Real-Time Numerical Differentiation of Sampled Data Using Adaptive Input and State Estimation

Shashank Verma[a], Sneha Sanjeevini[a], E. Dogan Sumer,[b] and Dennis S. Bernstein[a]

[a]Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan, USA
[b]Ford Motor Company, Dearborn, Michigan, USA

ABSTRACT

Real-time numerical differentiation plays a crucial role in many digital control algorithms, such as PID control, which requires numerical differentiation to implement derivative action. This paper proposes an algorithm for estimating the numerical derivative of a signal from noisy sampled data measurements. The method uses adaptive input estimation with adaptive state estimation (AIE/ASE), and thus it requires only minimal prior information about the signal and noise statistics. Furthermore, since the estimates of the derivative at step k provided by AIE/ASE depend only on data available up to step k, AIE/ASE is thus implementable in real time. The accuracy of AIE/ASE is compared numerically to several conventional numerical differentiation methods. Finally, AIE/ASE is applied to simulated vehicle position data, generated in the CarSim simulator software.

KEYWORDS

Numerical differentiation, Adaptive system, Kalman filter, Input estimation

1. Introduction

The dual operations of integration and differentiation provide the foundation for much of mathematics. Analytically, differentiation is often considered less complex than integration, as evidenced by the relative difficulty encountered in differentiating versus integrating functions such as $\log(1+\sin^2 x^3)$. In numerical analysis, integration techniques have been extensively developed in Davis and Rabinowitz (1984), whereas differentiation techniques have been developed more sporadically in Cullum (1971); Savitzky and Golay (1964), Hamming (1973, pp. 565, 566).

In practice, numerical integration and differentiation techniques are applied to sequences of measurements, that is, discrete-time signals composed of sampled data. Although strictly speaking, integration and differentiation are defined on continuous spaces and not for discrete-time signals, the goal is to compute a discrete-time "integral" or "derivative" estimate that approximates the true integral or derivative of the pre-sampled, analog signal.

In addition to the effect of sampling, numerical integration and differentiation methods must address the effect of sensor noise in sampled data. For numerical integration of sampled data, constant noise in data, that is, bias, leads to a spurious ramp, while stochastic noise leads to random-walk divergence due to the numerical integration of

CONTACT Shashank Verma. Email: shaaero@umich.edu
CONTACT Sneha Sanjeevini. Email: snehasnj@umich.edu
CONTACT E. Dogan Sumer. Email: esumer@ford.com
CONTACT Dennis S. Bernstein. Email: dsbaero@umich.edu

noise present in the data. Mitigation of these effects is of extreme importance in applications such as inertial navigation as shown in Farrell (2008); Grewel, Andrews, and Bartone (2020).

Compared to numerical integration, the effect of noise on numerical differentiation is far more severe. This situation is due to the fact that, whereas integration is a bounded operator on a complete inner-product space, differentiation is an unbounded operator on a dense subspace. Unboundedness implies a lack of continuity, which is manifested as high sensitivity to sensor noise. Consequently, numerical differentiation typically involves assumptions on the smoothness of the signal and spectrum of the noise as considered in Ahn, Choi, and Ramm (2006); Haimovich, Seeber, Aldana-López, and Gómez-Gutiérrez (2022); Jauberteau and Jauberteau (2009); Knowles and Renka (2014); Listmann and Zhao (2013); Stickel (2010).

Numerical differentiation algorithms are crucial elements of many digital control algorithms. For example, PID control requires numerical differentiation to implement derivative action as presented in Astrom and Hagglund (2006); Vilanova and Visioli (2012). Flatness-based control is based on a finite number of derivatives as shown in Mboup, Join, and Fliess (2009); Nieuwstadt, Rathinam, and Murray (1998). In feedback control applications, real-time implementation of numerical differentiation algorithms is essential. However, the phase shift and latency associated with numerical differentiation can result in performance degradation and even instability. Phase shift arises from filtering, whereas latency arises from noncausal numerical differentiation, that is, numerical differentiation algorithms that require future data. For real-time applications, a noncausal differentiation algorithm that requires data at future time steps can be implemented causally by delaying the computation until the required data are available. For feedback controllers that require an estimate of the *current* derivative, the delayed estimate provided by a noncausal differentiation algorithm may not be a sufficiently accurate estimate of the required derivative.

In practice, analog or digital filters are used to suppress the effect of sensor noise, thereby allowing the use of differencing formulae in the form of inverted "V" filters, which have the required gain and phase lead at low frequencies and roll off at high frequencies. These techniques assume that the characteristics of the signal and noise are known, thereby allowing the user to tweak the filter parameters. When both the true signal and the noise have characteristics that are unknown and may change over time, filter tuning becomes impossible, significantly increasing the challenge of the problem. The recent work in Van Breugel, Kutz, and Brunton (2020) articulates these challenges and proposes a Pareto-tradeoff technique for addressing the absence of prior information. Additional techniques include high-gain observer methods, where the observer approximates the dynamics of a differentiator as shown in Dabroom and Khalil (1999). Peng Li and Parisini (2018) employed a kernel-based deadbeat observer for numerical differentiation, utilizing Volterra integral operators. Numerical differentiation based on integration using Jacobi polynomials was introduced in Da-yan Liu and Perruquetti (2011). Yet another approach is to apply sliding-mode algorithms as shown in Alwi and Edwards (2013); Levant (1998, 2003); López-Caamal and Moreno (2019); Mojallizadeh, Brogliato, and Acary (2021); Reichhartinger and Spurgeon (2018). Ibrir and Diop (2004) presented a method involving a simplified linear optimization problem to deduce a continuous spline signal, aiding in the estimation of the derivative of sampled data. Additionally, Polyakov, Efimov, and Perruquetti (2014) analyzed a homogeneous differentiator based on the implicit Lyapunov function method.

Another approach to numerical differentiation is to apply state estimation with integrator dynamics, where the state estimate includes an estimate of the derivative of

the measurement as shown in Bogler (1987); Kalata (1984). This approach has been widely used for target and vehicle tracking in H. Khaloozadeh (2009); Jia, Balasuriya, and Challa (2008); Lee and Tahk (1999); Rana, Halim, Rahamna, and Abdelhadi (2020). As an extension of state estimation, the present paper applies input estimation to numerical differentiation, where the goal is to estimate the input as well as the state. Input and state estimation methods are discussed in Alenezi, Zhang, Hui, and Zak (2021); Fang, Shi, and Yi (2011); Gillijns and De Moor (2007); Hsieh (2017); Naderi and Khorasani (2019); Orjuela, Marx, Ragot, and Maquin (2009); Yong, Zhu, and Frazzoli (2016).

The present paper is motivated by the situation where minimal prior information about the signal and noise is available. This case arises when the spectrum of the signal changes slowly or abruptly in an unknown way, and when the noise characteristics vary due to changes in the environment, such as weather. With this motivation, adaptive input estimation (AIE) was applied to target tracking in Ansari and Bernstein (2019), where it was used to estimate vehicle acceleration using position data. In particular, the approach of Ansari and Bernstein (2019) is based on retrospective cost input estimation (RCIE), where recursive least squares (RLS) is used to update the coefficients of the estimation subsystem. The error metric used for adaptation is the residual (innovations) of the state estimation algorithm, that is, the Kalman filter. This technique requires specification of the covariances of the process noise, input-estimation error, and sensor noise.

The present paper extends the approach of Ansari and Bernstein (2019) by replacing the Kalman filter with an adaptive Kalman filter in which the input-estimation-error covariance and the sensor-noise covariance are updated online. Adaptive extensions of the Kalman filter to the case where the variance of the disturbance is unknown are considered in Moghe, Zanetti, and Akella (2019); Shi, Han, and Liang (2009); Yaesh and Shaked (2008); Zhang et al. (2020). Adaptive Kalman filters based on the residual for integrating INS/GPS systems are discussed in Almagbile, Wang, and Ding (2010); Hide, Moore, and Smith (2003); Mohamed and Schwarz (1999). Several approaches to adaptive filtering, such as Bayesian, maximum likelihood, correlation, and covariance matching, are studied in Mehra (1972). A related algorithm involving a covariance constraint is developed in Mook and Junkins (1988).

The adaptive Kalman filter used in the present paper as part of adaptive input estimation with adaptive state estimation (AIE/ASE) is based on a search over the range of input-estimation error covariance. This technique has proven to be easy to implement and effective in the presence of unknown signal and noise characteristics. The main contribution of the present paper is a numerical investigation of the accuracy of AIE combined with the proposed adaptive state estimation (ASE) in the presence of noise with unknown properties. The accuracy of AIE/ASE is compared to the backward-difference differentiation, Savitzky-Golay differentiation (Mboup et al. (2009); Savitzky and Golay (1964); Schafer (2011); Staggs (2005)), and numerical differentiation based on high-gain observers (Dabroom and Khalil (1999)).

The present paper represents a substantial extension of preliminary results presented in Verma, Sanjeevini, Sumer, Girard, and Bernstein (2022). In particular, the algorithms presented in the present paper extend the adaptive estimation component of the approach of Verma et al. (2022) in Section 5, and the accuracy of these algorithms is more extensively evaluated and compared to prior methods in Section 6.

The contents of the paper are as follows. In section 2, we identify the challenges that arise from implementing numerical differentiation algorithms in real time. These challenges are primarily due to the delay in the availability of the estimated derivative,

which results from computation time and non-causality. This section also defines an error metric for comparing the accuracy of the algorithms considered in this paper. Section 3 summarizes three baseline numerical differentiation algorithms and identifies their limitations, which motivates the proposed algorithm. Section 4 summarizes the adaptive input estimation algorithm. Section 5 provides the paper's main contribution, namely, adaptive input estimation with adaptive state estimation (AIE/ASE), along with its two other variations. Section 6 applies three variations of AIE using harmonic signals with various noise levels. Finally, Section 7 applies the variations of AIE to simulated vehicle position data generated by CarSim.

2. Problem Statement and Error Metric

This section presents the problem statement and error metric used to assess the accuracy of the algorithms presented in this paper. The error metric is specifically chosen to reflect the implications of real-time implementation.

2.1. Problem Statement

Let y be a continuous-time signal with qth derivative $y^{(q)}$. We assume that the sampled values $y_k \triangleq y(kT_\mathrm{s})$ are available, where T_s is the sample time. The goal is to use the sampled values y_k to obtain an estimate $\hat{y}_k^{(q)}$ of $y_k^{(q)} \triangleq y^{(q)}(kT_\mathrm{s})$ in the presence of measurement noise with unknown properties. This paper focuses on the cases $q = 1$ and $q = 2$.

2.2. Real-Time Implementation and Error Metric

The time T_c required for computation in numerical differentiation invariably results in a delay of δ time steps before the estimated derivative becomes available. In this paper, we assume that $T_\mathrm{c} \leq T_\mathrm{s}$, and thus the delay due to computation time is $\delta = 1$.

This paper considers both causal and noncausal differentiation methods. To estimate the derivative at the current step, causal differentiation does not require future data; in contrast, noncausal differentiation utilizes future data. For real-time implementation, causal differentiation entails a delay of $\delta = 1$ step due to the computation time T_c, whereas noncausal differentiation entails a delay of $\delta \geq 2$ steps. For the case $\delta = 1$, Figure 1 shows that the estimate $\hat{y}_k^{(q)}$ of $y_k^{(q)}$ is not available until step $k + 1$.

