IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 17, 2021, accepted June 24, 2021, date of publication July 5, 2021, date of current version July 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094963

Multi-Pass Sequential Mini-Batch Stochastic Gradient Descent Algorithms for Noise Covariance Estimation in Adaptive Kalman Filtering

HEE-SEUNG KIM, (Member, IEEE), LINGYI ZHANG, (Member, IEEE), ADAM BIENKOWSKI, (Member, IEEE), AND KRISHNA R. PATTIPATI, (Life Fellow, IEEE)
Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA

Corresponding author: Hee-Seung Kim (hee-seung.kim@uconn.edu)

This work was supported in part by the U.S. Office of Naval Research (ONR); in part by the U.S. Naval Research Laboratory (NRL) under Grant N00014-18-1-1238, Grant N00014-21-1-2187, and Grant N00173-16-1-G905; and in part by the NASA's Space Technology Research Grants Program under Grant 80NSSC19K1076.

ABSTRACT Estimation of unknown noise covariances in a Kalman filter is a problem of significant practical interest in a wide array of applications. Although this problem has a long history, reliable algorithms for their estimation were scant, and necessary and sufficient conditions for identifiability of the covariances were in dispute until recently. Necessary and sufficient conditions for covariance estimation and a batch estimation algorithm were presented in our previous study. This paper presents stochastic gradient descent algorithms for noise covariance estimation in adaptive Kalman filters that are an order of magnitude faster than the batch method for similar or better root mean square error. More significantly, these algorithms are applicable to non-stationary systems where the noise covariances can occasionally jump up or down by an unknown magnitude. The computational efficiency of the new algorithms stems from adaptive thresholds for convergence, recursive fading memory estimation of the sample cross-correlations of the innovations, and accelerated stochastic gradient descent algorithms. The comparative evaluation of the proposed methods on a number of test cases demonstrates their computational efficiency and accuracy.

INDEX TERMS Adaptive Kalman filtering, noise covariance estimation, Adam, RMS prop, bold-driver, stochastic gradient descent, sequential, fading memory.

I. INTRODUCTION

The Kalman filter (KF) [8] is the minimum mean square error (MMSE) state estimator for discrete-time linear dynamic systems under Gaussian white noise with known mean and covariance parameters. It is also the best *linear* estimation algorithm when the noise processes are non-Gaussian with known first and second-order statistics, i.e., mean and covariance. It has found successful applications in numerous fields, such as navigation, weather forecasting, signal processing, econometrics and structural health monitoring, to name a few [2]. However, in many practical situations, including the ones mentioned above, the statistics of the noise covariances are often unknown or only partially known.

The associate editor coordinating the review of this manuscript and approving it for publication was Larbi Boubchir.

Exploiting the whiteness property of the innovations in an optimal Kalman filter [4], [13]–[15], previous work [15], [21] has derived necessary and sufficient conditions for estimating the unknown process noise and measurement noise covariances from data. In [21], we have presented a *batch* optimization algorithm to estimate the unknown covariances to minimize the sum of the normalized temporal cross-correlations of the innovations. In this paper, we explore enhanced covariance estimation methods based on *sequential* mini-batch stochastic gradient descent (SGD) algorithms with adaptive step sizes and iteration-dependent convergence thresholds.

A. PRIOR WORK

The key to process noise and measurement noise covariance estimation is an expression for the covariance of the

state estimation error and of the innovations of any stable suboptimal filter as a function of noise covariances. This expression serves as a fundamental building block for the correlation-based methods for noise covariance estimation. Pioneering contributions using this approach were made by [4], [13]–[15]. Relatively recently, Särkkä and Nummenmaa [17] proposed a variational Bayesian method for the joint recursive estimation of the dynamic state and measurement noise parameters in linear state space models. This method is based on forming a separable variational approximation to the joint posterior distribution of states and noise parameters at each time step separately. This method does not consider changes in process noise. In general, the variational methods require tuning parameters to converge to the correct noise covariances and often converge to a local minimum.

Recently, Zhang *et al.* [21] proposed a new approach to estimate the process noise and measurement noise covariances of a system using the entire time series of innovation samples. For estimating the unknown covariances, Zhang *et al.* [21] proved that the identifiability conditions are related to the rank of a matrix involving the auto and cross-covariances of a weighted sum of innovations, where the weights are the coefficients of the minimal polynomial of the closed-loop system transition matrix of any stable, not necessarily optimal, Kalman filter. An optimization algorithm was developed to estimate the steady-state Kalman filter gain, the unknown noise covariance matrices, as well as the state prediction (and updated) error covariance matrix. A drawback of this approach is that it is a *batch* optimization method requiring the entire observation sequence to compute the cost and the gradient, is computationally expensive, and is applicable only to stationary systems. The present paper seeks to overcome these limitations.

B. CONTRIBUTION AND ORGANIZATION

In this paper, we develop sequential mini-batch estimation methods with adaptive step size rules to improve the computational efficiency of the algorithm in [21]. Our approach replaces the batch innovation covariance estimates with sequential fading memory mini-batch estimates when updating the Kalman filter gain. We also enhance the algorithm by investigating the performance of several accelerated stochastic gradient descent (SGD) algorithms, viz., Bold driver [3], [21], Constant, Subgradient [6], RMSProp [19], Adam [11], Adadelta [20] step-size selection methods, and enhance their computational efficiency via iteration-dependent dynamic thresholds for convergence. The SGD methods, together with change detection algorithms, enable the estimation of noise covariances in non-stationary systems in a decision-directed manner. The jumps are assumed to occur occasionally and after the filter has reached a steady-state. The validation of the proposed method on several test cases from the literature and on non-stationary systems demonstrates its efficacy.

The paper is organized as follows. In Section II, we provide an overview of the batch gradient descent method for estimating the unknown noise covariances. Then, in Section III, we present the sequential mini-batch gradient descent method for optimizing the Kalman gain and consequently estimating the measurement and process noise covariances. We also discuss how we enhanced our approach using a fading memory filter-based innovation correlation estimation, SGD update of the Kalman gain, dynamic thresholding for algorithm convergence, and accelerated step size rules for the SGD update. Section IV provides evidence that the computational performance of the proposed methods is substantially better than the batch estimation algorithm via numerical examples and demonstrates their applicability to non-stationary systems where the noise covariances abruptly jump by an unknown, but finite, magnitude occasionally. Lastly, we conclude the paper and discuss our future work in Section V.

II. BATCH GRADIENT DESCENT METHOD FOR ESTIMATING Q AND R

Consider a linear discrete-time stochastic dynamic system

$$x(k+1) = Fx(k) + \Gamma v(k) \tag{1}$$
$$z(k) = Hx(k) + w(k) \tag{2}$$

where $x(k)$ is the state vector, $z(k)$ is the measurement vector, F and H are the state transition matrix and the measurement matrix, respectively, and Γ is the noise gain matrix. Here, the process noise, $v(k)$ and the measurement noise, $w(k)$ are assumed to be zero-mean white Gaussian noise processes with *unknown* process noise covariance Q and *unknown* measurement noise covariance R, respectively. We allow for Q and R to change occasionally (see Section IV). These two noise covariances and the initial state are assumed to be mutually independent in time.

When Q and R are known, the Kalman filter involves consecutive processes of prediction and update given by

$$\hat{x}(k+1|k) = F\hat{x}(k|k) \tag{3}$$
$$\nu(k+1) = z(k+1) - H\hat{x}(k+1|k) \tag{4}$$
$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)\nu(k+1) \tag{5}$$
$$P(k+1|k) = FP(k|k)F' + \Gamma Q\Gamma' \tag{6}$$
$$S(k+1) = HP(k+1|k)H' + R \tag{7}$$
$$W(k+1) = P(k+1|k)H'S(k+1)^{-1} \tag{8}$$
$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)' \tag{9}$$

The Kalman filter predicts the next state estimate at time index $(k+1)$, given the observations up to time index k in (3) and the concomitant predicted state estimation error covariance in (6), using system dynamics, the updated state error covariance $P(k|k)$ at time index k and Q. The updated state estimate at time $(k+1)$ in (5) incorporates the measurement at time $(k+1)$ via the Kalman gain matrix in (8), which depends on the innovation covariance $S(k+1)$ (which in turn depends on R) and the predicted state error covariance $P(k+1|k)$. The updated state error covariance $P(k+1|k+1)$ is computed via (9).

A. IDENTIFIABILITY CONDITIONS FOR ESTIMATING Q AND R

When Q and R are unknown, consider the innovations corresponding to a stable, suboptimal closed-loop filter matrix $\bar{F} = F(I_{n_x} - WH)$ given by [18], [21]

$$\begin{aligned} v(k) &= H\bar{F}^m\tilde{x}(k-m|k-m-1) \\ &\quad + \Big\{H\sum_{j=0}^{m-1}\bar{F}^{m-1-j}\Big[\Gamma v(k-m+j) \\ &\quad - FWw(k-m+j)\Big]\Big\} + w(k) \end{aligned} \tag{10}$$

where $\tilde{x}(k-m|k-m-1) = x(k-m) - \hat{x}(k-m|k-m-1)$ is the predicted error at time $(k-m)$. Given the innovation sequence (10), let us define a weighted sum of innovations, $\xi(k) = \sum_{i=0}^{m} a_i v(k-i)$, where the weights are the coefficients of the minimal polynomial of the closed-loop filter matrix $\bar{F}$, $\sum_{i=0}^{m} a_i\bar{F}^{m-i} = 0, a_0 = 1$. It is easy to see that $\xi(k)$ is the sum of two moving average processes driven by the process noise and measurement noise, respectively, given by [21]

$$\xi(k) = \sum_{l=1}^{m} B_l v(k-l) + \sum_{l=0}^{m} G_l w(k-l) \tag{11}$$

Here, B_l and G_l are given by

$$B_l = H\Big(\sum_{i=0}^{l-1} a_i\bar{F}^{l-i-1}\Big)\Gamma \tag{12}$$

$$G_l = \Big[a_l I_{n_z} - H\Big(\sum_{i=0}^{l-1} a_i\bar{F}^{l-i-1}\Big)FW\Big] \tag{13}$$

$$G_0 = I_{n_z} \tag{14}$$

Then, if we define the cross-covariance between $\xi(k)$ and $\xi(k-j)$ as L_j, we obtain

$$L_j = E[\xi(k)\xi(k-j)'] = \sum_{i=j+1}^{m} B_i Q B'_{i-j} + \sum_{i=j}^{m} G_i R G'_{i-j} \tag{15}$$

The noise covariance matrices $Q = [q_{ij}]$ of dimension $n_v \times n_v$ and $R = [r_{ij}]$ of dimension $n_z \times n_z$ are positive definite and symmetric. By converting noise covariance matrices and the L_j matrices as vectors, Zhang *et al*. [21] show that they are related by the noise covariance identifiability matrix I as in (16).

$$I\begin{bmatrix}\mathrm{vec}(Q)\\ \mathrm{vec}(R)\end{bmatrix} = \begin{bmatrix}\mathrm{L}_0\\ \mathrm{L}_1\\ \vdots\\ \mathrm{L}_m\end{bmatrix} \tag{16}$$

As shown in [21], if matrix I has full column rank, then the unknown noise covariance matrices, Q and R, are identifiable.

