and AISE/v are similar.

These observations are further verified by the $RMSE_{q,l_{\max}}$ error metric (48), as shown in Table 3. Table 3 presents RMSE values for 100, 200, and 300-step prediction horizons, comparing the true position with the predicted position. For $RMSE_{x,l_{\max}}$, AISE/v has the lowest values, whereas for $RMSE_{y,l_{\max}}$, AISE/va has the lowest values across all $l_{\max} = 100, 200, \text{ and } 300$. This discrepancy is due to the fact that the true acceleration $\ddot{r}_{x,k}$ is zero for the entire duration of the parabolic trajectory in (49), whereas the estimated acceleration $\hat{\ddot{r}}_{x,k}$ is not zero for all $k > 0$, resulting in marginally worse performance when the acceleration estimate is considered in AISE/va. Figure 5 shows the variation of $RMSE_{q,300}$ with respect to the tracking parameter $\Gamma \in [0, 1]$ for ABG/v and ABG/va prediction method, with $q = x$ and $q = y$.

Prediction Method	$RMSE_{x,100}$	$RMSE_{y,100}$	$RMSE_{x,200}$	$RMSE_{y,200}$	$RMSE_{x,300}$	$RMSE_{y,300}$
BDB/v	1.05	5.14	2.27	19.69	3.36	44.17
BDB/va	87.63	81.99	347.65	323.66	805.44	777.71
ABG/v	1.24	4.95	2.53	19.36	3.95	43.97
ABG/va	40.72	42.51	181.87	178.12	388.08	407.97
AISE/v	0.13	6.16	0.20	22.05	0.28	47.73
AISE/va	0.23	1.58	0.87	3.46	1.84	5.69

Table 3 RMSE values for 100, 200, and 300 steps prediction horizons. The value in bold represents the minimum RMSE value in each column.

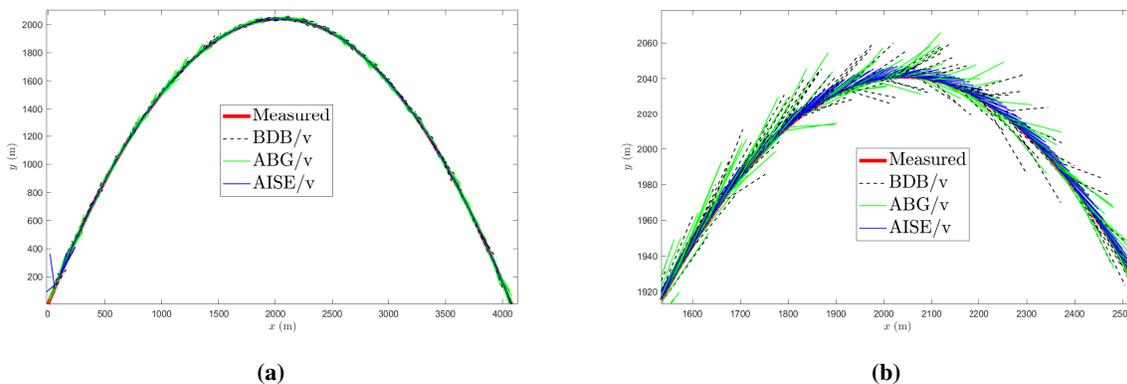
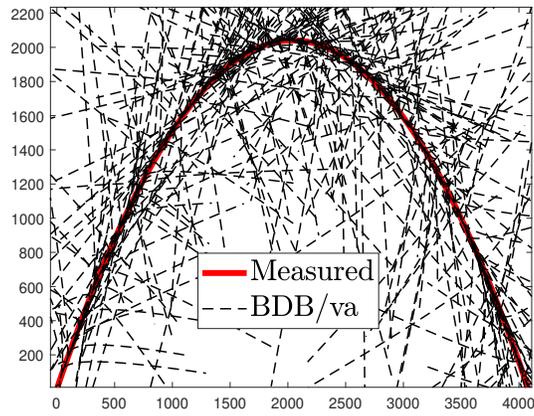