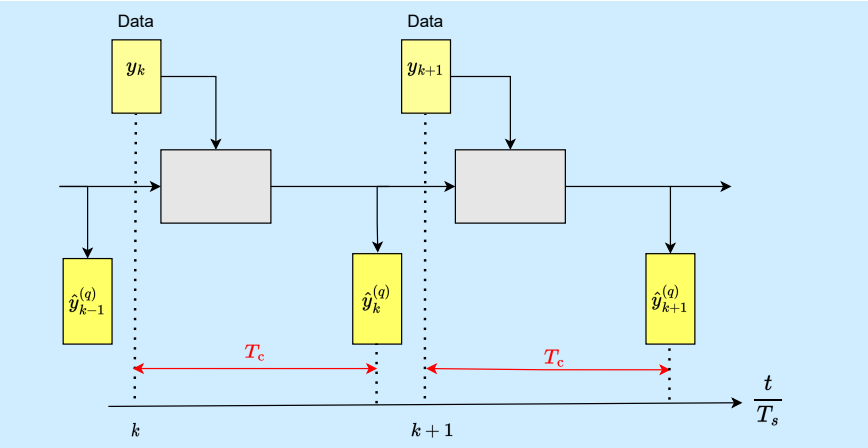

Figure 1.: *Timing diagram for causal numerical differentiation.* The causal numerical differentiator uses data obtained at step k to estimate the derivative of the signal y. Because of the computation time T_c, the estimate $\hat{y}_k^{(q)}$ of $y_k^{(q)}$ is not available until step $k+1$. In this case, the delay is $\delta = 1$ step. For noncausal differentiation, $\delta \geq 2$.

To quantify the accuracy of each numerical differentiation algorithm, for all $k \geq \delta$, we define the relative root-mean-square error (RMSE) of the estimate of the qth derivative as

$$\rho_k^{(q)} \triangleq \sqrt{\frac{\sum_{i=\delta}^{k} (y_i^{(q)} - \hat{y}_{i-\delta}^{(q)})^2}{\sum_{i=\delta}^{k} (y_{i-\delta}^{(q)})^2}}. \tag{1}$$

Note that the numerator of (1) accounts for the effect of the delay δ. For real-time implementation, the relevant error metric depends on the difference between the true current derivative and the currently available estimate of the past derivative, as can be seen in the numerator of (1). When the derivative estimates are exact, (1) determines an RMSE value that can be viewed as the *delay floor* for the qth derivative, that is, the error due solely to the fact that a noncausal differentiation algorithm must be implemented with a suitable delay. Note that the delay floor depends on δ and is nonnegative.

The *true* values of $y_k^{(q)}$ are the sampled values of $y^{(q)}$ in the absence of sensor noise. Of course, the true values of $y_k^{(q)}$ are unknown in practice and thus cannot be used as an online error criterion. However, these values are used in (1), which is computable in simulation for comparing the accuracy of the numerical differentiation algorithms.

3. Comparison and Limitations of Baseline Algorithms

This section summarizes three algorithms for numerically differentiating sampled data. These algorithms provide a baseline for evaluating the accuracy of the adaptive input and state estimation algorithms described in Section 5.

3.1. Backward-Difference (BD) Differentiation

As define in Astrom and Hagglund (2006), let $\mathbf{q}^{-1}$ denote the backward-shift operator. Then the *backward-difference single differentiator* is given by

$$G_{\mathrm{sd}}(\mathbf{q}^{-1}) \triangleq \frac{1-\mathbf{q}^{-1}}{T_{\mathrm{s}}}, \tag{2}$$

and the *backward-difference double differentiator* is given by

$$G_{\mathrm{dd}}(\mathbf{q}^{-1}) \triangleq \frac{(1-\mathbf{q}^{-1})^2}{T_{\mathrm{s}}^2}. \tag{3}$$

3.2. Savitzky–Golay (SG) Differentiation

As shown in Savitzky and Golay (1964); Schafer (2011); Staggs (2005), in SG differentiation at each step k, a polynomial

$$P_k(s) = \sum_{i=0}^{p_{\mathrm{d}}} a_{i,k} s^i \tag{4}$$

of degree p_{d} is fit over a sliding data window of size $2\ell+1$ centered at step k, where $\ell \geq 1$. At each step k, this leads to the least-squares problem

$$\min \|\mathcal{Y}_k - \mathcal{A}_k \mathcal{X}_k\|, \tag{5}$$

where

$$\mathcal{Y}_k \triangleq \begin{bmatrix} y_{k-\ell} \\ \vdots \\ y_{k+\ell} \end{bmatrix}, \quad \mathcal{X}_k \triangleq \begin{bmatrix} a_{0,k} \\ \vdots \\ a_{p_{\mathrm{d}},k} \end{bmatrix}, \tag{6}$$

$$\mathcal{A}_k \triangleq \begin{bmatrix} 1 & (k-\ell)T_{\mathrm{s}} & \dots & ((k-\ell)T_{\mathrm{s}})^{p_{\mathrm{d}}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (k+\ell)T_{\mathrm{s}} & \dots & ((k+\ell)T_{\mathrm{s}})^{p_{\mathrm{d}}} \end{bmatrix}. \tag{7}$$

Solving (5) with $q \leq p_{\mathrm{d}} \leq 2\ell$ yields

$$\hat{\mathcal{X}}_k = \begin{bmatrix} \hat{a}_{0,k} \\ \vdots \\ \hat{a}_{p_{\mathrm{d}},k} \end{bmatrix}. \tag{8}$$

Differentiating (4) q times with respect to s, setting $s = kT_{\mathrm{s}}$, and replacing the coefficients of P_k in (4) with the components of $\hat{\mathcal{X}}_k$, the estimate $\hat{y}_k^{(q)}$ of $y_k^{(q)}$ is given

by

$$\hat{y}_k^{(q)} = \sum_{i=q}^{p_\mathrm{d}} Q_{i,q} \hat{a}_{i,k} (kT_\mathrm{s})^{i-q}, \tag{9}$$

where, for all $i = q, \ldots, p_\mathrm{d}$,

$$Q_{i,q} \triangleq \prod_{j=1}^{q} (i - j + 1). \tag{10}$$

3.3. High-Gain-Observer (HGO) Differentiation

A state space model for the rth-order continuous-time HGO in Dabroom and Khalil (1999) is given by

$$\dot{\hat{x}} = A_\mathrm{co}\hat{x} + B_\mathrm{co}y, \quad \hat{y} = C_\mathrm{o}\hat{x}, \tag{11}$$

$$A_\mathrm{co} \triangleq \begin{bmatrix} 0_{(r-1)\times 1} & I_{r-1} \\ 0 & 0_{1\times(r-1)} \end{bmatrix} - H \begin{bmatrix} 1 & 0_{1\times(r-1)} \end{bmatrix}, \tag{12}$$

$$C_\mathrm{o} \triangleq \begin{bmatrix} 0_{(r-1)\times 1} & I_{r-1} \end{bmatrix}, \tag{13}$$

$$B_\mathrm{co} = H \triangleq \begin{bmatrix} \dfrac{\alpha_1}{\varepsilon} & \dfrac{\alpha_2}{\varepsilon^2} & \cdots & \dfrac{\alpha_r}{\varepsilon^r} \end{bmatrix}^\mathsf{T}, \tag{14}$$

where $\varepsilon > 0$ and $\alpha_1, \ldots, \alpha_r$ are constants chosen such that the polynomial

$$p(s) \triangleq s^r + \alpha_1 s^{r-1} + \cdots + \alpha_{r-1} s + \alpha_r \tag{15}$$

is Hurwitz. The transfer function from y to $\hat{y}$ is given by

$$G(s) = C_\mathrm{o}(sI - A_\mathrm{co})^{-1} H = D_G^{-1}(s) N_G(s), \tag{16}$$

where

$$D_G(s) \triangleq \varepsilon^r s^r + \alpha_1 \varepsilon^{r-1} s^{r-1} + \cdots + \alpha_{r-1} \varepsilon s + \alpha_r, \tag{17}$$

$$N_G(s) \triangleq \begin{bmatrix} \alpha_2 \varepsilon^{r-2} s^{r-1} + \cdots + \alpha_{r-1} \varepsilon s^2 + \alpha_r s \\ \alpha_3 \varepsilon^{r-3} s^{r-1} + \cdots + \alpha_{r-1} \varepsilon s^3 + \alpha_r s^2 \\ \vdots \\ \alpha_{r-1} \varepsilon s^{r-1} + \alpha_r s^{r-2} \\ \alpha_r s^{r-1} \end{bmatrix}. \tag{18}$$

Since

$$\lim_{\varepsilon \to 0} G(s) = \begin{bmatrix} s & s^2 & \cdots & s^{r-1} \end{bmatrix}^\mathsf{T}, \tag{19}$$

it follows that, for all $i = 1, \ldots, r-1$, the ith component of $\hat{y}$ is an approximation of $y^{(i)}$. Applying the bilinear transformation to (11) yields the discrete-time observer

$$\hat{x}_{k+1} = A_{\mathrm{do}}\hat{x}_k + B_{\mathrm{do}}y_k, \quad \hat{y}_k = C_{\mathrm{o}}\hat{x}_k, \tag{20}$$

where

$$A_{\mathrm{do}} \triangleq (I_{\mathrm{r}} - \tfrac{1}{2}T_{\mathrm{s}}A_{\mathrm{co}})^{-1}(I_{\mathrm{r}} + \tfrac{1}{2}T_{\mathrm{s}}A_{\mathrm{co}}), \tag{21}$$

$$B_{\mathrm{do}} \triangleq (I_{\mathrm{r}} - \tfrac{1}{2}T_{\mathrm{s}}A_{\mathrm{co}})^{-1}B_{\mathrm{co}}T_{\mathrm{s}}. \tag{22}$$

Implementation of (20) provides estimates $\hat{y}_k^{(1)}, \ldots, \hat{y}_k^{(r-1)}$ of $y_k^{(1)}, \ldots, y_k^{(r-1)}$.

Several noteworthy differences exist among BD, SG, and HGO. First, BD differentiation operates on adjacent pairs of data points, whereas SG differentiation operates on a moving window of data points. Consequently, SG differentiation is potentially more accurate than BD differentiation.

To compare the presented baseline algorithms, we consider numerical differentiation of the continuous-time signal $y(t) = \sin(20t)$, where t is time in seconds. The signal $y(t)$ is sampled with sample time $T_{\mathrm{s}} = 0.01$ sec. The measurements are assumed to be corrupted by noise, and thus the noisy sampled signal is given by $y_k = \sin(0.2k) + Dv_k$, where v_k is standard (zero-mean, unit-variance, Gaussian) white noise. The value of D is chosen to set the desired signal-to-noise ratio (SNR).

For single differentiation with SG, let $\ell = 2$ and $p_d = 3$. For single differentiation with HGO, let HGO/1 denote HGO with $r = 2$, $\alpha_1 = 2$, $\alpha_2 = 1$, and $\varepsilon = 0.2$, and let HGO/2 denote HGO/1 with $\varepsilon = 0.2$ replaced by $\varepsilon = 0.7$. Note that $\delta = 1$ for BD and HGO, whereas $\delta = \ell + 1$ for SG with window size $2\ell + 1$. Figure 2 shows the relative RMSE $\rho_{k_{\mathrm{f}}}^{(1)}$ of the estimate of the first derivative for SNR ranging from 20 dB to 60 dB, where $k_{\mathrm{f}} = 2000$ steps.