B. OBJECTIVE FUNCTION AND THE GRADIENT

Innovations of an optimal Kalman filter are white, meaning that they are uncorrelated over time [4], [13]–[15]. In contrast, the innovation sequence $\{v(k)\}_{k=1}^{N_k}$ will be correlated if the Kalman filter gain is not optimal. The correlation methods exploit this non-whiteness property of the innovations of a suboptimal Kalman filter by minimizing a measure of the cross-correlations of the innovations over a lag window of length up to $M \geq n_x$ for estimating the optimal Kalman filter gain W from the computed correlated innovations. The M sample covariance matrices $\hat{C}(i)$ for each lag $i = 0, 1, 2, .., M-1$ are given by (17).

$$\hat{C}(i) = \frac{1}{N_k - M}\sum_{j=1}^{N_k - M} v(j)v(j+i)', \ i = 0, 1, 2, \ldots, M-1 \tag{17}$$

On the other hand, the ensemble cross-correlations of the innovations of a steady-state suboptimal Kalman filter,$C(i), i = 1, 2, ..$ are related to the closed-loop filter matrix $\bar{F}$, the matrix F, the measurement matrix H, the predicted covariance matrix $\bar{P}$, filter gain W and the innovation covariance, $C(0)$ via [4], [13]

$$C(i) = E[v(k)v(k-i)'] = H\bar{F}^{i-1}F[\bar{P}H' - WC(0)] \tag{18}$$

The objective function J formulated in [21] involves minimization of the sum of normalized $C(i)$ with respect to the corresponding diagonal elements of $C(0)$ for $i > 0$. Formally, we can define the objective function J to be minimized with respect to W as

$$\begin{aligned} J = \frac{1}{2}tr\Big\{\sum_{i=1}^{M-1}\Big[diag(C(0))\Big]^{-\frac{1}{2}}C(i)' \\ \times\Big[diag(C(0))\Big]^{-1}C(i)\Big[diag(C(0))\Big]^{-\frac{1}{2}}\Big\} \end{aligned} \tag{19}$$

where $diag(C)$ denotes the diagonal matrix of C or equivalently the Hadamard product of an identity matrix with C. We can rewrite the objective function by substituting (19) into (18) as

$$J = \frac{1}{2}tr\Big\{\sum_{i=1}^{M-1}\Theta(i)X\varphi X'\Big\} \tag{20}$$

where

$$\Theta(i) = [H\bar{F}^{i-1}F]'\varphi[H\bar{F}^{i-1}F] \tag{21}$$

$$X = \bar{P}H' - WC(0) \tag{22}$$

$$\varphi = [diag(C(0))]^{-1} \tag{23}$$

The gradient of objective function $\nabla_W J$ can be computed as [21]

$$\begin{aligned} \nabla_W J = -\sum_{i=1}^{M-1}[H\bar{F}^{i-1}F]'\varphi C(i)\varphi C(0) - F'ZFX \\ - \sum_{l=0}^{i-2}[C(l+1)\varphi C(i)'\varphi H\bar{F}^{i-l-2}]' \end{aligned} \tag{24}$$

The Z term in (24) is computed by the Lyapunov equation; it is often small and can be neglected for computational efficiency.

$$Z = \bar{F}' Z \bar{F} + \frac{1}{2}\sum_{i=1}^{M-1}(H\bar{F}^{i-1}F)'\varphi C(i)\varphi H + ((H\bar{F}^{i-1}F)'\varphi C(i)\varphi H)' \quad (25)$$

In computing the objective function and the gradient, we replace $C(i)$ by their sample estimates, $\hat{C}(i)$. Evidently, the noise covariance estimation is a stochastic optimization problem.

C. ESTIMATION OF Q AND R

1) ESTIMATION OF R

Let us define $\mu(k), k = 1, 2, \cdots, N_k$, as the post-fit residual sequence of the Kalman filter, which is related to the innovations $\nu(k), k = 1, 2, \cdots, N_k$ via

$$\mu(k) = z(k) - H\hat{x}(k|k) = (I_{n_z} - HW)\nu(k) \quad (26)$$

From the joint covariance of the innovation sequence $\nu(k)$ and the post-fit residual sequence $\mu(k)$, and the Schur determinant identity [5], [7], one can show that [21]

$$G = E[\mu(k)\mu(k)'] = RS^{-1}R \quad (27)$$

where S is the innovation covariance. Knowing the sampled estimates of G and S, the measurement noise covariance R is estimated. Because (27) can be interpreted as a simultaneous diagonalization problem in linear algebra [7] or as a continuous-time algebraic Riccati equation, the measurement covariance R can be estimated by solving the simultaneous diagonalization problem via Cholesky decomposition and Eigen decomposition, or by solving a continuous-time Riccati equation as in [1], [21].

2) ESTIMATION OF Q

Given the estimated R, we can compute the process noise covariance Q and the steady-state updated covariance P. This requires an iterative process because Q and P are coupled in the general case, Wiener process being an exception where an explicit non-iterative solution $Q = WSW'$ is possible [21]. Letting t and l denote the iteration indices starting with $t = 0$ and $l = 0$, and using an initial $Q^{(0)} = WSW'$, we initialize the steady-state updated covariance matrix P as the solution of the Lyapunov equation in (28)

$$P^{(0)} = \tilde{F}P^{(0)}\tilde{F}' + WRW' + (I_{n_x} - WH)\Gamma Q^{(t)}\Gamma'(I_{n_x} - WH) \quad (28)$$

where $\tilde{F} = (I_{n_x} - WH)F$. We iteratively update P as in (29) until convergence

$$P^{(l+1)} = \left[\left(FP^{(l)}F' + \Gamma Q^{(t)}\Gamma'\right)^{-1} + H'R^{-1}H\right]^{-1} \quad (29)$$

Given the converged P, Q will be updated in the t-loop until the estimate of Q converges. Note that λ_Q is a regularization parameter used for ill-conditioned estimation problems.

$$Q^{(t+1)} = \Gamma^{\dagger}\left[\left(P + WSW' - FPF'\right)^{(t+1)} + \lambda_Q I_{n_x}\right](\Gamma')^{\dagger} \quad (30)$$

D. BATCH ESTIMATION ALGORITHM

A sample pseudocode of the batch estimation algorithm is shown below. In the outer-loop, the estimated Q and R of the previous iteration are used to refine the estimates. Typically, less than 10 outer-loop iterations are needed.

Algorithm 1 Pseudocode of Batch Gradient Descent-Based on Adaptive Kalman Filtering Algorithm

```
input: W_0, α ▷ W_0: initial gain, for example, by solving a steady-state Riccati equation from an initial R^(0) and Q^(0) α: step size
for t = 1 to N_t do ▷ N_t: Max. Num. outer-loop iter.
  for l = 1 to N_l do ▷ N_l: Max. Num. inner-loop iter.
    for k = 1 to N_k do ▷ N_k: Num. samples
      compute the innovation correlations, ν(k)
    end for
    compute Ĉ
    compute objective function J
    compute gradient ∇_W J
    update the step size α^(l)
    update gain W^(l+1) = W^(l) − α^(l) ∇_W J(W^(l))
    check the convergence every inner-loop iteration
  end for
  check the convergence every outer-loop iteration
  update R^(t+1) and Q^(t+1)
end for
```

III. SEQUENTIAL MINI-BATCH GRADIENT DESCENT METHOD FOR ESTIMATING Q AND R

In this section, we provide an overview of the sequential mini-batch SGD method for estimating the unknown Q and R. We introduce five enhancements to the batch estimation algorithm: recursive fading memory estimation of the sample cross-correlations of innovations, mini-batch SGD, dynamic thresholds for inner-loop termination, accelerated SGD, and simplified gradient computation by neglecting the Z term in (24). The recursive nature of our algorithm makes it amenable for the estimation of noise covariances in non-stationary systems. The non-stationarity is assumed to arise from abrupt changes of unknown magnitude in noise covariances occasionally and after the filter reaches the steady-state.