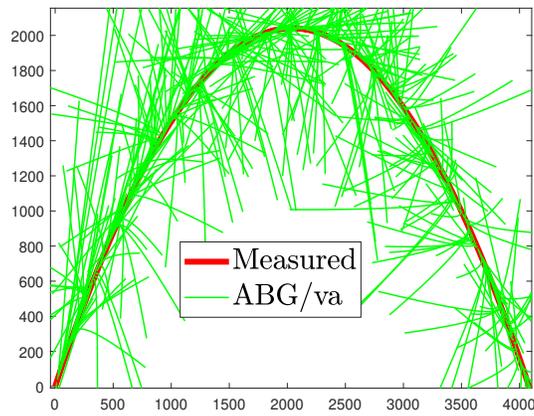
Fig. 2 Example 1: Trajectory prediction for a parabolic trajectory. (a) The dashed black line, green line, and blue line show the predicted trajectory using BDB/v, ABG/v, and AISE/v, respectively. (b) Zoomed view of (a). AISE/v provides a more accurate prediction than BDB/v and ABG/v. The prediction horizon $l_{\max} = 100$ steps.

V. Conclusions

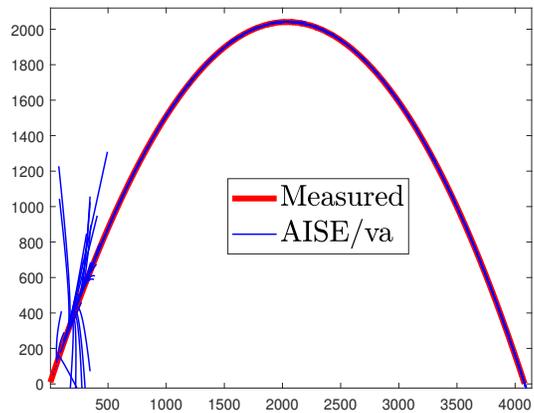
This paper investigated the performance of a novel target tracking algorithm that uses adaptive real-time numerical differentiation to estimate the velocity and acceleration of a maneuvering target. Adaptive input and state estimation (AISE) is used to estimate the first and second derivatives of noisy, sampled position data with minimal latency. The velocity and acceleration estimates are then used to predict the position of the target over an l -step prediction horizon. Future research will consider three-dimensional target-tracking scenarios such as ballistic trajectories with aerodynamic and central gravity effects, integrate bearing and range-rate data, estimate the curvature and torsion of the trajectory, and apply the methodology to maneuvering targets with the ultimate goal of discriminating between maneuvering and non-maneuvering targets.



(a)



(b)



(c)

Fig. 3 Example 1: Trajectory prediction for a parabolic trajectory. The dashed black line, green line, and blue line show the predicted trajectory using BDB/va, ABG/va, and AISE/va in (a), (b), and (c) subfigures, respectively. AISE/va provides a more accurate prediction than BDB/va. The prediction horizon $l_{\max} = 100$ steps.