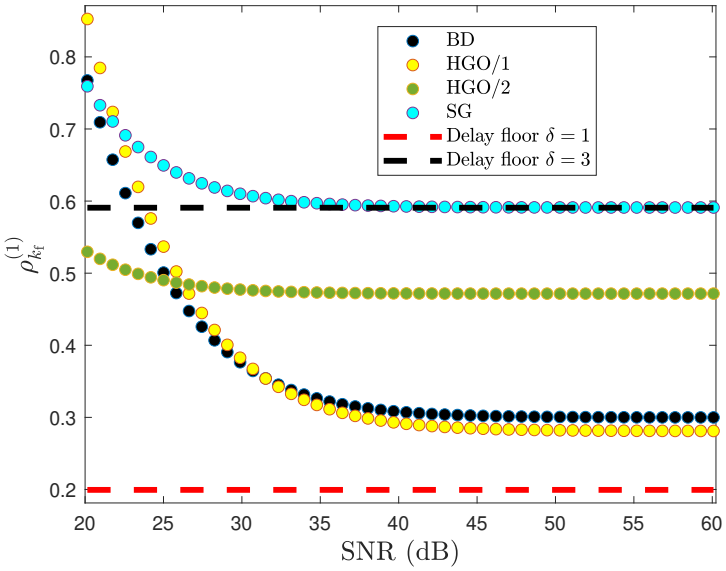

Figure 2.: Relative RMSE $\rho^{(1)}_{k_f}$ of the estimate of the first derivative versus SNR, where $k_f = 2000$ steps, for BD, SG, HGO/1, and HGO/2. For the first derivative, the red dashed line denotes the delay floor for $\delta = 1$, and the black dashed line denotes the delay floor for $\delta = 3$.

The comparison between HGO/1 and HGO/2 in Figure 2 shows that the performance of HGO differentiation depends on the noise level, and thus tuning is needed to achieve the best possible performance. When the noise level is unknown, however, this tuning is not possible. Hence, we now consider a differentiation technique that adapts to the actual noise characteristics.

4. Adaptive Input Estimation

This section summarizes adaptive input estimation (AIE), which is a specialization of retrospective cost input estimation (RCIE) derived in Ansari and Bernstein (2019). This section explains how AIE specializes RCIE to the problem of causal numerical differentiation.

Consider the linear discrete-time system

$$x_{k+1} = Ax_k + Bd_k, \tag{23}$$

$$y_k = Cx_k + D_2v_k, \tag{24}$$

where $k \geq 0$ is the step, $x_k \in \mathbb{R}^n$ is the state, $d_k \triangleq d(kT_s) \in \mathbb{R}$, $v_k \in \mathbb{R}$ is standard white noise, and $D_2v_k \in \mathbb{R}$ is the sensor noise. The matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, $C \in \mathbb{R}^{1\times n}$, and $D_2 \in \mathbb{R}$ are assumed to be known. Define the sensor-noise covariance $V_2 \triangleq D_2D_2^{\mathsf{T}}$. The *goal* of AIE is to estimate d_k and x_k.

AIE consists of 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

$$x_{\mathrm{fc},k+1} = Ax_{\mathrm{da},k} + B\hat{d}_k, \tag{25}$$
$$y_{\mathrm{fc},k} = Cx_{\mathrm{fc},k}, \tag{26}$$
$$z_k = y_{\mathrm{fc},k} - y_k, \tag{27}$$

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

Next, to obtain $\hat{d}_k$, the input-estimation subsystem of order n_{e} is given by

$$\hat{d}_k = \sum_{i=1}^{n_{\mathrm{e}}} P_{i,k}\hat{d}_{k-i} + \sum_{i=0}^{n_{\mathrm{e}}} Q_{i,k}z_{k-i}, \tag{28}$$

where $P_{i,k} \in \mathbb{R}$ and $Q_{i,k} \in \mathbb{R}$ are time-varying coefficients. Note that (28) represents an exactly proper transfer function. AIE minimizes z_k by using recursive least squares (RLS) to update $P_{i,k}$ and $Q_{i,k}$ as shown below. The subsystem (28) can be reformulated as

$$\hat{d}_k = \Phi_k\theta_k, \tag{29}$$

where the regressor matrix Φ_k is defined by

$$\Phi_k \triangleq \begin{bmatrix}\hat{d}_{k-1} & \cdots & \hat{d}_{k-n_{\mathrm{e}}} & z_k & \cdots & z_{k-n_{\mathrm{e}}}\end{bmatrix} \in \mathbb{R}^{1\times l_\theta}, \tag{30}$$

the coefficient vector θ_k is defined by

$$\theta_k \triangleq \begin{bmatrix}P_{1,k} & \cdots & P_{n_{\mathrm{e}},k} & Q_{0,k} & \cdots & Q_{n_{\mathrm{e}},k}\end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{l_\theta}, \tag{31}$$

and $l_\theta \triangleq 2n_{\mathrm{e}} + 1$.

In terms of the backward-shift operator $\mathbf{q}^{-1}$, (28) can be written as

$$\hat{d}_k = G_{\hat{d}z,k}(\mathbf{q}^{-1})z_k, \tag{32}$$

where

$$G_{\hat{d}z,k} \triangleq D_{\hat{d}z,k}^{-1}N_{\hat{d}z,k}, \tag{33}$$
$$D_{\hat{d}z,k}(\mathbf{q}^{-1}) \triangleq I_{l_d} - P_{1,k}\mathbf{q}^{-1} - \cdots - P_{n_{\mathrm{e}},k}\mathbf{q}^{-n_{\mathrm{e}}}, \tag{34}$$
$$N_{\hat{d}z,k}(\mathbf{q}^{-1}) \triangleq Q_{0,k} + Q_{1,k}\mathbf{q}^{-1} + \cdots + Q_{n_{\mathrm{e}},k}\mathbf{q}^{-n_{\mathrm{e}}}. \tag{35}$$

To update the coefficient vector θ_k, we define the filtered signals

$$\Phi_{\mathrm{f},k} \triangleq G_{\mathrm{f},k}(\mathbf{q}^{-1})\Phi_k, \quad \hat{d}_{\mathrm{f},k} \triangleq G_{\mathrm{f},k}(\mathbf{q}^{-1})\hat{d}_k, \tag{36}$$

where, for all $k \geq 0$,

$$G_{\mathrm{f},k}(\mathbf{q}^{-1}) = \sum_{i=1}^{n_{\mathrm{f}}} \mathbf{q}^{-i} H_{i,k}, \tag{37}$$

$$H_{i,k} \triangleq \begin{cases} CB, & k \geq i = 1, \\ C\overline{A}_{k-1} \cdots \overline{A}_{k-(i-1)} B, & k \geq i \geq 2, \\ 0, & i > k, \end{cases} \tag{38}$$

and $\overline{A}_k \triangleq A(I + K_{\mathrm{da},k} C)$, where $K_{\mathrm{da},k}$ is the Kalman filter gain given by (44) below.

Furthermore, define the *retrospective variable*

$$z_{\mathrm{r},k}(\hat{\theta}) \triangleq z_k - (\hat{d}_{\mathrm{f},k} - \Phi_{\mathrm{f},k} \hat{\theta}), \tag{39}$$

where the coefficient vector $\hat{\theta} \in \mathbb{R}^{l_\theta}$ denotes a variable for optimization, and define the retrospective cost function

$$\mathcal{J}_k(\hat{\theta}) \triangleq \sum_{i=0}^{k} [R_z z_{\mathrm{r},i}^2(\hat{\theta}) + R_d(\Phi_i \hat{\theta})^2] + (\hat{\theta} - \theta_0)^{\mathsf{T}} R_\theta (\hat{\theta} - \theta_0), \tag{40}$$

where $R_z \in (0, \infty)$, $R_d \in (0, \infty)$, and $R_\theta \in \mathbb{R}^{l_\theta \times l_\theta}$ is positive definite. Then, for all $k \geq 0$, the unique global minimizer θ_{k+1} of (40) is given by the RLS update as shown in Islam and Bernstein (2019)

$$P_{k+1} = P_k - P_k \widetilde{\Phi}_k^{\mathsf{T}} \Gamma_k \widetilde{\Phi}_k P_k, \tag{41}$$

$$\theta_{k+1} = \theta_k - P_k \widetilde{\Phi}_k^{\mathsf{T}} \Gamma_k (\widetilde{z}_k + \widetilde{\Phi}_k \theta_k), \tag{42}$$

where

$$P_0 \triangleq R_\theta^{-1}, \quad \Gamma_k \triangleq (\widetilde{R}^{-1} + \widetilde{\Phi}_k P_k \widetilde{\Phi}_k^{\mathsf{T}})^{-1}, \quad \widetilde{\Phi}_k \triangleq \begin{bmatrix} \Phi_{\mathrm{f},k} \\ \Phi_k \end{bmatrix},$$

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

Using the updated coefficient vector given by (42), the estimated input at step $k+1$ is given by replacing k by $k+1$ in (29). We choose $\theta_0 = 0$, and thus $\hat{d}_0 = 0$. Implementation of AIE requires that the user specify the orders n_{e} and n_{f}, as well as the weightings R_z, R_d, and R_θ. These parameters are specified for each example in the paper.

4.1. State Estimation

The forecast variable $x_{\mathrm{fc},k}$ given by (25) is used to obtain the estimate $x_{\mathrm{da},k}$ of x_k given by the Kalman filter data-assimilation step

$$x_{\mathrm{da},k} = x_{\mathrm{fc},k} + K_{\mathrm{da},k} z_k, \tag{43}$$

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

$$K_{\mathrm{da},k} = -P_{\mathrm{f},k}C^{\mathsf{T}}(CP_{\mathrm{f},k}C^{\mathsf{T}} + V_{2,k})^{-1}, \tag{44}$$

$$P_{\mathrm{da},k} = (I_n + K_{\mathrm{da},k}C)P_{\mathrm{f},k}, \tag{45}$$

$$P_{\mathrm{f},k+1} = AP_{\mathrm{da},k}A^{\mathsf{T}} + V_{1,k}, \tag{46}$$

where $P_{\mathrm{f},0} = 0$ and $V_{1,k} \triangleq B\mathrm{var}(d_k - \hat{d}_k)B^{\mathsf{T}} + A\mathrm{cov}(x_k - x_{\mathrm{da},k}, d_k - \hat{d}_k)B^{\mathsf{T}} + B\mathrm{cov}(d_k - \hat{d}_k, x_k - x_{\mathrm{da},k})A^{\mathsf{T}}$.