A. RECURSIVE FADING MEMORY

We compute the sample correlation matrix $\hat{C}^k_{seq}(i)$ at sample k for time lag i as a weighted combination of the correlation matrix $\hat{C}^{k-1}_{seq}(i)$ at the previous sample $(k-1)$ for time lag i and the samples of innovations $\nu(k-i)$ and $\nu(k)$. The tuning parameter λ, a positive constant between 0 and 1, is the weight associated with the previous sample correlation matrix. The tuning parameter λ assumes that innovations at the current

time do not depend on infinite past and serves as a means to model non-stationary noise processes in the current context. The current M sample correlation matrices are used as the initial values for the next pairs of samples.

$$\hat{C}^{k}_{seq}(i) = (1-\lambda)v(k-i)v(k)' + \lambda \hat{C}^{k-1}_{seq}(i), \quad (31)$$

$$\hat{C}^{0}_{seq}(i) = 0, i = 0, 1, 2, \ldots M-1;\ k = M, \ldots, N_k \quad (32)$$

B. UPDATING GAIN W SEQUENTIALLY

Let B be the mini-batch size and let $K = N_k/B$ be the number of min-batches (we assume that N_k is divisible by B for simplicity). While the mini-batch gradient descent sequentially updates the M sample covariance matrices at every input sample, we update the Kalman filter gain W when the sample index k is divisible by the size of mini-batch B using the gradient of the objective function at sample k. Several accelerated gradient methods, which will be discussed in the next section, can be applied for updating the optimal Kalman filter gain. Sequential mini-batch gradient descent allows more opportunities to converge to a better local minimum by frequent update of the gain than the batch algorithm. This is the key to estimation of noise covariances in non-stationary systems, since the Kalman gain can be sequentially adapted as the system noise parameters change. This also enables us to experiment with dynamic convergence thresholds, while the batch estimation algorithm performs poorly with such adaptations. The generic form of gain update is

$$W^{(r+1)} = W^{(r)} - \alpha^{(r)} \nabla_{W^{(r)}} J \quad (33)$$

Equation (9) for the updated state error covariance $P(k+1|k+1)$ can be replaced with (34) in the Joseph form. Although (34) is more expensive computationally, it is less sensitive to round-off error because it guarantees that the updated state covariance matrix will remain positive definite. Raghavan *et al*. [16] suggested another way to ensure positive definiteness of the updated state covariance matrix via square root factorization algorithms.

$$P = (I_{n_x} - WH)\bar{P}(I_{n_x} - WH)' + WRW' \quad (34)$$

C. ENHANCEMENT OF COMPUTATIONAL EFFICIENCY

1) SIMPLIFIED GRADIENT COMPUTATION

Computational testing has shown that the Z term in (25) needed for the gradient computation in (24) is typically small. Consequently, we simplified the gradient computation by neglecting the Z term. Although the gradient is approximate, the simplification can reduce the computational cost of the algorithm with little or no impact on the RMSE of the covariance estimates.

$$\nabla_W J = -\sum_{i=1}^{M-1} \left[H\bar{F}^{i-1}F\right]' \varphi^2 C(i) \varphi^2 C(0) - \sum_{l=0}^{i-2} \left[C(l+1)\varphi^2 C(i)' \varphi^2 H \bar{F}^{i-l-2}\right]' \quad (35)$$

2) DYNAMIC THRESHOLDS FOR THE INNER-LOOP TERMINATION

We have found experimentally that the termination threshold during the initial outer-loop iterations can be relaxed considerably and can be made progressively tighter as the iterations progress. Consequently, we apply an exponentially decaying dynamic threshold to terminate the inner-loop gain updating process. The following outer-loop iteration-dependent dynamic threshold to terminate the inner-loop gain update was found to work well in our computational experiments. This has a substantial impact on the speedup of the sequential algorithm by up to 8 times without loss in the accuracy of RMSE. However, the batch algorithm in [21] exhibits erratic behavior with a significant increase in RMSE when dynamic thresholds are used.

$$\zeta : e^{-6} + e^{-10(\frac{t-1}{N_t})} \times (e^{-3} - e^{-6});\ t = 1, 2, \cdots, N_t \quad (36)$$

Note that the dynamic threshold is related to conditions 1 to 3 of the following termination conditions:

Condition 1: The converged Kalman filter gain W is less than the dynamic threshold ζ.
Condition 2: The gradient of Kalman filter gain (24) is less than the dynamic threshold ζ.
Condition 3: The objective function value in (19) is less than the dynamic threshold ζ from zero.
Condition 4: The number of epochs exceeds a parameter "Patience" described in [21].
Condition 5: The maximum number of inner-loop iterations is reached.

D. ACCELERATED GRADIENT METHODS

The incremental gradient algorithm in (33) can be sped up by adaptively selecting the step size $\alpha^{(r)}$.

1) BOLD DRIVER METHOD

The bold driver method used in [3], [21] is an adaptive step size algorithm. We initialize this algorithm as

$$\alpha^{(0)} = min\left(\frac{c}{K}\left(\frac{N_k}{N_s}\right)^2, c\right) \quad (37)$$

where $c > 0$ is a constant, N_s is a hyper-parameter, N_k is the total number of observed samples and $K = N_k/B$. We use a smaller step size to prevent unstable gains if the mini-batch size is small.

The step size $\alpha^{(r)}$ is determined automatically at each iteration after comparing the current objective function value $J^{(r)}$ to the previous value $J^{(r-1)}$, shown in (38):

$$\alpha^{(r)} = \begin{cases} 0.5\alpha^{(r-1)}, & \text{if } J^{(r)} > J^{(r-1)} \\ max(1.1\alpha^{(r-1)}, \bar{c}), & \text{otherwise} \end{cases} \quad (38)$$

where $\bar{c}$ is the maximum step size defined as,

$$\bar{c} = min\left(\left(\frac{N_k}{N_s}\right)^2, c_{max}\right) \quad (39)$$

We set the maximum step size c_{max} at 0.2.

2) CONSTANT STEP SIZE METHOD

Compared to the bold driver method, the constant step size method uses a fixed step size at every iteration. The step size needs to be small to avoid unstable gains.

3) SUBGRADIENT

The subgradient method [6] is a simple algorithm for minimizing non-differentiable functions. Since zero is a lower bound on the cost function (recall that innovations of an optimal Kalman filter are uncorrelated over time), we use the following step size for this method

$$\alpha^{(r)} = \alpha^{(0)} \frac{J}{||\nabla_{W^{(r)}} J||_F^2} \tag{40}$$

where $0 \leq \alpha^{(0)} \leq 2$. We used $\alpha^{(0)} = \bar{c}$ in (39). The subgradient method can be combined with the bold driver method as discussed in [5]. We have not experimented with this combination.

4) RMSProp

Root Mean Square Propagation (RMSProp) [19] keeps track of the moving average of the squared incremental gradients for each gain element for adapting the element-wise step size.

$$\tau_{r,ij} = \gamma \tau_{r-1,ij} + (1-\gamma)[(\nabla_{W^{(r)}} J)_{ij}]^2; \quad \tau_0 = 0 \tag{41}$$

$$\alpha_{ij}^{(r)} = \frac{\alpha^{(0)}}{\sqrt{\tau_{r,ij} + \epsilon}} \tag{42}$$

Here, $\gamma = 0.9$ is the default value and $\epsilon = 10^{-8}$ to prevent division by zero.

5) ADAM

Adaptive Moment Estimation (Adam) [11] update computes an adaptive learning rate from the filtered estimates of the gradient and the mean square value of the gradient. The method is appropriate for non-stationary systems and for problems with very noisy and/or sparse gradients. Adam keeps track of exponentially decaying averages of the past gradients m_r and the past mean squared gradients τ_r via

$$m_r = \beta_1 m_{r-1} + (1-\beta_1)\nabla_{W^{(r)}} J; \quad m_0 = 0 \tag{43}$$

$$\tau_{r,ij} = \beta_2 \tau_{r-1,ij} + (1-\beta_2)[(\nabla_{W^{(r)}} J)_{ij}]^2; \quad \tau_0 = 0 \tag{44}$$

Here $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ are used as default values in this paper.

Then, we compute bias-corrected gradient and mean square estimates $\hat{m}_r$ and $\hat{\tau}_r$ as

$$\hat{m}_r = \frac{m_r}{1-\beta_1^r} \tag{45}$$

$$\hat{\tau}_r = \frac{\tau_r}{1-\beta_2^r} \tag{46}$$

Finally, (33) is used with the filtered and bias-corrected gradient $\hat{m}_r$ instead of the incremental gradient $\nabla_{W^{(r)}} J$ and with the adaptive step size $\alpha_{ij}^{(r)}$ given by

$$\alpha_{ij}^{(r)} = \frac{\alpha^{(0)}}{\sqrt{\hat{\tau}_{r,ij}} + \epsilon} \tag{47}$$

6) ADADELTA

Adadelta [20] dynamically adapts the learning rate over time using a fading memory average of all past squared gradients and squared gain (parameter) changes. The filtered gradient squared τ_r and filtered squared gain increments Δ_r are computed as

$$\tau_{r,ij} = \gamma \tau_{r-1,ij} + (1-\gamma)[(\nabla_{W^{(r)}} J)_{ij}]^2; \quad \tau_0 = 0 \tag{48}$$

$$\Delta_{r,ij} = \delta \Delta_{r-1,ij} + (1-\delta)(W_{ij}^{(r)} - W_{ij}^{(r-1)})^2; \quad \Delta_0 = 0 \tag{49}$$

Typical default decay rates are $\delta = \gamma = 0.9$. The element-wise step size for Adadelta algorithm is given by

$$\alpha_{ij}^{(r)} = \frac{\sqrt{\Delta_{r-1,ij} + \epsilon}}{\sqrt{\tau_{r,ij} + \epsilon}} \tag{50}$$

The pseudocode for the multi-pass sequential mini-batch SGD estimation algorithm is included as Algorithm 2.