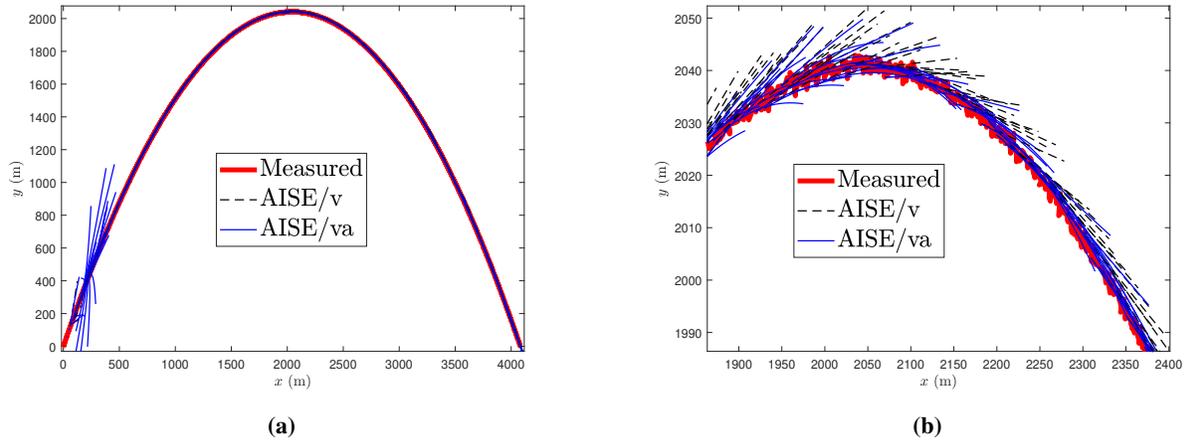
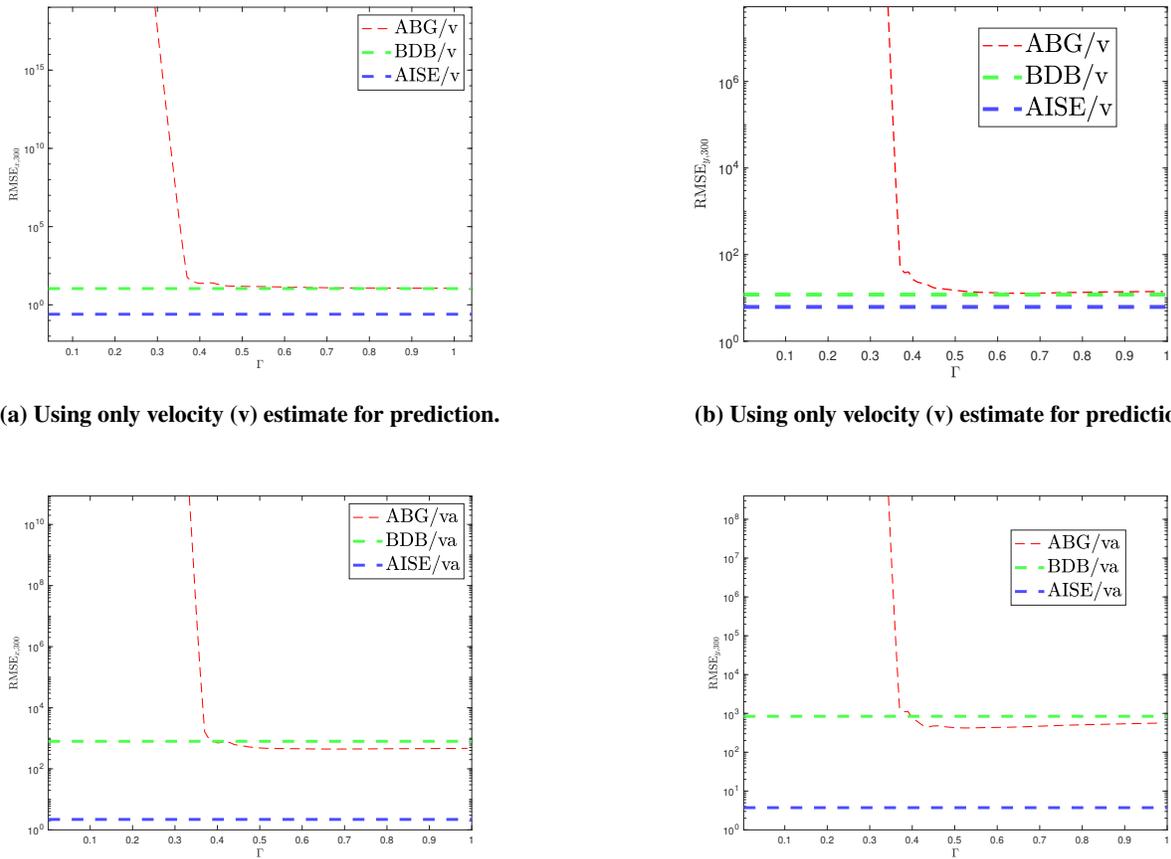


Fig. 4 Example 1: Trajectory prediction for a parabolic trajectory. (a) The dashed black line shows the predicted trajectory using AISE/v. The blue line shows the predicted trajectory using AISE/va. (b) Zoomed view of (a). AISE/va provides similar accuracy to AISE/v. The prediction horizon $l_{\max} = 100$ steps.