4.2. Application of AIE to Numerical Differentiation

To apply AIE to causal numerical differentiation, (23) and (24) are used to model a discrete-time integrator. AIE then yields an estimate $\hat{d}_k$ of the derivative of the sampled output y_k. For single discrete-time differentiation, $A = 1, B = T_{\mathrm{s}}$, and $C = 1$, whereas, for double discrete-time differentiation,

$$A = \begin{bmatrix} 1 & T_{\mathrm{s}} \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{2}T_{\mathrm{s}}^2 \\ T_{\mathrm{s}} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{47}$$

5. Adaptive Input and State Estimation

In practice, $V_{1,k}$ and $V_{2,k}$ may be unknown in (46) and (44). To address this problem, three versions of AIE are presented. In each version, $V_{1,k}$ and $V_{2,k}$ may or may not be adapted. These versions are summarized in Table 1.

	$V_{1,k}$ Adaptation	$V_{2,k}$ Adaptation
AIE/NSE	No	No
AIE/SSE	Yes	No
AIE/ASE	Yes	Yes

Table 1.: Definitions of AIE/NSE, AIE/SSE, and AIE/ASE. Each version of AIE is determined by whether or not $V_{1,k}$ and/or $V_{2,k}$ is adapted in the state-estimation subsystem.

To adapt $V_{1,k}$ and $V_{2,k}$, at each step k we define the computable performance metric

$$J_k(V_1, V_2) \triangleq |\widehat{S}_k - S_k|, \tag{48}$$

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

$$\widehat{S}_k = \frac{1}{k}\sum_{i=0}^{k}(z_i - \overline{z}_k)^2, \tag{49}$$

$$\overline{z}_k = \frac{1}{k+1}\sum_{i=0}^{k} z_i, \tag{50}$$

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

$$S_k \triangleq C(AP_{\mathrm{da},k-1}A^{\mathsf{T}} + V_1)C^{\mathrm{T}} + V_2. \tag{51}$$

Note that (48) is the difference between the theoretical and empirical variances of z_k, which provides an indirect measure of the accuracy of V_1 and V_2.

5.1. AIE with Non-adaptive State Estimation (AIE/NSE)

In AIE/NSE, V_1 is fixed at a user-chosen constant value, and V_2 is assumed to be known and fixed constant at its true value. AIE/NSE is thus a specialization of AIE with $V_{1,k} \equiv V_1$ in (46) and $V_{2,k} \equiv V_{2,\mathrm{true}}$ in (44), where $V_{2,\mathrm{true}}$ is the true value of the sensor-noise covariance. A block diagram of AIE/NSE is shown in Figure 3.

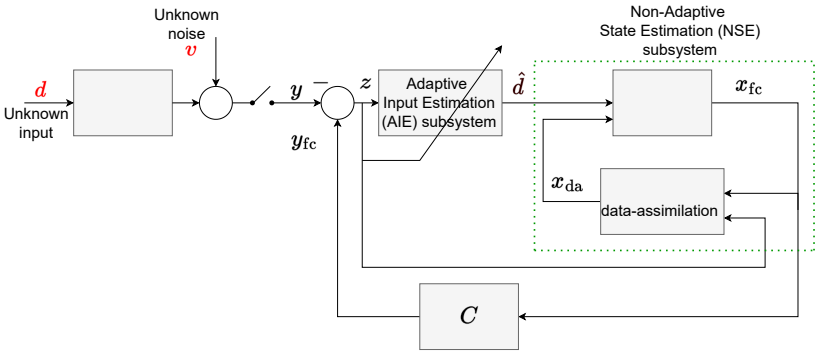

Figure 3.: Block diagram of AIE/NSE. The unknown input d is the signal whose estimates are desired, v is sensor noise, and y is the noisy measurement. In this version of AIE, V_1 is fixed at a user-chosen value and V_2 is fixed at its true value. The state estimator is thus not adaptive.

5.2. AIE with Semi-adaptive State Estimation (AIE/SSE)

In AIE/SSE, V_1 is adapted, and V_2 is assumed to be a known and fixed constant at its true value. Let $V_{1,\mathrm{adapt},k}$ denote the adapted value of $V_{1,k}$. AIE/SSE is thus a specialization of AIE with $V_{1,k} = V_{1,\mathrm{adapt},k}$ in (46) and $V_{2,k} \equiv V_{2,\mathrm{true}}$ in (44). For all $k \geq 0$, we assume that $V_{1,\mathrm{adapt},k} \triangleq \eta_k I_n$ and we define $\eta_k \in \mathbb{R}$ as

$$\eta_k \triangleq \underset{\eta\in[\eta_L,\eta_U]}{\arg\min}\ J_k(\eta I_n, V_{2,\mathrm{true}}), \tag{52}$$

where $0 \leq \eta_{\mathrm{L}} \leq \eta_{\mathrm{U}}$. Using (48) and (51) to rewrite (52) yields

$$\eta_k = \underset{\eta \in [\eta_{\mathrm{L}}, \eta_{\mathrm{U}}]}{\arg\min} \left| \widehat{S}_k - CAP_{\mathrm{da},k-1}A^{\mathsf{T}}C^{\mathsf{T}} - V_{2,\mathrm{true}} - \eta_k CC^{\mathsf{T}} \right|. \tag{53}$$

A block diagram of AIE/SSE is shown in Figure 4.

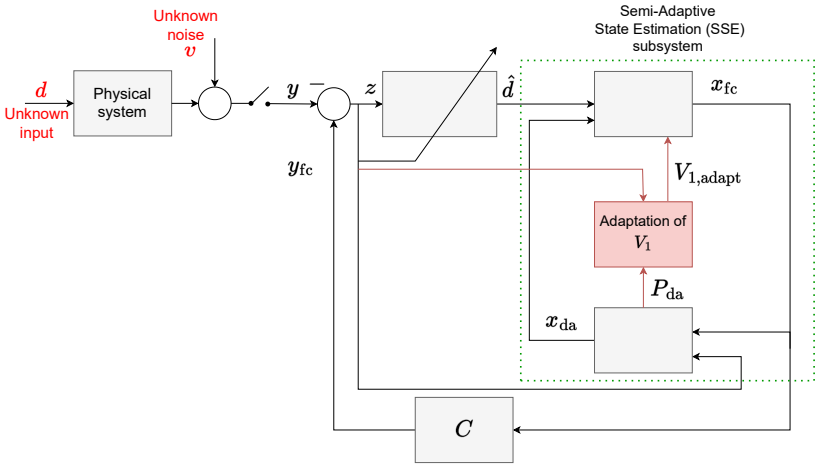

Figure 4.: Block diagram of AIE/SSE. In this version of AIE, V_1 is adapted and V_2 is fixed at its true value. The state estimator is thus semi-adaptive.

5.3. AIE with Adaptive State Estimation (AIE/ASE)

In AIE/ASE, both V_1 and V_2 are adapted. Let $V_{1,\mathrm{adapt},k} = \eta_k I_n$, where $\eta_k \geq 0$, and $V_{2,\mathrm{adapt},k}$ denote the adapted values of V_1 and V_2, respectively. Hence, AIE/ASE can be viewed as a specialized form of AIE, with $V_{1,k} = V_{1,\mathrm{adapt},k}$ in (46) and $V_{2,k} = V_{2,\mathrm{adapt},k}$ in (44). The objective is thus to determine $\eta_k \geq 0$ and $V_{2,\mathrm{adapt},k} \geq 0$ such that J_k in (48) is minimized, that is,

$$(\eta_k, V_{2,\mathrm{adapt},k}) \triangleq \underset{\eta \in [\eta_{\mathrm{L}}, \eta_{\mathrm{U}}], V_2 \geq 0}{\arg\min} J_k(\eta I_n, V_2), \tag{54}$$

where $0 \leq \eta_{\mathrm{L}} < \eta_{\mathrm{U}}$. The following result provides the minimizing values of η_k and $V_{2,\mathrm{adapt},k}$.

Proposition 5.1. Consider the optimization problem (54). Define the function $J_{\mathrm{f},k} \colon \mathbb{R}^{n \times n} \to \mathbb{R}$ by

$$J_{\mathrm{f},k}(V_1) \triangleq \widehat{S}_k - C(AP_{\mathrm{da},k-1}A^{\mathrm{T}} + V_1)C^{\mathrm{T}}, \tag{55}$$

and the set

$$\mathcal{J}_{\mathrm{f},k} \triangleq \{J_{\mathrm{f},k}(\eta I_n) \colon J_{\mathrm{f},k}(\eta I_n) > 0, \eta_{\mathrm{L}} \leq \eta < \eta_{\mathrm{U}}\} \subseteq (0, \infty). \tag{56}$$

If $\mathcal{J}_{\mathrm{f},k}$ is empty, then a minimizer $(\eta_k, V_{2,\mathrm{adapt},k})$ of (54) is given by

$$\eta_k = \underset{\eta\in[\eta_{\mathrm{L}},\eta_{\mathrm{U}}]}{\arg\min}\ |J_{\mathrm{f},k}(\eta I_n)|, \tag{57}$$

$$V_{2,\mathrm{adapt},k} = 0, \tag{58}$$

and the minimum value of J_k is

$$J_k(\eta_k I_n, V_{2,\mathrm{adapt},k}) = \widehat{S}_k - CAP_{\mathrm{da},k-1}A^{\mathrm{T}}C^{\mathrm{T}} - \eta_k CC^{\mathrm{T}}.$$

Now, assume that $\mathcal{J}_{\mathrm{f},k}$ is not empty, and let $\widehat{J}_{\mathrm{f},k} \in [\min \mathcal{J}_{\mathrm{f},k}, \max \mathcal{J}_{\mathrm{f},k}]$. Then, a minimizer $(\eta_k, V_{2,\mathrm{adapt},k})$ of (54) is given by

$$\eta_k = \underset{\eta\in[\eta_{\mathrm{L}},\eta_{\mathrm{U}}]}{\arg\min}\ |J_{\mathrm{f},k}(\eta I_n) - \widehat{J}_{\mathrm{f},k}|, \tag{59}$$

$$V_{2,\mathrm{adapt},k} = J_{\mathrm{f},k}(\eta_k I_n), \tag{60}$$

and the minimum value of J_k is

$$J_k(\eta_k I_n, V_{2,\mathrm{adapt},k}) = 0. \tag{61}$$

Proof. First note that, for all $\eta \in [\eta_{\mathrm{L}}, \eta_{\mathrm{U}}]$ and $V_2 \geq 0$,

$$J_k(\eta I_n, V_2) = |J_{\mathrm{f},k}(\eta I_n) - V_2|. \tag{62}$$

We first consider the case where $\mathcal{J}_{\mathrm{f},k}$ is empty. In this case, for all $\eta \in [\eta_{\mathrm{L}}, \eta_{\mathrm{U}}]$, $J_{\mathrm{f},k}(\eta I_n) \leq 0$. Hence it follows from (62) that (54) is minimized by (57) and (58).

Next, we consider the case where $\mathcal{J}_{\mathrm{f},k}$ is not empty, and thus $\widehat{J}_{\mathrm{f},k} > 0$. With η_k given by (59), it follows that $V_{1,\mathrm{adapt},k} = \eta_k I_n$. Hence, it follows from (59) and (60) that the minimum value of (54) is given by

$$\begin{aligned} J_k(V_{1,\mathrm{adapt},k}, V_{2,\mathrm{adapt},k}) &= |J_{\mathrm{f},k}(V_{1,\mathrm{adapt},k}) - V_{2,\mathrm{adapt},k}| \\ &= |J_{\mathrm{f},k}(\eta_k I_n) - J_{\mathrm{f},k}(\eta_k I_n)| \\ &= 0. \end{aligned}$$

□

Numerical examples show that

$$\widehat{J}_{\mathrm{f},k} = \tfrac{1}{2}[\min \mathcal{J}_{\mathrm{f},k} + \max \mathcal{J}_{\mathrm{f},k}] \tag{63}$$

yields a value of η_k that approximately minimizes the RMSE (1) of the estimate of the derivative. A block diagram of AIE/ASE is shown in Figure 5. AIE/ASE is summarized by Algorithm 1.