Algorithm 2 Pseudocode of Multi-Pass Sequential Mini-Batch Gradient Descent-Based on Adaptive Kalman Filtering Algorithm

```
input: W^0, α^0, B ▷ W^0: initial gain, α^0: initial step size, B: batch size
for t = 1 to N_t do                          ▷ N_t: Max. Num. outer-loop iter.
  for l = 1 to N_l do                        ▷ N_l: Max. Num. inner-loop iter.
    r = 0                                    ▷ Initialize the updating index r
    for k = 1 to N_k do                      ▷ N_k: Num. samples
      compute innovation correlations ν(k)
      if k > N_b + M then                    ▷ N_b: Num. burn-in samp.
        compute Ĉ
        if Mod(k, B) = 0 then
          compute objective function J
          compute gradient ∇_W J
          update the step size α^(r)
          W_ij^(r+1) = W_ij^(r) − α_ij^(r) [∇_W^(r) J)]_ij
          r = r + 1
        end if
      end if
    end for
    check the convergence every inner-loop iteration
  end for
  check the convergence every outer-loop iteration
  update R^(t) and Q^(t)
end for
```

E. SEQUENTIAL MINI-BATCH GRADIENT DESCENT METHOD FOR NON-STATIONARY SYSTEMS

Killick *et al*. [9] proposed an algorithm for detecting multiple abrupt changes in a signal sequence. The changes can be in the mean of the signal, its root-mean-square (RMS) value, or its standard deviation (STD). The sequential algorithm of this paper was coupled with the change-point detection algorithm in [9] to detect changes in the statistical behavior of innovations to restart the parameter estimation procedure at every change point. We used the RMS deviation in innovations as the characteristic change-point to be detected.

Our decision-directed estimation procedure works as follows: First, we invoke the estimation algorithm on all observation samples assuming a stationary system. Then, the change-point detection algorithm is invoked to detect abrupt changes in the RMS value of the innovation sequence using the algorithm in [9]. Then, the sequential estimation algorithm is invoked for samples between two consecutive change points, that is, the sequential estimation algorithm is restarted at every change point. The pseudocode of the sequential mini-batch SGD algorithm for non-stationary systems is shown below.

Algorithm 3 Pseudocode of Multi-Pass Sequential Mini-Batch SGD Algorithm for a Non-Stationary System

input: W^0, Q^0, R^0, α^0, B, N_k, N_p ▷ *W^0: initial gain, Q^0: initial Q,R^0: initial R, α^0: initial step size, B: batch size, N_k: Num. of samples, N_p: Num. of change points*
Obtain the innovation sequence $\{\nu(k)\}_{k=1}^{N_k}$
▷ *Call Algorithm 2 assuming a stationary system*
Detect the p abrupt change-points Ω_y, $y = 1, 2, \cdots, p$
▷ *Call Change-point detection algorithm given $\{\nu(k)\}_{k=1}^{N_k}$ and N_p described in Algorithm 4*
for $y = 1$ to p **do**
  Select samples from $N_{range} = [\Omega_{y-1} : \Omega_y]$; $\Omega_0 = 1$
  Update $R^{(y)}$, $Q^{(y)}$ ▷ *Call Algorithm 2 with samples from $N_{range} = [\Omega_{y-1} : \Omega_y]$*
end for

IV. NUMERICAL EXAMPLES

In this section, we first investigate the performance of the accelerated stochastic gradient descent (SGD) methods using the five examples in [21]. These examples consist of 1) a second-order kinematic system (a white noise acceleration or nearly constant velocity model); 2) A system described in [14]; 3) A five-state system with diagonal Q and R; 4) A detectable, but not completely observable, system; and 5) A three-state ill-conditioned system.

We perform 100 Monte Carlo simulations for each case with an assumed "patience" of 5, the lags $M = 5$, a dynamic threshold ζ in (36), $c_{max} = 0.2$, $c = 0.01$, $N_t = 20$, $N_l = 100$, $N_k = 1,000$ and $N_s = 1,000$. The lag $M = 30$ is applied to Case 1 to conform with [21], and we used 10,000 samples for Case 3. The ill-conditioned Case 5 was simulated with 200 Monte Carlo runs to be consistent with [15], [21]. We show that the SGD methods are considerably faster than the batch method used in [21], especially for difficult Cases 3-5. We show that there is a modest speedup of 10% when the Z term in (24) is neglected without loss in RMSE accuracy.

For each case, the best gradient descent method and the optimal mini-batch size were somewhat different in terms of RMSE and computation time. However, when considered in aggregate, Adam update with a mini-batch size of 64 works well across all the example cases. Consequently, we show the computational speedup of the Adam

Algorithm 4 Pseudocode of Change-Point Detection Algorithm

input: $\{\nu(k)\}_{k=1}^{N_k}$, N_p ▷ *$\{\nu(k)\}_{k=1}^{N_k}$: innovation sequence, N_p: Num. change points*
$\mathcal{E}_{RES.M} = N_k \log(\frac{1}{N_k}\sum_{k=1}^{N_k}[\nu(k)]^2)$
$\mathcal{E}_{RES.F}$, $\mathcal{E}_{RES.R}$ = zeros(size(ν(k),1)),respectively
▷ *$\mathcal{E}_{RES.M}$: Maximum residual error, $\mathcal{E}_{RES.F}$: Forward residual, $\mathcal{E}_{RES.R}$: Reverse residual*
$L_m = \log(R_{min})$
▷ *R_{min}: The smallest positive normalized floating-point number, $R_{min} = 2^{-1022}$*
for $t = 1$ to N_k **do**
  $\mathcal{E}_{RES.F} = \mathcal{E}_{RES.F} + \nu^2(t)$
  $\mathcal{E}_{RES.R} = \mathcal{E}_{RES.R} + \nu^2(N_k - t + 1)$
  $\mathcal{F}_{RES.F} = sum(max(L_m - \log(t), \log(\frac{\mathcal{E}_{RES.F}}{t})), 1)$
  $\mathcal{R}_{RES.R} = sum(max(L_m - \log(t), \log(\frac{\mathcal{E}_{RES.R}}{t})), 1)$
  $\mathcal{E}_{RES.T}(t) = t \cdot (\mathcal{F}_{RES.F} + \mathcal{R}_{RES.R})$
end for
$\beta = \mathcal{E}_{RES.M} - min(\mathcal{E}_{RES.T})$ ▷ *β: Penalty constant*
$\Omega = [\,]$ ▷ *Set Ω as change-points indexes*
$F(0) = -\beta$ ▷ *Set the optimal cost value F(0)*
while Num. of $\Omega \leqq N_p$ **do**
  $\beta = \frac{\beta}{2}$
  for $k = 1$ to N_k **do**
    $F(k) = min_{s\in R_t}[F(s) + \mathcal{C}(\nu(s+1:k)) + \beta]$
    ▷ *F(k): The optimal cost value of $\nu(1:k)$*
    $\Omega_{latest} = argmin_{s\in R_t}[F(s) + \mathcal{C}(\nu(s+1:k)) + \beta]$
    ▷ *Ω_{latest}: The latest change-point in the optimal segmentation of $\nu(1:k)$*
    $\Omega_k = [\Omega_k, \Omega_{latest}]$
    ▷ *Ω_k: The change-points in the optimal segmentation of $\nu(1:N_k)$*
    $R_t = \{s \in R_t : F(s) + \mathcal{C}(\nu(s+1:k)) < F(t)\}$
  end for
end while

version of the sequential mini-batch algorithm over the batch method. The SGD algorithms are particularly effective on Case 3 and the ill-conditioned estimation problems represented by Cases 4 and 5.

We also demonstrate the utility of the sequential method in estimating Q and R in non-stationary systems by coupling it with a change detection method [9], [12] to obtain a decision-directed adaptive Kalman filter. For the non-stationary system, we apply our proposed method to the system used in [14]. In this paper, computational simulations were run on a computer with an Intel Core i7-8665U processor and 16 GB of RAM.

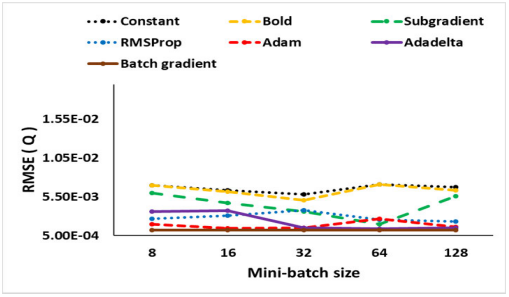
A. COMPARISON OF ACCELERATED SGD METHODS AND THE BATCH METHOD

1) CASE 1

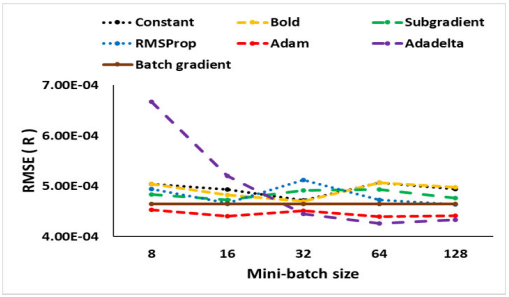
The system is described by

$$F = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix} \quad (51)$$

Fig. 1 shows the RMSE of accelerated gradient descent methods for estimating Q and R for varying mini-batch sizes. In estimating Q, RMSE values of all sequential algorithms are larger than the batch algorithm, but the RMSE itself was very small. In estimating R, RMSProp and Adam show good performance over all mini-batch sizes. Constant, Bold driver and Subgradient methods have slightly higher RMSE for Case 1. All the update methods resulted in consistent state estimates, as measured by the normalized innovation squared (NIS) metric [2].



(a) RMSE of estimated Q



(b) RMSE of estimated R

FIGURE 1. Performance of accelerated gradient descent methods on Case 1.

Table 1 shows Monte Carlo simulation results for estimating the noise parameters using Adam for various mini-batch sizes. Adam with a mini-batch size of 128 shows a good trade-off between computational efficiency and RMSE for this example.

Fig. 2 shows the averaged NIS of SGD (Adam; a mini-batch size of 64) method and the 95% probability region. As with the batch method, the Adam SGD-based Kalman filter is consistent.