(c) Using both velocity (v) and acceleration (a) estimates for prediction.

(d) Using both velocity (v) and acceleration (a) estimates for prediction.

Fig. 5 Example 1: Trajectory prediction for a parabolic trajectory. $RMSE_{q,300}$ versus tracking index Γ for $q = x, y$.

Acknowledgments

This research was supported by NSF grant CMMI 2031333.

References

- [1] Akcal, M. U., and Chowdhary, G., "A Predictive Guidance Scheme for Pursuit-Evasion Engagements," *AIAA Scitech Forum*, 2021. AIAA 2021-1226.
- [2] Breivik, M., Hovstein, V. E., and Fossen, T. I., "Straight-Line Target Tracking for Unmanned Surface Vehicles," *Modeling, Identification and Control: A Norwegian Research Bulletin*, Vol. 29, No. 4, 2008, pp. 131–149.
- [3] Chan, Y., Hu, A., and Plant, J., "A Kalman Filter Based Tracking Scheme with Input Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-15, No. 2, 1979, pp. 237–244.
- [4] Chuang, H.-M., He, D., and Namiki, A., "Autonomous Target Tracking of UAV Using High-Speed Visual Feedback," *Applied Sciences*, Vol. 9, No. 21, 2019, p. 4552.
- [5] Ghosh, S., and Mukhopadhyay, S., "Tracking Reentry Ballistic Targets using Acceleration and Jerk Models," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 47, No. 1, 2011-01, pp. 666–683.
- [6] Guo, Y., Yao, Y., Wang, S., Yang, B., He, F., and Zhang, P., "Maneuver Control Strategies to Maximize Prediction Errors in Ballistic Middle Phase," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013-07, pp. 1225–1234.
- [7] Gupta, R., D'Amato, A., Ali, A., and Bernstein, D., "Retrospective-Cost-Based Adaptive State Estimation and Input Reconstruction for a Maneuvering Aircraft with Unknown Acceleration," *AIAA 2012-4600, AIAA Guidance, Navigation, and Control Conference*, 2012.
- [8] Ansari, A., and Bernstein, D. S., "Input Estimation for Nonminimum-Phase Systems With Application to Acceleration Estimation for a Maneuvering Vehicle," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 4, 2019, pp. 1596–1607.
- [9] Han, L., Ren, Z., and Bernstein, D. S., "Maneuvering Target Tracking Using Retrospective-Cost Input Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 52, No. 5, 2016, pp. 2495–2503.
- [10] Hardiman, D. F., Kerce, J. C., and Brown, G. C., "Nonlinear Estimation Techniques for Impact Point Prediction of Ballistic Targets," *Proc. SPIE 6236, Signal and Data Processing of Small Targets*, 2006, p. 62360C.
- [11] Harlin, W., and Cicci, D., "Ballistic Missile Trajectory Prediction Using a State Transition Matrix," *Applied Mathematics and Computation*, Vol. 188, No. 2, 2007-05, pp. 1832–1847.
- [12] Hungu Lee, and Min-Jea Tahk, "Generalized Input-Estimation Technique for Tracking Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 35, No. 4, 1999, pp. 1388–1402.
- [13] Bar-Shalom, Y., Chang, K., and Blom, H., "Tracking a Maneuvering Target Using Input Estimation Versus the Interacting Multiple Model Algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 25, No. 2, 1989, pp. 296–300.
- [14] Khaloozadeh, H., and Karsaz, A., "Modified Input Estimation Technique for Tracking Manoeuvring Targets," *IET Radar, Sonar & Navigation*, Vol. 3, No. 1, 2009, p. 30.
- [15] Kim, J., Menon, P., and Ohlmeyer, E., "Motion Models for use with the Maneuvering Ballistic Missile Tracking Estimators," *AIAA Guidance, Navigation, and Control Conference*, 2010.
- [16] Lee, B., Park, J., Joo, Y., and Jin, S., "Intelligent Kalman filter for tracking a manoeuvring target," *IEE Proceedings - Radar, Sonar and Navigation*, Vol. 151, No. 6, 2004, p. 344.
- [17] Li, Y., Li, X., and Wang, H., "Target Tracking in a Collaborative Sensor Network," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 4, 2014, pp. 2694–2714.
- [18] Tenne, D., and Singh, T., "Characterizing Performance of α - β - γ Filters," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 3, 2002, pp. 1072–1087.
- [19] Kalata, P. R., "The Tracking Index: A Generalized Parameter for α - β and α - β - γ Target Trackers," *The 22nd IEEE Conference on Decision and Control*, 1983, pp. 559–561.
- [20] Gray, J. E., and Murray, W. J., "A Derivation of an Analytic Expression for the Tracking Index for the Alpha-Beta-Gamma Filter," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 3, 1993, pp. 1064–1065.