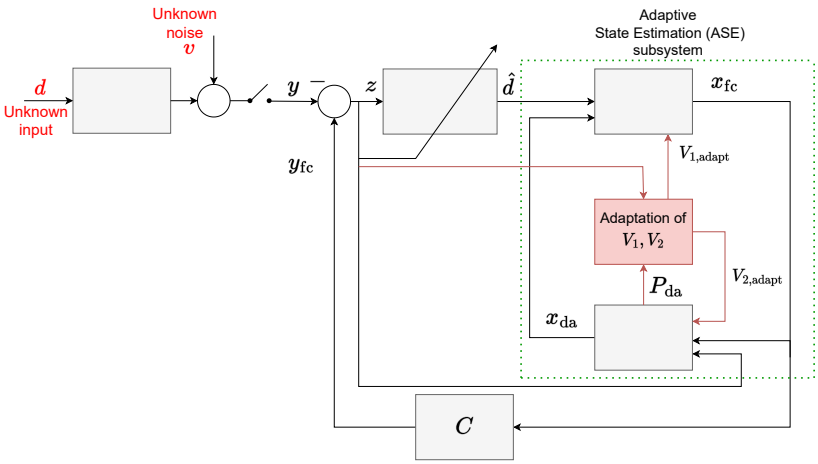

Figure 5.: Block diagram of AIE/ASE. In this version of AIE, both V_1 and V_2 are adapted. The state estimator is thus adaptive.

Algorithm 1 Adaptive Input Estimation/Adaptive State Estimation (AIE/ASE)

Choose $n_e \geq 1$, $n_{\mathrm{f}} \geq 1$, R_z, R_d, R_θ, η_{L}, η_{U}.
Set $x_{\mathrm{fc},0} = 0$, $P_{\mathrm{f},0} = 0_{n\times n}$, $K_{\mathrm{da},0} = 0_{n\times 1}$, $\hat{d}_0 = 0$, $\theta_{k_n-1} = 0_{l_\theta\times 1}$, $P_{k_n-1} = R_\theta^{-1}$, $V_{1,\mathrm{adapt},0} = 0_{n\times n}$, $V_{2,\mathrm{adapt},0} = 0$.
$k_n = \max(n_e, n_{\mathrm{f}})$; $\widetilde{R} = \mathrm{blockdiag}(R_z, R_d)$;
for $k = 0$ **to** $N-1$ **do**
(▷) *Residual*
    $y_{\mathrm{fc},k} = Cx_{\mathrm{fc},k}$;
    $z_k = y_{\mathrm{fc},k} - y_k$;
(▷) *Adaptive Input Estimation*
    if $k < k_n - 1$ **do**
        $\hat{d}_k = \hat{d}_0$;
    else do
        $\Phi_k = \begin{bmatrix}\hat{d}_{k-1} & \cdots & \hat{d}_{k-n_{\mathrm{e}}} & z_k & \cdots & z_{k-n_{\mathrm{e}}}\end{bmatrix}$;
        $\hat{d}_k = \Phi_k\theta_k$;
        $\overline{A}_{k-1} = A(I_n + K_{\mathrm{da},k-1}C)$;
        for $i = 2$ **to** n_{f} **do**
            $H_{i,k} = C\overline{A}_{k-1}\cdots\overline{A}_{k-(i-1)}B$;
        end for
        $\widetilde{H}_k = \begin{bmatrix}CB & H_{2,k} & \cdots & H_{n_{\mathrm{f}},k}\end{bmatrix}$;
        $\Phi_{\mathrm{f},k} = \widetilde{H}_k\begin{bmatrix}\Phi_{k-1}^{\mathrm{T}} & \cdots & \Phi_{k-n_{\mathrm{f}}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$;
        $\hat{d}_{\mathrm{f},k} = \widetilde{H}_k\begin{bmatrix}\hat{d}_{k-1}^{\mathrm{T}} & \cdots & \hat{d}_{k-n_{\mathrm{f}}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$;
        $\widetilde{\Phi}_k = \begin{bmatrix}\Phi_{\mathrm{f},k}^{\mathrm{T}} & \Phi_k^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$;
        $\widetilde{z}_k = \begin{bmatrix}(z_k - \hat{d}_{\mathrm{f},k})^{\mathrm{T}} & 0\end{bmatrix}^{\mathrm{T}}$;
        $\Gamma_k = (\widetilde{R}^{-1} + \widetilde{\Phi}_k P_k \widetilde{\Phi}_k^{\mathsf{T}})^{-1}$;
        $P_{k+1} = P_k - P_k\widetilde{\Phi}_k^{\mathsf{T}}\Gamma_k\widetilde{\Phi}_k P_k$;
        $\theta_{k+1} = \theta_k - P_k\widetilde{\Phi}_k^{\mathsf{T}}\Gamma_k(\widetilde{z}_k + \widetilde{\Phi}_k\theta_k)$;
    end if
(▷) *Adaptive State Estimation*
    if $k \geq 1$ **do**
        $\mathcal{J}_{\mathrm{f},k} = [\,]$; (▷) *empty set*
        $\widehat{S}_k = \texttt{variance}([z_0 \cdots z_k])$; (▷) *using* (49)
        for $i = 0$ **to** w **do** (▷) *Choose* $w > 0$
            $\eta_i = \eta_{\mathrm{L}} + i(\eta_{\mathrm{U}} - \eta_{\mathrm{L}})/w$;
            $\widetilde{P}_{\mathrm{f},k,i} = AP_{\mathrm{da},k-1}A^{\mathrm{T}} + \eta_i I_n$;
            $\widetilde{J}_{\mathrm{f},k,i} = \widehat{S}_k - C\widetilde{P}_{\mathrm{f},k,i}C^{\mathrm{T}}$;
            if $\widetilde{J}_{\mathrm{f},k,i} > 0$ **do**
                $\mathcal{J}_{\mathrm{f},k} = \texttt{append}(\mathcal{J}_{\mathrm{f},k},\ \widetilde{J}_{\mathrm{f},k,i})$;
            end if
        end for

Algorithm 1 Adaptive Input Estimation/Adaptive State Estimation (AIE/ASE) (continued)

if $\mathcal{J}_{\mathrm{f},k}$ is non-empty **do**
  $\widehat{J}_{\mathrm{f},k} = (\min \mathcal{J}_{\mathrm{f},k} + \max \mathcal{J}_{\mathrm{f},k})/2$;
  $V_{1,\mathrm{adapt},k} = \underset{\eta I_n}{\arg\min} \; |J_{\mathrm{f},k}(\eta I_n) - \widehat{J}_{\mathrm{f},k}|$;
  $V_{2,\mathrm{adapt},k} = J_{\mathrm{f},k}(V_{1,\mathrm{adapt},k})$;
else do
  $V_{1,\mathrm{adapt},k} = \underset{\eta I_n}{\arg\min} \; |J_{\mathrm{f},k}(\eta I_n)|$;
  $V_{2,\mathrm{adapt},k} = 0$;
end if
end if
(▷) *Kalman Filter Data − Assimilation*
$K_{\mathrm{da,k}} = -P_{\mathrm{f,k}} C^{\mathrm{T}} (C P_{\mathrm{f},k} C^{\mathrm{T}} + V_{2,\mathrm{adapt},k})^{-1}$;
$P_{\mathrm{da,k}} = (I_n + K_{\mathrm{da},k} C) P_{\mathrm{f},k}$;
$x_{\mathrm{da},k} = x_{\mathrm{fc},k} + K_{\mathrm{da,k}} z_k$;
(▷) *Kalman Filter Forecast*
$P_{\mathrm{f},k+1} = A P_{\mathrm{da},k} A^{\mathrm{T}} + V_{1,\mathrm{adapt},k}$;
$x_{\mathrm{fc},k+1} = A x_{\mathrm{da},k} + B \hat{d}_k$
end for

6. Numerical Differentiation of Two-Tone Harmonic Signal

In this section, a numerical example is given to compare the accuracy of the numerical differentiation algorithms discussed in the previous sections. We consider a two-tone harmonic signal, and we compare the accuracy (relative RMSE) of BD, HGO/1, SG, AIE/NSE, AIE/SSE, and AIE/ASE. For single and double differentiation, the parameters for HGO/1 and SG are given in Section 2.

Example 6.1. *Differentiation of a two-tone harmonic signal*

Consider the continuous-time signal $y(t) = \sin(20t) + \sin(30t)$, where t is time in seconds. The signal $y(t)$ is sampled with sample time $T_{\mathrm{s}} = 0.01$ sec. The measurements are assumed to be corrupted by noise, and thus the noisy sampled signal is given by $y_k = \sin(0.2k) + \sin(0.3k) + D_2 v_k$, where v_k is standard white noise.

Single Differentiation. For AIE/NSE, let $n_{\mathrm{e}} = 12$, $n_{\mathrm{f}} = 25$, $R_z = 1, R_d = 10^{-5}, R_\theta = 10^{-1} I_{25}$, $V_1 = 10^{-6}$, and $V_2 = 0.01$ for SNR 20 dB. For AIE/SSE, the parameters are the same as those of AIE/NSE, except that $V_{1,k}$ is adapted, where $\eta_{\mathrm{L}} = 10^{-6}$ and $\eta_{\mathrm{U}} = 10^2$ in Section 5.2. Similarly, for AIE/ASE, the parameters are the same as those of AIE/SSE except that $V_{2,k}$ is adapted as in Section 5.3.

Figure 6 compares the true first derivative with the estimates obtained from AIE/NSE, AIE/SSE, and AIE/ASE. Figure 7 shows that AIE/ASE has the best accuracy over the range of SNR. Figure 8 shows that the accuracy of AIE/ASE is close to the best accuracy of AIE/NSE.

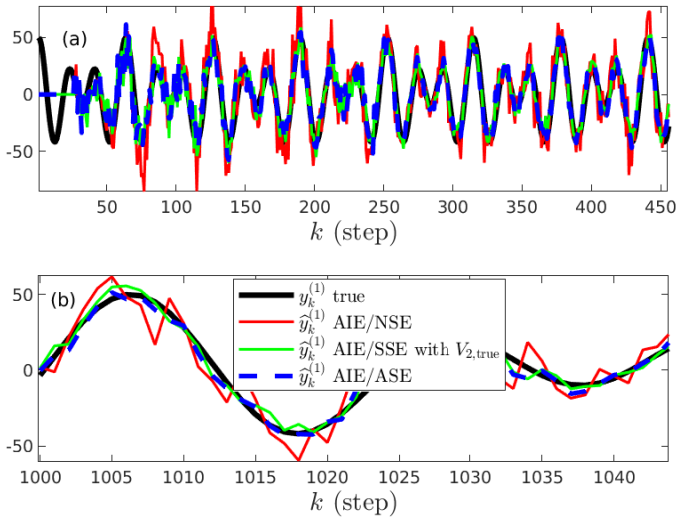

Figure 6.: *Example 6.1: Single differentiation of a sampled two-tone harmonic signal.* (a) The numerical derivatives estimated by AIE/NSE, AIE/SSE with $V_2 = V_{2,\mathrm{true}}$, and AIE/ASE follow the true first derivative $y^{(1)}$ after an initial transient. (b) Zoom of (a). At steady state, AIE/ASE is more accurate than both AIE/NSE and AIE/SSE with $V_2 = V_{2,\mathrm{true}}$. The SNR is 20 dB.