2) CASE 2

For Case 2, the system simulated is assumed to be as follows.

$$F = \begin{bmatrix} 0.8 & 1 \\ -0.4 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \tag{52}$$

In Case 2, RMSProp shows similar or better performance than the batch method in estimating Q and R as shown in Fig. 3. Adam can provide very good estimates of Q and R when the mini-batch size is larger than 64. The Subgradient, Constant, and Bold driver updates tend to have higher RMSE values than the batch method for the ranges of mini-batch

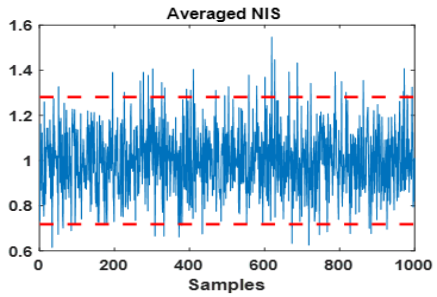
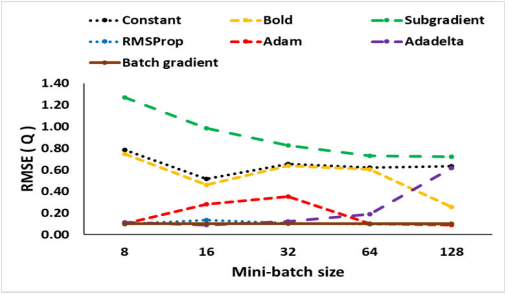
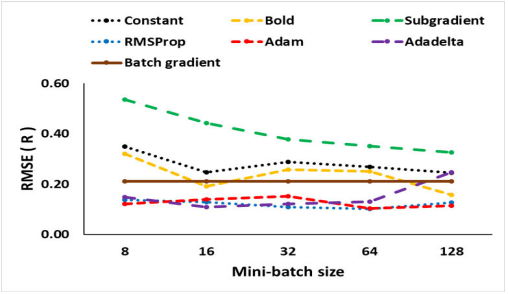


FIGURE 2. Averaged NIS of SGD method for Case 1.



(a) RMSE of estimated Q



(b) RMSE of estimated R

FIGURE 3. Performance of accelerated gradient descent methods on Case 2.

sizes examined as shown in Fig. 3, but the state estimates are consistent as measured by NIS.

As one can see from Table 2, Adam shows improved computational efficiency over the batch method for all mini-batch sizes, but it is more effective when the batch size is larger than 64. Note that RMSProp shows the least RMSE in estimating the noise parameters among the accelerated SGD methods for Case 2. In particular, RMSProp with a mini-batch size of 64 provided the best parameter estimates with the least computation time of 114 seconds for all 100 MC runs, i.e., 1.14 seconds per run (not shown).

3) CASE 3

The system matrices in Case 3 are

$$F = \begin{bmatrix} 0.75 & -1.74 & -0.3 & 0 & -0.15 \\ 0.09 & 0.91 & -0.0015 & 0 & -0.008 \\ 0 & 0 & 0.95 & 0 & 0 \\ 0 & 0 & 0 & 0.55 & 0 \\ 0 & 0 & 0 & 0 & 0.905 \end{bmatrix} \tag{53}$$

TABLE 1. Monte Carlo simulation for Case 1 with M = 30 (100 Runs; 1,000 samples).

Type (Batch size)	Computation time (s)	R (Truth = 0.01)		Q (Truth = 0.0025)		$\bar{P}_{11}$ (Truth = 0.0011)		$\bar{P}_{22}$ (Truth = $5.13 \cdot 10^{-4}$)		W_{11} (Truth = 0.0952)		W_{21} (Truth = 0.0476)	
		Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
Batch	123	0.01	$4.64 \cdot 10^{-4}$	0.0027	0.0012	0.0011	$1.55 \cdot 10^{-4}$	$5.45 \cdot 10^{-4}$	$1.97 \cdot 10^{-4}$	0.0936	0.0154	0.0480	0.0113
Adam(8)	536	0.01	$4.53 \cdot 10^{-4}$	0.0038	0.0020	0.0012	$1.61 \cdot 10^{-4}$	$6.97 \cdot 10^{-4}$	$2.76 \cdot 10^{-4}$	0.1023	0.0147	0.0569	0.0149
Adam(16)	408	0.01	$4.40 \cdot 10^{-4}$	0.0028	0.0014	0.0011	$1.52 \cdot 10^{-4}$	$5.57 \cdot 10^{-4}$	$2.14 \cdot 10^{-4}$	0.0930	0.0136	0.0486	0.0125
Adam(32)	421	0.01	$4.51 \cdot 10^{-4}$	0.0029	0.0015	0.0011	$1.80 \cdot 10^{-4}$	$5.74 \cdot 10^{-4}$	$2.40 \cdot 10^{-4}$	0.0962	0.0151	0.0497	0.0133
Adam(64)	124	0.01	$4.40 \cdot 10^{-4}$	0.0044	0.0027	0.0012	$2.06 \cdot 10^{-4}$	$7.68 \cdot 10^{-4}$	$3.64 \cdot 10^{-4}$	0.1043	0.0157	0.0607	0.0193
Adam(128)	102	0.01	$4.42 \cdot 10^{-4}$	0.0033	0.0016	0.0011	$1.51 \cdot 10^{-4}$	$6.20 \cdot 10^{-4}$	$2.30 \cdot 10^{-4}$	0.0957	0.0124	0.0524	0.0129

TABLE 2. Monte Carlo Simulation for Case 2 with M = 5 (100 Runs; 1,000 samples).

Type (Batch size)	Computation time (s)	R (Truth = 1.00)		Q (Truth = 1.00)		$\bar{P}_{11}$ (Truth = 1.89)		$\bar{P}_{22}$ (Truth = 0.35)		W_{11} (Truth = 0.65)		W_{21} (Truth = 0.09)	
		Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
Batch	478	1.03	0.21	0.98	0.10	1.88	0.12	0.35	0.02	0.64	0.07	0.10	0.06
Adam(8)	298	0.95	0.12	1.05	0.10	1.94	0.13	0.36	0.02	0.67	0.04	0.10	0.03
Adam(16)	228	0.97	0.14	1.05	0.28	1.94	0.27	0.37	0.06	0.67	0.05	0.10	0.04
Adam(32)	198	1.00	0.15	1.04	0.35	1.93	0.35	0.36	0.08	0.65	0.06	0.10	0.05
Adam(64)	157	0.94	0.10	1.05	0.10	1.94	0.13	0.36	0.03	0.67	0.03	0.10	0.02
Adam(128)	161	0.97	0.11	1.02	0.09	1.91	0.12	0.36	0.02	0.66	0.04	0.09	0.03

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 24.64 & 0 & 0 \\ 0 & 0.835 & 0 \\ 0 & 0 & 1.83 \end{bmatrix} \tag{54}$$

Fig. 4 shows that all the accelerated SGD methods have larger RMSE values than the batch method in estimating Q_{11}, and the subgradient and Adam updates show lower RMSE for the estimated R_{11} at all mini-batch sizes. The RMSE values of the SGD methods are slightly larger than the batch method in Case 3.

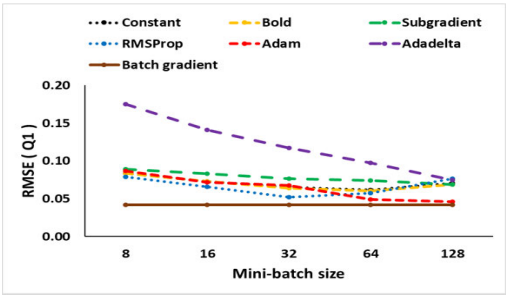
Table 3a and Table 3b show the simulation results with 100 Monte Carlo runs for estimating the noise parameters based on 10,000 samples using the Adam update. Adam with a mini-batch size of 64 shows a good trade-off between computational efficiency and RMSE among all the accelerated SGD methods.

4) CASE 4

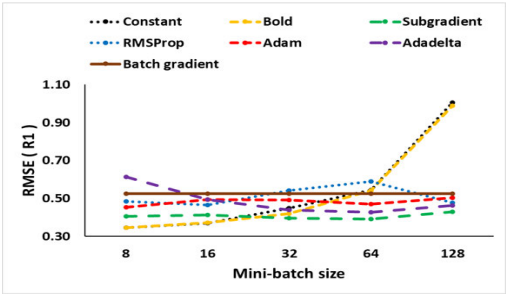
The unobservable (but detectable) system for Case 4 is described by

$$F = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.2 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \Gamma = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{55}$$

Fig. 5 clearly shows that SGD methods can estimate Q and R with lower RMSE when compared to the batch method. In this example, Constant and Bold driver methods have lower RMSE values for estimating Q and R for mini-batch sizes less than 32. The RMSE values in estimating noise parameters are smaller with larger mini-batch sizes as shown in Table 4. As in the case of batch gradient descent method, better estimates are obtained by the addition of a regularization term with $\lambda_Q = 0.1$.



(a) RMSE of estimated Q_{11}



(b) RMSE of estimated R_{11}

FIGURE 4. Performance of accelerated gradient descent methods on Case 3.

TABLE 3. ***W*** **Estimates for Case 3.**

(a) R, Q and P Estimates for Case 3

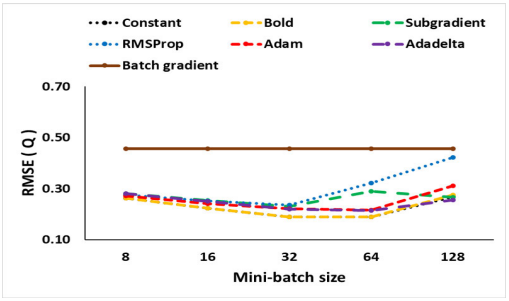
Type (Batch size)	Computation time (s)	R_{11} (Truth = 1.00)		R_{22} (Truth 1.00)		Q_{11} (Truth = 1.00)		Q_{22} (Truth = 1.00)		Q_{33} (Truth = 1.00)	
		Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
Batch	8635	1.07	0.52	1.00	0.04	0.99	0.03	1.06	0.13	1.03	0.08
Adam(8)	3159	0.90	0.45	1.05	0.07	0.95	0.09	0.82	0.19	1.22	0.25
Adam(16)	2730	1.00	0.49	1.04	0.06	0.96	0.07	0.88	0.14	1.14	0.17
Adam(32)	2227	0.93	0.49	1.02	0.05	0.97	0.07	0.94	0.13	1.12	0.16
Adam(64)	1962	0.91	0.47	1.02	0.05	0.97	0.05	0.94	0.17	1.15	0.19
Adam(128)	2018	0.93	0.50	1.03	0.06	0.98	0.05	0.88	0.15	1.18	0.21
Type (Batch size)	Computation time (s)	$\bar{P}_{11}$ (Truth = 72.31)		$\bar{P}_{22}$ (Truth 1.14)		$\bar{P}_{33}$ (Truth = 1213)		$\bar{P}_{44}$ (Truth = 0.93)		$\bar{P}_{55}$ (Truth = 11.74)	
		Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
Batch	8635	72.30	1.52	1.19	0.08	1200	32.37	0.99	0.12	12.20	0.90
Adam(8)	3159	71.89	3.61	1.15	0.05	1157	96.57	0.78	0.17	13.82	2.33
Adam(16)	2730	71.97	3.01	1.15	0.05	1173	79.23	0.83	0.13	13.03	1.62
Adam(32)	2227	71.71	3.68	1.17	0.07	1176	76.53	0.89	0.12	12.94	1.60
Adam(64)	1962	72.57	2.82	1.18	0.09	1187	54.37	0.88	0.16	13.26	1.90
Adam(128)	2018	73.16	2.44	1.16	0.06	1192	49.71	0.83	0.14	13.44	2.04