- [21] Mookerjee, P., and Reifler, F., "Reduced State Estimators for Consistent Tracking of Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 41, No. 2, 2005, pp. 608–619.
- [22] Hasan, A. H., and Grachev, A., "Adaptive α - β -filter for Target Tracking Using Real Time Genetic Algorithm," *Journal of Electrical and Control Engineering*, Vol. 3, 2013, pp. 32–38.
- [23] Oshman, Y., and Arad, D., "Enhanced Air-to-Air Missile Tracking Using Target Orientation Observations," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 4, 2004, pp. 595–606.
- [24] Prevost, C. G., Desbiens, A., and Gagnon, E., "Extended Kalman Filter for State Estimation and Trajectory Prediction of a Moving Object Detected by an Unmanned Aerial Vehicle," *American Control Conference*, IEEE, 2007, pp. 1805–1810.
- [25] Zhou, H., and Kumar, K., "A 'Current' Statistical Model and Adaptive Algorithm for Estimating Maneuvering Targets," *Journal of Guidance, Control, and Dynamics*, Vol. 7, No. 5, 1984, pp. 596–602.
- [26] Verma, S., Sanjeevini, S., Sumer, E. D., and Bernstein, D. S., "Real-time Numerical Differentiation of Sampled Data Using Adaptive Input and State Estimation," *International Journal of Control*, 2024, pp. 1–13.
- [27] Verma, S., Lai, B., and Bernstein, D. S., "Adaptive Real-Time Numerical Differentiation with Variable-Rate Forgetting and Exponential Resetting," *Proc. Amer. Contr. Conf.*, 2024, pp. 3103–3108.
- [28] Verma, S., Sanjeevini, S., Sumer, E. D., Girard, A., and Bernstein, D. S., "On the Accuracy of Numerical Differentiation Using High-Gain Observers and Adaptive Input Estimation," *Proc. Amer. Contr. Conf.*, 2022, pp. 4068–4073.
- [29] 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.
- [30] Åström, K. J., Borisson, U., et al., "Theory and Applications of Self-Tuning Regulators," *Automatica*, Vol. 13, No. 5, 1977, pp. 457–476.
- [31] Lai, B., and Bernstein, D. S., "Exponential Resetting and Cyclic Resetting Recursive Least Squares," *IEEE Contr. Sys. Lett.*, Vol. 7, 2022, pp. 985–990.
- [32] Malik, O., Hope, G., and Cheng, S., "Some Issues on the Practical Use of Recursive Least Squares Identification in Self-Tuning Control," *Int. J. Contr.*, Vol. 53, No. 5, 1991, pp. 1021–1033.
- [33] 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.
- [34] McKeon, J. J., "F Approximations to the Distribution of Hotelling's T_0^2 ," *Biometrika*, Vol. 61, No. 2, 1974, pp. 381–383.