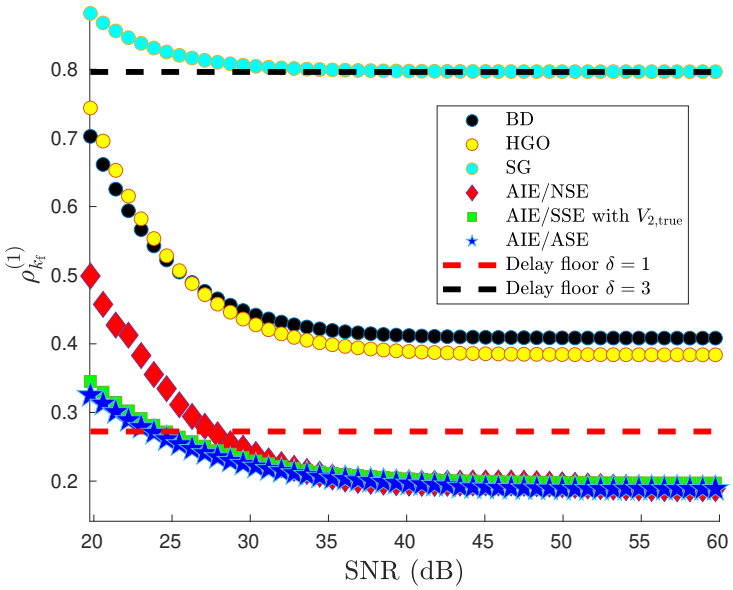

Figure 7.: *Example 6.1: Relative RMSE $\rho_{k_\mathrm{f}}^{(1)}$ of the estimate of the first derivative of a two-tone harmonic signal versus SNR.* AIE/ASE has the best accuracy over the range of SNR. Here $k_\mathrm{f} = 2000$ steps.

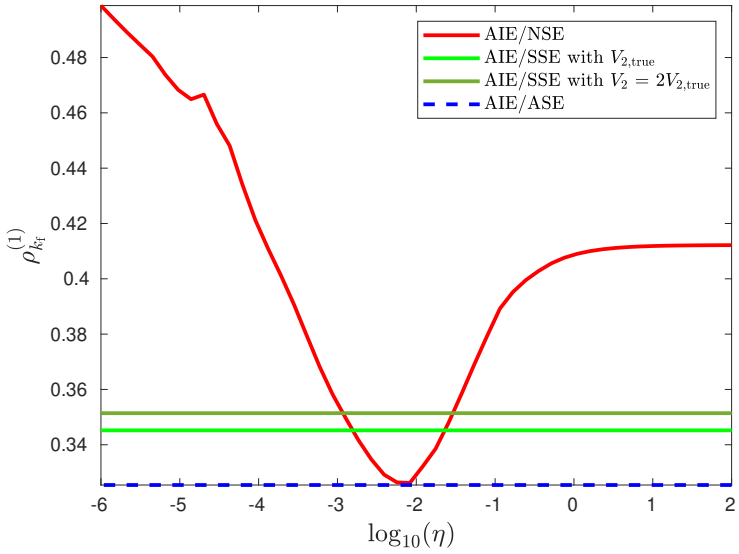

Figure 8.: *Example 6.1: Relative RMSE* $\rho^{(1)}_{k_\text{f}}$ *of the estimate of the first derivative of a two-tone harmonic signal versus* η*, such that* $V_1 = \eta$. AIE/SSE with $V_2 = V_{2,\text{true}}$ is more accurate than AIE/SSE with $V_2 = 2V_{2,\text{true}}$, which shows the effect of V_2 on accuracy. The accuracy of AIE/ASE is close to the best accuracy of AIE/NSE. The SNR is 20 dB, and $k_\text{f} = 2000$ steps.

Double Differentiation. For AIE/NSE, let $n_\text{e} = 12$, $n_\text{f} = 20$, $R_z = 1, R_d = 10^{-5}, R_\theta = 10^{-0.1} I_{25}$, $V_1 = 10^{-1} I_2$, and $V_2 = 0.0001$ for SNR 40 dB. For AIE/SSE, the parameters are the same as those of AIE/NSE, except that $V_{1,k}$ is adapted, where $\eta_\text{L} = 10^{-6}$ and $\eta_\text{U} = 1$ in Section 5.2. Similarly, for AIE/ASE, the parameters are the same as those of AIE/SSE except that $V_{2,k}$ is adapted as in Section 5.3.

Figure 9 compares the true second derivative with the estimates obtained from AIE/NSE, AIE/SSE with $V_2 = V_{2,\text{true}}$, and AIE/ASE. Figure 10 shows that AIE/ASE has the best accuracy over the range of SNR. Figure 11 shows that the accuracy of AIE/ASE is close to the best accuracy of AIE/NSE.

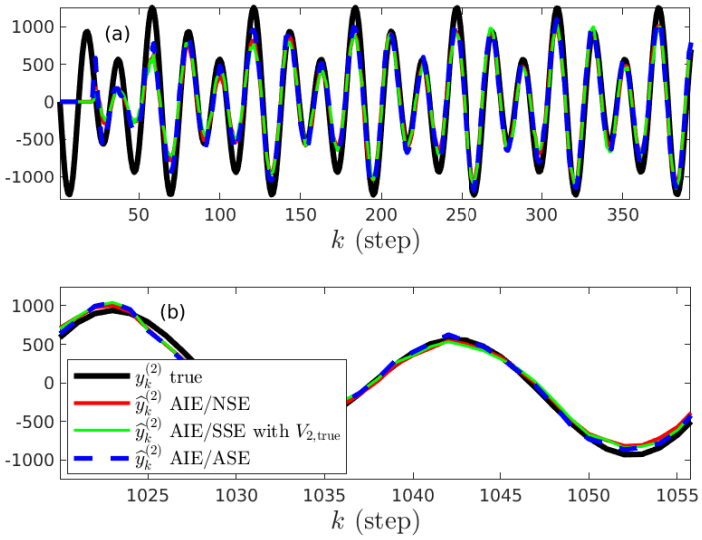

Figure 9.: *Example 6.1: Double differentiation of a sampled two-tone harmonic signal.* (a) The numerical derivatives estimated by AIE/NSE, AIE/SSE with $V_2 = V_{2,\mathrm{true}}$, and AIE/ASE follow the true second derivative $y^{(2)}$ after an initial transient. (b) Zoom of (a). At steady state, AIE/ASE is more accurate than AIE/SSE with $V_2 = V_{2,\mathrm{true}}$ and AIE/NSE. The SNR is 40 dB.

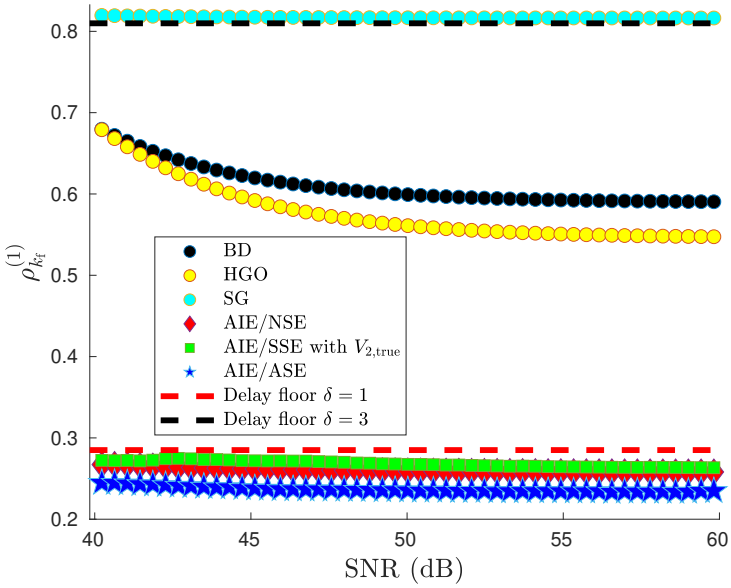

Figure 10.: *Example 6.1: Relative RMSE $\rho_{k_\mathrm{f}}^{(2)}$ of the estimate of the second derivative of a two-tone harmonic signal versus SNR.* AIE/ASE has the best accuracy over the range of SNR. Here $k_\mathrm{f} = 2000$ steps.

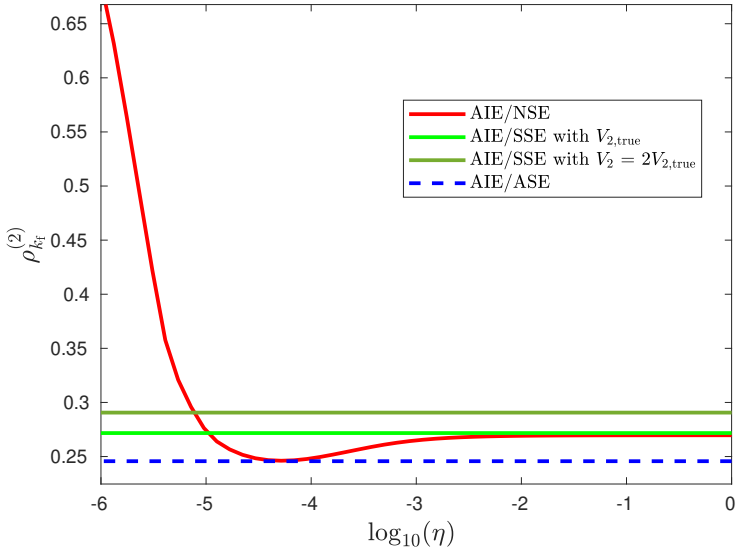

Figure 11.: *Example 6.1: Relative RMSE* $\rho^{(2)}_{k_f}$ *of the estimate of the second derivative of a two-tone harmonic signal versus* η*, such that* $V_1 = \eta I_2$. AIE/SSE with $V_2 = V_{2,\text{true}}$ is more accurate than AIE/SSE with $V_2 = 2V_{2,\text{true}}$, which shows the effect of V_2 on accuracy. The accuracy of AIE/ASE is close to the best accuracy of AIE/NSE. The SNR is 40 dB, and $k_f = 2000$ steps.

7. Application to Ground-Vehicle Kinematics

In this section, CarSim is used to simulate a scenario in which an oncoming vehicle (the white van in Figure 12) slides over to the opposing lane. The host vehicle (the blue van) performs an evasive maneuver to avoid a collision. Relative position data along the global y-axis (shown in Figure 12) is differentiated to estimate the relative velocity and acceleration along the same axis. Figure 13 shows the relative position trajectory of the vehicles on the x-y plane.

Figure 12.: *Collision-avoidance scenario in CarSim.* In this scenario, the oncoming vehicle (the white van) enters the opposite lane, and the host vehicle (the blue van) performs an evasive maneuver to avoid a collision.