(b) W Estimates for Case 3

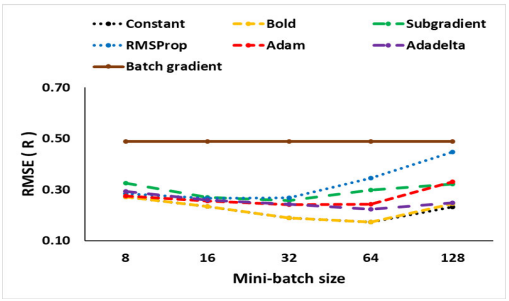
Type (Batch size)	Computation time (s)	W_{11} (Truth = 0.59)		W_{21} (Truth = $2.80 \cdot 10^{-3}$)		W_{31} (Truth = -2.86)		W_{41} (Truth = $-1.76 \cdot 10^{-4}$)		W_{51} (Truth = 0.03)	
		Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
Batch	8635	0.96	0.01	$3.69 \cdot 10^{-4}$	$7.51 \cdot 10^{-3}$	-2.84	0.04	$3.40 \cdot 10^{-3}$	$1.17 \cdot 10^{-2}$	0.03	0.01
Adam(8)	3159	0.95	0.01	$1.81 \cdot 10^{-3}$	$2.71 \cdot 10^{-3}$	-2.78	0.12	$5.76 \cdot 10^{-4}$	$3.22 \cdot 10^{-3}$	0.04	0.02
Adam(16)	2730	0.96	0.01	$1.41 \cdot 10^{-3}$	$3.69 \cdot 10^{-3}$	-2.81	0.10	$6.48 \cdot 10^{-4}$	$4.61 \cdot 10^{-3}$	0.04	0.01
Adam(32)	2227	0.96	0.02	$-1.07 \cdot 10^{-3}$	$8.08 \cdot 10^{-3}$	-2.81	0.10	$4.51 \cdot 10^{-3}$	$1.04 \cdot 10^{-2}$	0.03	0.02
Adam(64)	1962	0.96	0.01	$-1.76 \cdot 10^{-4}$	$7.65 \cdot 10^{-3}$	-2.82	0.07	$3.34 \cdot 10^{-3}$	$1.09 \cdot 10^{-2}$	0.03	0.01
Adam(128)	2018	0.95	0.01	$1.43 \cdot 10^{-3}$	$3.60 \cdot 10^{-3}$	-2.83	0.07	$1.20 \cdot 10^{-3}$	$5.27 \cdot 10^{-3}$	0.04	0.01
Type (Batch size)	Computation time (s)	W_{12} (Truth = 0.77)		W_{22} (Truth = 0.34)		W_{32} (Truth = -1.49)		W_{42} (Truth = 0.25)		W_{52} (Truth = -0.77)	
		Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
Batch	8635	0.78	0.03	0.33	0.01	-1.51	0.05	0.26	0.02	-0.78	0.03
Adam(8)	3159	0.88	0.12	0.35	0.02	-1.58	0.11	0.22	0.04	-0.87	0.11
Adam(16)	2730	0.84	0.08	0.35	0.01	-1.56	0.08	0.23	0.03	-0.84	0.08
Adam(32)	2227	0.83	0.08	0.34	0.01	-1.55	0.08	0.24	0.03	-0.83	0.08
Adam(64)	1962	0.84	0.09	0.34	0.02	-1.56	0.09	0.24	0.03	-0.84	0.09
Adam(128)	2018	0.86	0.10	0.35	0.02	-1.57	0.09	0.23	0.03	-0.86	0.10

TABLE 4. **Monte Carlo Simulation for Case 4 with M = 5 (100 Runs; 1,000 samples).**

Type (Batch size)	Computation time (s)	R (Truth = 1.00)		Q (Truth = 1.00)		$\bar{P}_{11}$ (Truth = 1.01)		$\bar{P}_{22}$ (Truth = 4.08)		W_{11} (Truth = 0.50)		W_{21} (Truth = 1.01)	
		Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
Batch	1444	0.98	0.49	1.06	0.46	1.06	0.46	4.29	1.81	0.51	0.25	1.04	0.45
Adam(8)	138	1.25	0.28	0.75	0.27	0.75	0.27	3.07	1.09	0.37	0.14	0.74	0.29
Adam(16)	146	1.20	0.26	0.78	0.24	0.79	0.24	3.21	0.97	0.40	0.13	0.77	0.26
Adam(32)	163	1.17	0.24	0.82	0.22	0.82	0.22	3.35	0.89	0.41	0.12	0.80	0.24
Adam(64)	181	1.14	0.24	0.85	0.21	0.85	0.22	3.47	0.87	0.43	0.12	0.84	0.22
Adam(128)	260	1.05	0.33	0.94	0.31	0.95	0.31	3.85	1.24	0.47	0.17	0.93	0.30

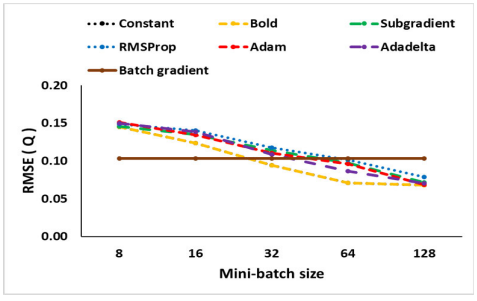


(a) RMSE of estimated Q

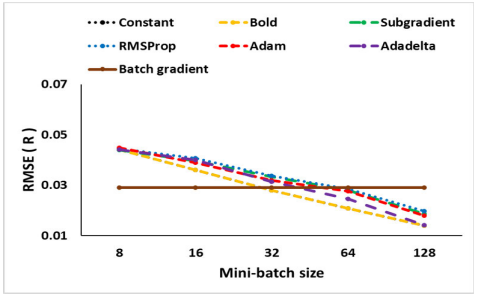


(b) RMSE of estimated R

FIGURE 5. Performance of accelerated gradient descent methods on Case 4.



(a) RMSE of estimated Q



(b) RMSE of estimated R

FIGURE 6. Performance of accelerated gradient descent methods on Case 5.

5) CASE 5

The system matrices for Case 5 are assumed to be as follows.

$$F = \begin{bmatrix} 0.1 & 0 & 0.1 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}, \; H = \begin{bmatrix} 0.1 & 0.2 & 0 \end{bmatrix}, \; \Gamma = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{56}$$

Fig. 6 shows the performance of accelerated SGD methods. RMSE values of estimated Q and R decrease as the mini-batch size increases, and the RMSE of all SGD methods are smaller than that of the batch method when the batch size is larger than 64. Note that Constant and Bold driver methods have lower RMSE values among the accelerated SGD methods.

As shown in Table 5, the sequential algorithm improves the computational efficiency by up to 8 times when compared to the batch method on Case 5. Note that Adam update with a mini-batch size of 128 estimates the noise parameter closer to the corresponding true value. We can improve the estimation accuracy by using a regularization term with $\lambda_Q = 0.3$.

B. COMPUTATIONAL SPEEDUP ANALYSIS

1) SIMPLIFIED GRADIENT COMPUTATIONS

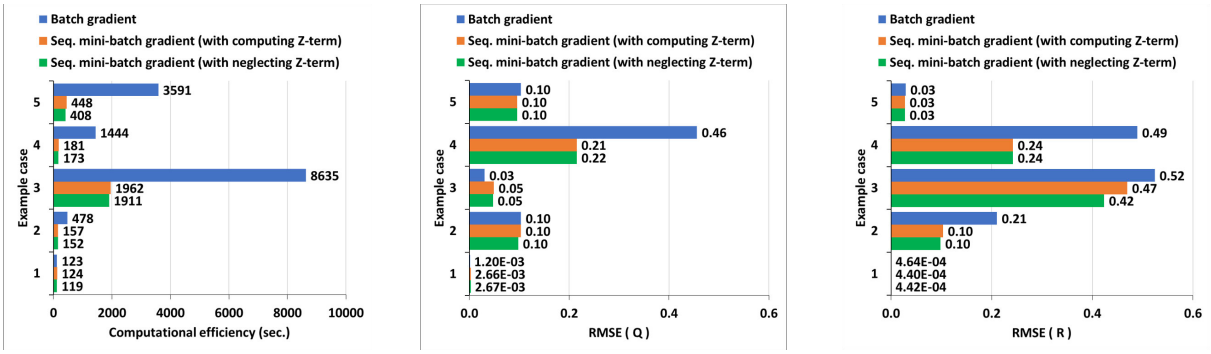
Fig. 7 shows the computational efficiency of the Adam update with a mini-batch size of 64 over the batch algorithm in [21] on all five example cases. The proposed method improves the computational efficiency by a factor of 4.4 on Case 3 and a factor of 8 on Cases 4 and 5 when compared to the batch gradient descent method. By neglecting the Z term in the gradient computations (24), the computation time is reduced by less than 10% with almost the same RMSE.