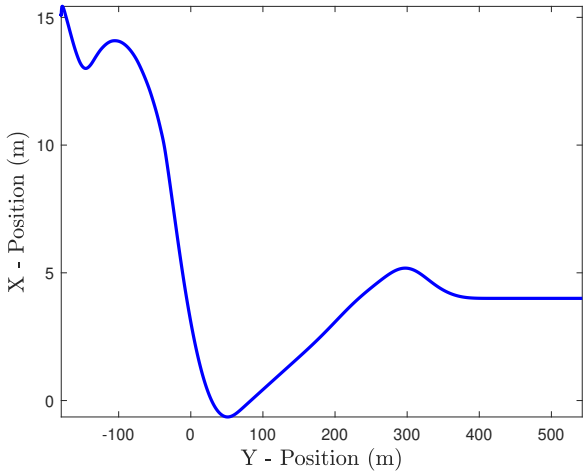

Figure 13.: *Collision-avoidance scenario in CarSim.* Relative position trajectory of the host and the target vehicles on x-y plane.

Example 7.1. *Differentiation of CarSim position data.*

Discrete-time position data generated by CarSim is corrupted with discrete-time, zero-mean, Gaussian white noise whose variance is chosen to vary the SNR.

Single Differentiation

For AIE/NSE, let $n_{\mathrm{e}} = 25$, $n_{\mathrm{f}} = 50$, $R_z = 1, R_d = 10^{-6}, R_\theta = 10^{-0.1} I_{51}$, $V_1 = 10^{-5}$, and $V_2 = 0.0049$ for SNR 40 dB. For AIE/SSE, the parameters are the same as those of AIE/NSE, except that $V_{1,k}$ is adapted, where $\eta_{\mathrm{L}} = 10^{-6}$ and $\eta_{\mathrm{U}} = 10^{-2}$ in Section 5.2. Similarly, for AIE/ASE, the parameters are the same as those of AIE/SSE except that $V_{2,k}$ is adapted as in Section 5.3.

Figure 14 compares the true first derivative with the estimates obtained from AIE/NSE, AIE/SSE with $V_2 = V_{2,\mathrm{true}}$, and AIE/ASE. Figure 15 shows that the accuracy of AIE/ASE is close to the best accuracy of AIE/NSE.

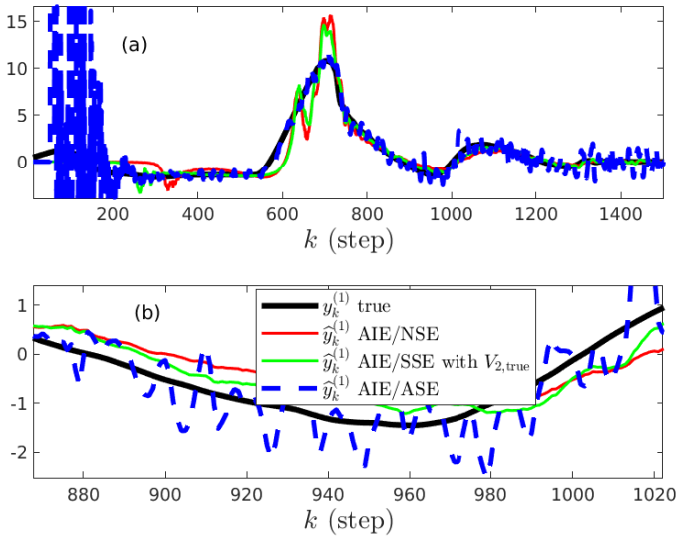

Figure 14.: *Example 7.1: Single differentiation of CarSim data.* (a) The numerical derivatives estimated by AIE/NSE, AIE/SSE with $V_2 = V_{2,\mathrm{true}}$, and AIE/ASE follow the true first derivative $y^{(1)}$ after an initial transient of 200 steps. (b) Zoom of (a). At steady state, AIE/ASE is more accurate than both AIE/NSE and AIE/SSE with $V_2 = V_{2,\mathrm{true}}$. The SNR is 40 dB.

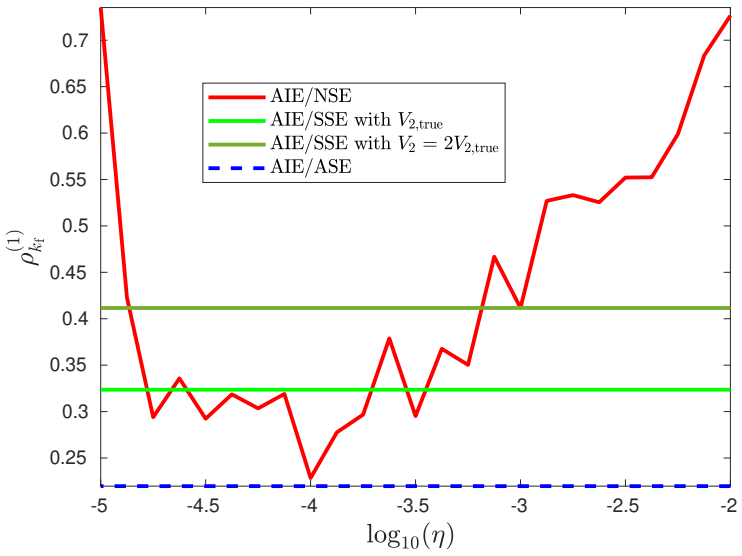

Figure 15.: *Example 7.1: Relative RMSE* $\rho^{(1)}_{k_\text{f}}$ *of the estimate of the first derivative of CarSim data versus* η*, such that* $V_1 = \eta$. AIE/SSE with $V_2 = V_{2,\text{true}}$ is more accurate than AIE/SSE with $V_2 = 2V_{2,\text{true}}$, which shows the effect of V_2 on accuracy. The accuracy of AIE/ASE is close to the best accuracy of AIE/NSE. The SNR is 40 dB, and $k_\text{f} = 1500$ steps.

Double Differentiation

For AIE/NSE, Let $n_\text{e} = 25$, $n_\text{f} = 21$, $R_z = 1$, $R_d = 10^{-5}$, $R_\theta = 10^{-8} I_{51}$, $V_1 = 10^{-3} I_2$, and $V_2 = 0.0049$ for SNR 40 dB. For AIE/SSE, the parameters are the same as those of AIE/NSE, except that $V_{1,k}$ is adapted, where $\eta_\text{L} = 10^{-3}$ and $\eta_\text{U} = 1$ in Section 5.2. Similarly, for AIE/ASE, the parameters are the same as those of AIE/SSE except that $V_{2,k}$ is adapted as in Section 5.3.

Figure 16 compares the true second derivative with the estimates obtained from AIE/NSE, AIE/SSE with $V_2 = V_{2,\text{true}}$, and AIE/ASE. Figure 17 shows that the accuracy of AIE/ASE is close to the best accuracy of AIE/NSE.

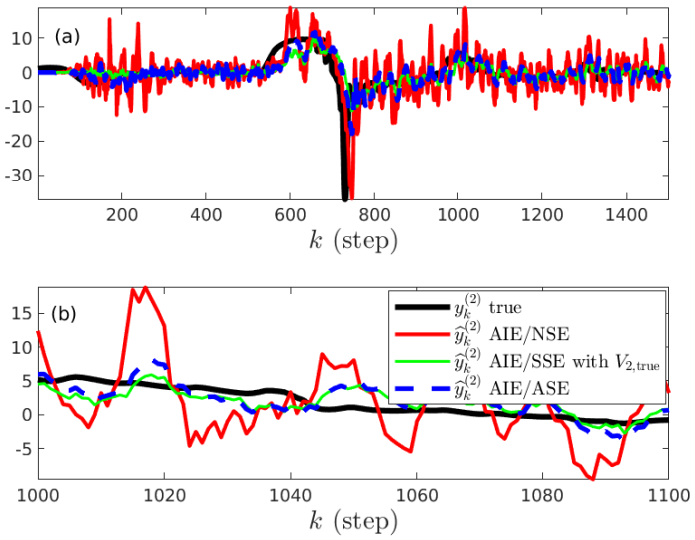

Figure 16.: *Example 7.1: Double differentiation of CarSim data.* (a) The numerical derivatives estimated by AIE/NSE, AIE/SSE with $V_2 = V_{2,\mathrm{true}}$, and AIE/ASE follow the true first derivative $y^{(2)}$ after an initial transient. (b) Zoom of (a). At steady state, AIE/ASE is more accurate than both AIE/NSE and AIE/SSE with $V_2 = V_{2,\mathrm{true}}$. The SNR is 40 dB.

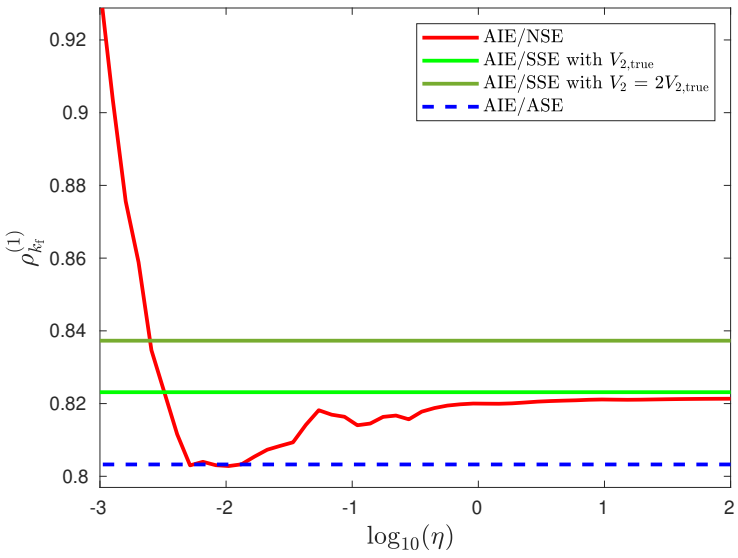

Figure 17.: *Example 7.1: Relative RMSE* $\rho_{k_\mathrm{f}}^{(2)}$ *of the estimate of the second derivative of CarSim data versus* η*, such that* $V_1 = \eta I_2$. AIE/SSE with $V_2 = V_{2,\mathrm{true}}$ is more accurate than AIE/SSE with $V_2 = 2V_{2,\mathrm{true}}$, which shows the effect of V_2 on accuracy. The accuracy of AIE/ASE is close to the best accuracy of AIE/NSE. The SNR is 40 dB, and $k_\mathrm{f} = 1500$ steps.

8. Conclusions

This paper presented the adaptive input and state estimation algorithm AIE/ASE for causal numerical differentiation. AIE/ASE uses the Kalman-filter residual to adapt the input-estimation subsystem and an empirical estimate of the estimation error to adapt the input-estimation and sensor-noise covariances. For dual-tone harmonic signals with various levels of sensor noise, the accuracy of AIE/ASE was compared to several conventional numerical differentiation methods. Finally, AIE/ASE was applied to simulated vehicle position data generated by CarSim.