2) DYNAMIC THRESHOLDS

Using a dynamic threshold has a substantial impact on the speedup of the sequential algorithm, but it is not suitable for implementation in the batch estimation algorithm. Table 6 shows the estimation results of batch algorithm with

TABLE 5. Monte Carlo Simulation for Case 5 with M = 5 (200 Runs; 1,000 samples).

Type (Batch size)	Computation time (s)	R (Truth = 0.10)		Q (Truth = 0.50)		$\bar{P}_{11}$ (Truth = 0.54)		$\bar{P}_{22}$ (Truth = 2.04)		$\bar{P}_{33}$ (Truth = 4.69)		W_{11} (Truth = 1.14)		W_{21} (Truth = 2.24)		W_{31} (Truth = 3.39)	
		Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
Batch	3591	0.11	0.03	0.49	0.10	0.53	0.10	1.99	0.41	4.59	0.93	1.06	0.25	2.07	0.50	3.14	0.74
Adam(8)	794	0.12	0.04	0.44	0.15	0.47	0.15	1.78	0.61	4.10	1.37	0.93	0.39	1.81	0.77	2.75	1.16
Adam(16)	564	0.12	0.04	0.45	0.13	0.48	0.14	1.82	0.54	4.19	1.22	0.95	0.34	1.86	0.68	2.83	1.02
Adam(32)	477	0.12	0.03	0.46	0.11	0.50	0.10	1.89	0.44	4.35	1.00	1.00	0.28	1.95	0.55	2.96	0.83
Adam(64)	448	0.11	0.03	0.48	0.10	0.51	0.10	1.94	0.38	4.48	0.86	1.03	0.24	2.01	0.47	3.06	0.71
Adam(128)	485	0.10	0.02	0.51	0.07	0.55	0.07	2.07	0.27	4.77	0.62	1.11	0.15	2.16	0.30	3.28	0.45



(a) Computational efficiency (b) Estimated Q (c) Estimated R

FIGURE 7. The improvement of computational efficiency.

TABLE 6. Batch estimation algorithm with dynamic thresholds for Case 2 and Case 4 (100 Runs; 1,000 samples; M = 5 each case).

Example case		Case 2		Case 4	
Threshold type		Fixed.	Dynamic.	Fixed.	Dynamic.
Computation time (s)		478	9	1444	6
R	Truth	1.00		1.00	
	Mean	1.03	0.31	0.98	1.32
	RMSE	0.21	0.69	0.49	0.33
Q	Truth	1.00		1.00	
	Mean	0.98	2.62	1.06	0.69
	RMSE	0.10	1.62	0.46	0.31
$\bar{P}_{11}$	Truth	1.89		1.01	
	Mean	1.88	3.28	1.06	0.70
	RMSE	0.12	1.40	0.46	0.31
$\bar{P}_{22}$	Truth	0.35		4.08	
	Mean	0.35	0.70	4.29	2.84
	RMSE	0.02	0.35	1.81	1.25
W_{11}	Truth	0.65		0.50	
	Mean	0.64	0.91	0.51	0.34
	RMSE	0.07	0.26	0.25	0.16
W_{21}	Truth	0.09		1.01	
	Mean	0.10	0.33	1.04	0.68
	RMSE	0.06	0.24	0.45	0.33

a dynamic threshold applied for Case 2 and Case 4, respectively. In each example case, the batch algorithm with a dynamic threshold estimates parameters that are far from their corresponding true values. Consequently, a fixed threshold of 10^{-6} is used for conditions 1 to 3 discussed in III-C2 for terminating the batch algorithm.

C. APPLICATION TO NON-STATIONARY SYSTEMS

We simulated 100 MC runs with 5,000 observation samples in each run using system matrices of Case 2 in (52) and varying Q and R. We considered three non-stationary scenarios: 1) Only Q changes every 1,000 samples, 2) Only R changes every 1,000 samples, and 3) both Q and R change every 1,000 samples. In the estimation procedure, we set the outer-loop iterations $N_t = 20$, inner-loop iterations $N_l = 100$, number of burn-in of samples $N_b = 50$, $c = 0.2$ and the number of lags $M = 5$. The iteration-dependent dynamic thresholds for convergence were also applied.

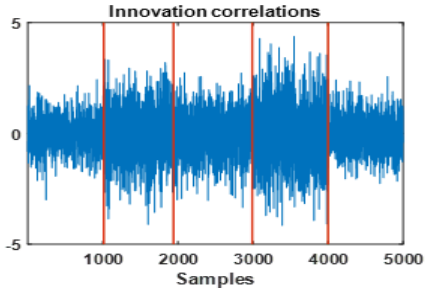


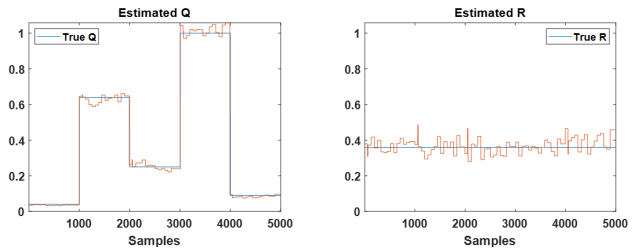
FIGURE 8. The change-points in innovation correlations in a non-stationary system.

For non-stationary systems, both RMSProp and Adam showed the best performance among all the accelerated SGD algorithms. Here, we show the results averaged over 100 Monte Carlo runs when using the Adam update with a mini-batch size of 64. Given innovation correlations, the algorithm for detecting the change points is able to track the multiple abrupt changes accurately, as shown in Fig. 8.

1) VARYING Q

In this scenario, the process noise variance Q changes every 1,000 samples, while measurement noise variance R is constant over all 5,000 samples, as shown in Fig. 9. The decision-directed estimation algorithm can track Q and R values quite well and the resulting Kalman filter is consistent as measured by NIS.

Table 7a shows the MC simulation results for the non-stationary system with varying Q, and the estimated parameters are very close to their corresponding true values.



(a) Estimated Q (b) Estimated R

FIGURE 9. Trajectory of Q and R in varying Q system.

TABLE 7. The case of varying *Q* and *R*.

(a) The case of varying Q

	R			Q			$\bar{P}_{11}$			$\bar{P}_{22}$			W_{11}			W_{21}		
	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE
1	0.36	0.37		0.04	0.04		0.11	0.12		0.02	0.03		0.24	0.25		-0.01	-0.02	
2	0.36	0.36		0.64	0.65		1.07	1.07		0.20	0.20		0.75	0.75		0.15	0.15	
3	0.36	0.36	0.05	0.25	0.26	0.07	0.51	0.52	0.29	0.10	0.10	0.05	0.59	0.59	0.14	0.06	0.06	0.06
4	0.36	0.36		1.00	0.99		1.52	1.50		0.30	0.29		0.81	0.80		0.20	0.20	
5	0.36	0.35		0.09	0.09		0.22	0.22		0.04	0.04		0.38	0.38		0.001	0.003	

(b) The case of varying R

	R			Q			$\bar{P}_{11}$			$\bar{P}_{22}$			W_{11}			W_{21}		
	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE
1	0.04	0.04		0.36	0.36		0.45	0.45		0.10	0.10		0.92	0.91		0.34	0.32	
2	0.64	0.64		0.36	0.36		0.77	0.77		0.15	0.15		0.55	0.55		0.04	0.04	
3	0.25	0.25	0.09	0.36	0.36	0.04	0.63	0.63	0.42	0.12	0.12	0.08	0.72	0.71	0.12	0.13	0.12	0.09
4	1.00	1.00		0.36	0.37		0.84	0.85		0.16	0.16		0.46	0.46		0.02	0.01	
5	0.09	0.09		0.36	0.36		0.51	0.51		0.10	0.10		0.85	0.85		0.24	0.24	

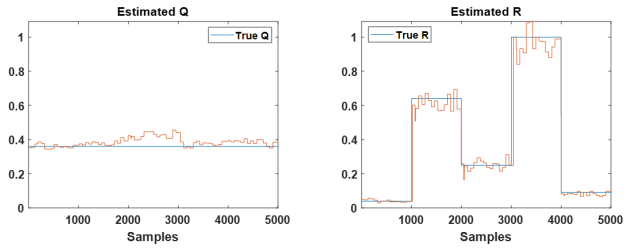
(c) The case of varying Q and R

	R			Q			$\bar{P}_{11}$			$\bar{P}_{22}$			W_{11}			W_{21}		
	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE	Truth	Mean	RMSE
1	0.42	0.43		0.04	0.04		0.11	0.12		0.02	0.03		0.21	0.22		-0.01	-0.01	
2	0.81	0.80		0.64	0.63		1.27	1.26		0.24	0.24		0.61	0.61		0.07	0.07	
3	0.49	0.47	0.08	0.25	0.26	0.08	0.54	0.56	0.28	0.10	0.11	0.05	0.53	0.54	0.15	0.03	0.04	0.05
4	0.16	0.17		1.00	0.99		1.31	1.31		0.27	0.27		0.89	0.88		0.30	0.29	
5	0.64	0.63		0.09	0.09		0.24	0.25		0.05	0.05		0.28	0.28		-0.01	-0.01	

2) VARYING *R*

In the scenario, measurement noise variance R changes every 1,000 observation samples, but the process noise variance Q is kept constant over the 5,000 samples. Fig. 10 shows that the sequential mini-batch gradient descent algorithm can track Q and R quite accurately.

As shown in Table 7b, the noise parameters can be estimated accurately by the sequential algorithm.



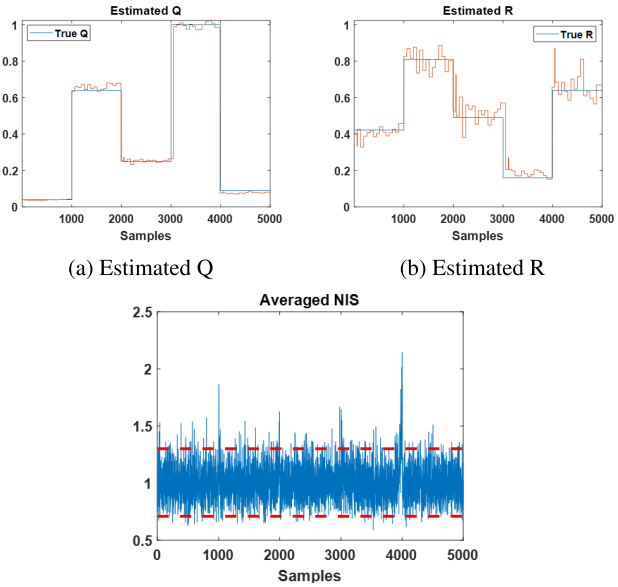
(a) Estimated Q (b) Estimated R

FIGURE 10. Trajectory of Q and R in varying R system.

3) VARYING *Q* AND *R*

The situation where Q and R vary is more difficult to track than when either Q and R is constant. However, the sequential algorithm, when coupled with the change-point detection algorithm, can track Q and R accurately as shown in Fig. 11. Fig. 11*c* shows the averaged NIS of the SGD (Adam; a mini-batch size of 64) algorithm when Q and R are varied. The Adam SGD-based Kalman filter is consistent as measured by NIS.

In the scenario when both Q and R vary, Table 7c shows that the proposed method can track the noise parameters quite accurately resulting in a consistent filter.

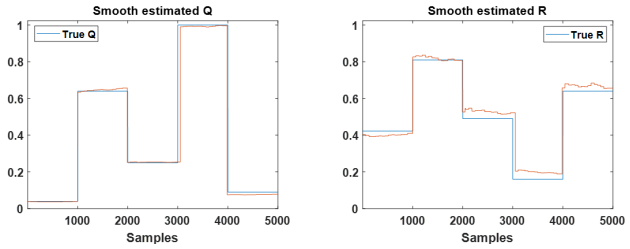


(a) Estimated Q (b) Estimated R

(c) Averaged NIS of SGD method in the non-stationary system

FIGURE 11. Trajectory of Q and R in varying Q and R system.

The trajectory of Q and R estimates can be smoothed by a simple first order fading memory filter. Fig. 12 shows the trajectory results with a smoothing weight of 0.9.



(a) Estimated Q (b) Estimated R

FIGURE 12. Trajectory of Q and R with signal smoothing at smoothing weight = 0.9.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented multi-pass stochastic gradient descent algorithms for the noise covariance estimation in adaptive Kalman filters that are an order of magnitude faster than the batch method for the same or better root mean square error and are applicable to non-stationary systems where the noise covariances can occasionally exhibit abrupt, but finite, jumps. The computational efficiency of the new algorithms stems from adaptive thresholds for convergence, recursive fading memory estimation of the sample cross-correlations of the innovations, and accelerated stochastic gradient descent algorithms. The comparative evaluation of the proposed method on a number of test cases demonstrated its computational efficiency, accuracy and filter consistency.

A limitation of the proposed method for noise covariance estimation is that it requires multiple passes through the observed samples to converge, thereby increasing the computational cost of the algorithm. For non-stationary systems, the accuracy of noise covariance estimates depends on the accuracy of the change-point detection algorithm because the proposed method is invoked for observed samples between consecutive change points. Our preliminary work on a single-pass algorithm that avoids the change-point detection step and multiple passes through the data may be found in [10]

In the future, we plan to pursue a number of research avenues, including 1) estimating Q and R using one-step lag smoothed residuals; 2) automatic model selection from a library of models; 3) formalization of decision-directed adaptive Kalman filtering with probabilistic change-point detection and state estimation; 4) explore the utility of the covariance estimation algorithm in adaptive interacting multiple model filters or as an alternative to interactive multiple model filters.

REFERENCES

[1] W. F. Arnold and A. J. Laub, "Generalized eigenproblem algorithms and software for algebraic Riccati equations," *Proc. IEEE*, vol. 72, no. 12, pp. 1746–1754, Dec. 1984.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2004.

[3] R. Battiti, "Accelerated backpropagation learning: Two optimization methods," *Complex Syst.*, vol. 3, no. 4, pp. 331–342, 1989.

[4] P. R. Bélanger, "Estimation of noise covariance matrices for a linear time-varying stochastic process," *Automatica*, vol. 10, no. 3, pp. 267–275, May 1974.

[5] D. Bertsekas, "Nonlinear programming," in *Athena Scientific Optimization and Computation Series*. Nashua, NH, USA: Athena Scientific, 2016. [Online]. Available: https://books.google.com/books?id=TwOujgEACAAJ

[6] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," Autumn Quarter, Stanford Univ., Stanford, CA, USA, Tech. Rep. Lect. Notes EE392o, 2004, pp. 2004–2005.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3. Baltimore, MD, USA: JHU Press, 2013.

[8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[9] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *J. Amer. Stat. Assoc.*, vol. 107, no. 500, pp. 1590–1598, Dec. 2012.

[10] H.-S. Kim, L. Zhang, A. Bienkowski, and K. Pattipati, "A single-pass noise covariance estimation algorithm in adaptive Kalman filtering for non-stationary systems," *TechRxiv*, Jun. 2021. [Online]. Available: https://www.techrxiv.org/articles/preprint/A_Single-pass_Noise_Covariance_Estimation_Algorithm_in_Adaptive_Kalman_Filtering_for_Non-stationary_Systems/14761005/1, doi: 10.36227/techrxiv.14761005.v1.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[12] M. Lavielle, "Using penalized contrasts for the change-point problem," *Signal Process.*, vol. 85, no. 8, pp. 1501–1510, Aug. 2005.

[13] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Trans. Autom. Control*, vol. AC-15, no. 2, pp. 175–184, Apr. 1970.

[14] C. Neethling and P. Young, "Comments on' identification of optimum filter steady-state gain for systems with unknown noise covariances,"' *IEEE Trans. Autom. Control*, vol. AC-19, no. 5, pp. 623–625, Oct. 1974.

[15] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings, "A new autocovariance least-squares method for estimating noise covariances," *Automatica*, vol. 42, no. 2, pp. 303–308, Feb. 2006.

[16] V. Raghavan, K. R. Pattipati, and Y. Bar-Shalom, "Efficient L-D factorization algorithms for PDA, IMM, and IMMPDA filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 4, pp. 1297–1310, Oct. 1993.

[17] S. Sarkka and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Trans. Autom. Control*, vol. 54, no. 3, pp. 596–600, Mar. 2009.

[18] K. Tajima, "Estimation of steady-state Kalman filter gain," *IEEE Trans. Autom. Control*, vol. AC-23, no. 5, pp. 944–945, Oct. 1978.

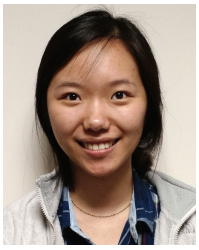
[19] T. Tieleman and G. Hinton, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[20] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: http://arxiv.org/abs/1212.5701

[21] L. Zhang, D. Sidoti, A. Bienkowski, K. R. Pattipati, Y. Bar-Shalom, and D. L. Kleinman, "On the identification of noise covariances and adaptive Kalman filtering: A new look at a 50 year-old problem," *IEEE Access*, vol. 8, pp. 59362–59388, 2020.

HEE-SEUNG KIM (Member, IEEE) received the B.S. and M.S. degrees from the Department of Computer and Communication Engineering, Chungbuk National University, Cheongju, South Korea, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA.

He was a Researcher with the Electronics and Telecommunications Research Institute (ETRI), Gwangju, South Korea, from 2013 to 2016. His research interests include machine learning, image processing, and adaptive Kalman filtering.



LINGYI ZHANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Connecticut, Storrs, in 2014, 2019, and 2020, respectively.

Her current research interests include modeling dynamic and uncertain environments for asset allocation and path planning, optimization-based techniques for mission planning and coordination, and adaptive Kalman filtering. She was a co-recipient of the Tammy Blair Award for Best Student Paper at FUSION 2016.

ADAM BIENKOWSKI (Member, IEEE) received the B.S. and M.Eng. degrees in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2013 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, under the advisement of Dr. K. R. Pattipati.

He was an Electrical Engineer at General Dynamics Electric Boat, Groton, CT, USA, from 2013 to 2017. His current research interests include modeling dynamic and uncertain environments for asset allocation and path planning, context aware decision support systems, and optimization and machine learning-based techniques for mission planning and coordination.

KRISHNA R. PATTIPATI (Life Fellow, IEEE) received the B.Tech. degree (Hons.) in electrical engineering from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from UCONN, Storrs, in 1977 and 1980, respectively.

He was with ALPHATECH, Inc., Burlington, MA, USA, from 1980 to 1986. He has been with the Department of Electrical and Computer Engineering, UCONN, since 1986, where he is currently the Board of Trustees Distinguished Professor and the UTC Chair Professor of systems engineering. His research interests include the application of systems theory, optimization, and inference techniques to agile planning, and anomaly detection, diagnostics, and prognostics. He has published over 500 scholarly journal articles and conference papers in these areas. He is a Co-Founder of Qualtech Systems, Inc., a firm specializing in advanced integrated diagnostics software tools (TEAMS, TEAMS-RT, TEAMS-RDS, TEAMATE, and PackNGo), and serves on the board for Aptima, Inc.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics (SMC) Society as the Outstanding Young Engineer of 1984 and received the Centennial Key to the Future Award. He has served as the Editor-in-Chief for IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS–PART B, from 1998 to 2001. He was a co-recipient of the Andrew P. Sage Award for the Best SMC Transactions Paper, in 1999, the Barry Carlton Award for the Best AES Transactions Paper, in 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," and "Real-time Update of Fault-Test Dependencies of Dynamic Systems: A Comprehensive Toolset for Model-Based Health Monitoring and Diagnostics," the 2005 School of Engineering Outstanding Teaching Award, and the 2003 AAUP Research Excellence Award at UCONN. He is an elected fellow of the Connecticut Academy of Science and Engineering.