Future work will focus on the following extensions. The minimization of (54) was performed by using a gridding procedure; more efficient optimization is possible. Furthermore, it is of interest to compare the accuracy of AIE/ASE to the adaptive sliding mode differentiator in Alwi and Edwards (2013). Finally, in practice, the spectrum of the measured signal and sensor noise may change abruptly. In these cases, it may be advantageous to replace the RLS update (41), (42) with RLS that uses variable-rate forgetting in Bruce, Goel, and Bernstein (2020); Mohseni and Bernstein (2022).

Acknowledgments

This research was supported by Ford and NSF grant CMMI 2031333.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Ahn, S., Choi, U. J., & Ramm, A. G. (2006). A Scheme for Stable Numerical Differentiation. *J. Comp. Appl. Math.*, *186*(2), 325–334.

Alenezi, B., Zhang, M., Hui, S., & Zak, S. H. (2021). Simultaneous Estimation of the State, Unknown Input, and Output Disturbance in Discrete-Time Linear Systems. *IEEE Trans. Autom. Contr.*, *66*(12), 6115–6122.

Almagbile, A., Wang, J., & Ding, W. (2010, 6). Evaluating the Performances of Adaptive Kalman Filter Methods in GPS/INS Integration. *J. Global Pos. Sys.*, *9*, 33-40.

Alwi, H., & Edwards, C. (2013). An Adaptive Sliding Mode Differentiator for Actuator Oscillatory Failure Case Reconstruction. *Automatica*, *49*(2), 642-651.

Ansari, A., & Bernstein, D. S. (2019). Input Estimation for Nonminimum-Phase Systems With Application to Acceleration Estimation for a Maneuvering Vehicle. *IEEE Trans. Contr. Sys. Tech.*, *27*(4), 1596–1607.

Astrom, K., & Hagglund, T. (2006). *Advanced PID Control.* ISA.

Bogler, P. (1987). Tracking a Maneuvering Target Using Input Estimation. *IEEE Trans. Aero. Elec. Sys.*, *AES-23*(3), 298-310.

Bruce, A., Goel, A., & Bernstein, D. S. (2020). Convergence and Consistency of Recursive Least Squares with Variable-Rate Forgetting. *Automatica*, *119*, 109052.

Cullum, J. (1971). Numerical Differentiation and Regularization. *SIAM J. Num. Anal.*, *8*, 254–265.

Dabroom, A. M., & Khalil, H. K. (1999). Discrete-time Implementation of High-Gain Observers for Numerical Differentiation. *Int. J. Contr.*, *72*(17), 1523–1537.

Davis, P. J., & Rabinowitz, P. (1984). *Methods of Numerical Integration* (second ed.). Dover.

Da-yan Liu, O. G., & Perruquetti, W. (2011). Differentiation by Integration with Jacobi Polynomials. *Jour. of Comp. and Appl. Math.*, *235*(9), 3015-3032.

Fang, H., Shi, Y., & Yi, J. (2011). On Stable Simultaneous Input and State Estimation for Discrete-Time Linear Systems. *Int. J. Adapt. Contr. Sig. Proc.*, *25*(8), 671–686.

Farrell, J. A. (2008). *Aided navigation: Gps with high rate sensors.* McGraw-Hill.

Gillijns, S., & De Moor, B. (2007). Unbiased Minimum-Variance Input and State Estimation for Linear Discrete-Time Systems. *Automatica*, *43*(1), 111–116.

Grewel, M. S., Andrews, A. P., & Bartone, C. G. (2020). *Global Navigation Satellite Systems, Inertial Navigation, and Integration* (fourth ed.). Wiley.

Haimovich, H., Seeber, R., Aldana-López, R., & Gómez-Gutiérrez, D. (2022). Differentiator for Noisy Sampled Signals With Best Worst-Case Accuracy. *IEEE Contr. Sys. Lett.*, *6*, 938-943.

Hamming, R. W. (1973). *Numerical Methods of Scientists and Engineers* (second ed.). Dover.

Hide, C., Moore, T., & Smith, M. (2003, 1). Adaptive Kalman Filtering for Low-Cost INS/GPS. *J. Nav.*, *56*, 143-152.

H. Khaloozadeh, A. K. (2009). Modified Input Estimation Technique for Tracking Manoeuvring Targets. *IET Radar Sonar Nav.*, *3*, 30-41.

Hsieh, C.-S. (2017). Unbiased Minimum-Variance Input and State Estimation for Systems with Unknown Inputs: A System Reformation Approach. *Automatica*, *84*, 236–240.

Ibrir, S., & Diop, S. (2004). A Numerical Procedure for Filtering and Efficient High-Order Signal Differnetiation. *Int. J. Appl. Math. Computer Sci.*(14.2), 201–208.

Islam, S. A. U., & Bernstein, D. S. (2019). Recursive Least Squares for Real-Time Implementation. *IEEE Contr. Sys. Mag.*, *39*(3), 82–85.

Jauberteau, F., & Jauberteau, J. (2009). Numerical Differentiation with Noisy Signal. *Appl. Math. Comp.*, *215*, 2283–2297.

Jia, Z., Balasuriya, A., & Challa, S. (2008). Autonomous Vehicles Navigation with Visual Target Tracking: Technical Approaches. *Algorithms*, *1*(2), 153–182.

Kalata, P. R. (1984). The Tracking Index: A Generalized Parameter for α-β and α-β-γ Target Trackers. *IEEE Trans. Aero. Elec. Sys.*, *AES-20*(2), 174-182.

Knowles, I., & Renka, R. J. (2014). Methods for Numerical Differentiation of Noisy Data. *Electron. J. Diff. Eqn.*, *21*, 235–246.

Lee, H., & Tahk, M.-J. (1999). Generalized Input-Estimation Technique for Tracking Maneuvering Targets. *IEEE Trans. Aero. Elec. Sys.*, *35*(4), 1388-1402.

Levant, A. (1998). Robust Exact Differentiation via Sliding Mode Technique. *Automatica*, *34*(3), 379-384.

Levant, A. (2003). Higher-Order Sliding Modes, Differentiation and Output-Feedback Control. *Int. J. Contr.*, *76*(9-10), 924–941.

Listmann, K. D., & Zhao, Z. (2013). A Comparison of Methods for Higher-Order Numerical Differentiation. In *Proc. eur. contr. conf.* (pp. 3676–3681).

López-Caamal, F., & Moreno, J. A. (2019). Generalised Multivariable Supertwisting Algorithm. *Int. J. Robust Nonlinear Contr.*, *29*(3), 634–660.

Mboup, M., Join, C., & Fliess, M. (2009). Numerical Differentiation with Annihilators in Noisy Environment. *Num. Algor.*, *50*, 439–467.

Mehra, R. K. (1972). Approaches to Adaptive Filtering. *IEEE Trans. Autom. Contr.*, *17*, 693-698.

Moghe, R., Zanetti, R., & Akella, M. R. (2019). Adaptive Kalman Filter for Detectable Linear Time-Invariant Systems. *J. Guid. Contr. Dyn.*, *42*(10), 2197–2205.

Mohamed, A. H., & Schwarz, K. P. (1999). Adaptive Kalman Filtering for INS/GPS. *J. Geodesy*, 193-203.

Mohseni, N., & Bernstein, D. S. (2022). Recursive Least Squares with Variable-Rate Forgetting Based on the F-Test. In *Proc. amer. contr. conf.* (pp. 3937–3942).

Mojallizadeh, M. R., Brogliato, B., & Acary, V. (2021). Discrete-Time Differentiators: Design and Comparative Analysis. *Int. J. Robust Nonlinear Contr.*, *31*(16), 7679–7723.

Mook, D. J., & Junkins, J. L. (1988). Minimum Model Error Estimation for Poorly Modeled

Dynamic Systems. *J. Guid. Contr. Dyn.*, *11*(3), 256–261.
Naderi, E., & Khorasani, K. (2019). Unbiased Inversion-Based Fault Estimation of Systems with Non-Minimum Phase Fault-to-Output Dynamics. *IET Contr. Theory Appl.*, *13*(11), 1629–1638.
Nieuwstadt, M. V., Rathinam, M., & Murray, R. M. (1998). Differential flatness and absolute equivalence of nonlinear control systems. *SIAM J. Contr. Optim.*, *36*(4), 1225–1239.
Orjuela, R., Marx, B., Ragot, J., & Maquin, D. (2009). On the Simultaneous State and Unknown Input Estimation of Complex Systems via a Multiple Model Strategy. *IET Contr. Theory Appl.*, *3*(7), 877–890.
Peng Li, G. F., Gilberto Pin, & Parisini, T. (2018). Non-Asymptotic Numerical Differentiation: A Kernel-Based Approach. *Int. J. Contr.*, *91*(9), 2090–2099.
Polyakov, A., Efimov, D., & Perruquetti, W. (2014). Homogeneous Differentiator Design Using Implicit Lyapunov Function Method. In *Proc. euro. contr. conf.* (p. 288-293).
Rana, M. M., Halim, N., Rahamna, M. M., & Abdelhadi, A. (2020). Position and Velocity Estimations of 2D-Moving Object Using Kalman Filter: Literature Review. In *Proc. int. conf. adv. comm. tech.* (p. 541-544).
Reichhartinger, M., & Spurgeon, S. (2018). An Arbitrary-Order Differentiator Design Paradigm with Adaptive Gains. *Int. J. Contr.*, *91*(9), 2028–2042.
Savitzky, A., & Golay, M. J. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chemistry*, *36*(8), 1627–1639.
Schafer, R. W. (2011). What is a Savitzky-Golay Filter? *IEEE Sig. Proc. Mag.*, *28*(4), 111-117.
Shi, Y., Han, C., & Liang, Y. (2009). Adaptive UKF for Target Tracking with Unknown Process Noise Statistics. In *Proc. int. conf. inf. fusion* (pp. 1815–1820).
Staggs, J. E. J. (2005). Savitzky–Golay Smoothing and Numerical Differentiation of Cone Calorimeter Mass Data. *Fire Safety J.*, *40*(6), 493-505.
Stickel, J. (2010). Data Smoothing and Numerical Differentiation by a Regularization Method. *Comp. Chem. Eng.*, *34*, 467–475.
Van Breugel, F., Kutz, J. N., & Brunton, B. W. (2020). Numerical Differentiation of Noisy Data: A Unifying Multi-Objective Optimization Framework. *IEEE Access*, *8*, 196865–196877.
Verma, S., Sanjeevini, S., Sumer, E. D., Girard, A., & Bernstein, D. S. (2022). On the Accuracy of Numerical Differentiation Using High-Gain Observers and Adaptive Input Estimation. In *Proc. amer. contr. conf.* (p. 4068-4073).
Vilanova, R., & Visioli, A. (2012). *PID Control in the Third Millennium: Lessons Learned and New Approaches*. Springer.
Yaesh, I., & Shaked, U. (2008). Simplified Adaptive Estimation. *Sys. Contr. Lett.*, *57*, 49–55.
Yong, S. Z., Zhu, M., & Frazzoli, E. (2016). A Unified Filter for Simultaneous Input and State Estimation of Linear Discrete-Time Stochastic Systems. *Automatica*, *63*, 321–329.
Zhang, L., Sidoti, D., Bienkowski, A., Pattipati, K. R., Bar-Shalom, Y., & Kleinman, D. L. (2020). On the Identification of Noise Covariances and Adaptive Kalman Filtering: A New Look at a 50 Year-Old Problem. *IEEE Access*, *8*, 59362–59388.