To appear in the *International Journal of Control*
Vol. 00, No. 00, Month 20XX, 1–21

REGULAR PAPER

Noise Covariance Identification for Time-varying and Nonlinear Systems

Ming Ge[a*] and Eric C. Kerrigan[b*]

[a]*Department of Electrical & Electronic Engineering, Imperial College London, London SW7 2AZ, UK;* [b]*Department of Electrical & Electronic Engineering and Department of Aeronautics, Imperial College London, London SW7 2AZ, UK*

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

Kalman-based state estimators assume a priori knowledge of the covariance matrices of the process and observation noise. However, in most practical situations, noise statistics and initial conditions are often unknown and need to be estimated from measurement data. This paper presents an auto-covariance least squares based algorithm for noise and initial state error covariance estimation of large scale linear time-varying and nonlinear systems. Compared to existing auto-covariance least squares (ALS) based algorithms, our method does not involve any approximations for LTV systems, has fewer parameters to determine and is more memory/computationally efficient for large scale systems. For nonlinear systems, our algorithm uses full information estimation (FIE)/moving horizon estimation (MHE) instead of the extended Kalman filter (EKF), so that the stability and accuracy of noise covariance estimation for nonlinear systems can be guaranteed or improved, respectively.

Keywords: Auto-covariance Least-squares, Noise Covariance Estimation, State Estimation, Linear Time-varying Systems, Nonlinear Systems, Kalman Filter, Extended Kalman Filter, Moving Horizon Estimation

Index to information contained in this paper

1. Introduction

The performance of a Kalman filter relies on properly defined noise statistics. Failure to do so in the design of a Kalman filter could result in large estimation errors or even a divergence of state estimates.

In the past four decades, many approaches have been taken for improving the accuracy of noise covariance estimation. The pioneering work of noise covariance estimation in Mehra (1972) intro-

[*] Corresponding author. Email: ming.ge08@imperial.ac.uk

[*] Email: e.kerrigan@imperial.ac.uk

duced two correlation least-squares based algorithms, namely output and innovation correlation methods, for obtaining the noise covariance matrices of linear time-invariant (LTI) systems.

In Odelson, Rajamani, and Rawlings (2006), an algorithm for noise covariance estimation of LTI systems is presented, which is a constrained auto-covariance least-squares (ALS) method inspired by the innovation correlation method of Mehra (1972). The method estimates noise covariance matrices using least-squares semi-definite programming (SDP), rather than solving an unconstrained optimization problem, and greatly reduces the variance of the estimation compared to the innovation correlation method. A "one-column" version, which is a simplified version of the ALS method, is given in Rajamani and Rawlings (2009). The computational complexity of constructing the ALS problem for high dimensional systems can be significantly reduced; instead of using the identity matrix as the weight in the least-squares cost function, a method was proposed in Rajamani and Rawlings (2009) to calculate the optimal weighting for further minimizing the variance of the estimation error.

The papers cited above are for LTI models only. The standard ALS method was extended to linear time-varying (LTV) and nonlinear systems in Rajamani and Rawlings (2007). However, due to the structure and approximations, if the historical data is not sufficiently long, the existing algorithm may not be able to provide accurate estimates. We therefore provide a new algorithm for noise covariance estimation by removing all approximations in the existing formulation, so the structure of our algorithm becomes simpler, has less parameters to determine and is able to provide more accurate results.

Our algorithm also provides an estimate of the initial state error covariance P_1 that is required in most state estimation algorithms. For high dimensional systems, we provide a less memory demanding formulation by splitting high dimensional Kronecker product in the sum of smaller dimensional Kronecker products, so that our ALS algorithm can be quickly applied to more realistically sized applications.

For nonlinear systems, we replace the extended Kalman filter (EKF) with full information estimation (FIE) (Rawlings and Bakshi, 2006) to reduce the possibility of instability and divergence of estimated states, thus the accuracy of noise covariance estimation for nonlinear systems can be improved.

2. Nomenclature and Preliminaries

$\mathbb{E}[\cdot]$ and $\mathrm{cov}(\cdot)$ denote the expected value and covariance of a random variable, respectively. $\|.\|_F$ is the Frobenius norm of a matrix. $P \succ 0$ denotes that P is a positive-definite symmetric matrix. $\mathbf{1}_{n_r,n_c}$ and $\mathbf{0}_{n_r,n_c}$ represent $n_r \times n_c$ matrices with all entries equal to one or zero, respectively. I_n denotes the $n \times n$ identity matrix. $A \circ B$ denotes the Schur or Hadamard product of two same size matrices A and B. $\otimes$ denotes the standard Kronecker product. $\oplus$ is the matrix direct sum:

$$\bigoplus_{k=1}^{N} G_k := \mathrm{diag}\left(G_1, \cdots, G_N\right).$$

$\Re$ denotes the set of real number. $(\cdot)_s$ denotes the column-wise vectorization of a matrix; recall also that $(ABC)_s = (C^\top \otimes A)(B)_s$. $\mathcal{I}_{N,q} \in \Re^{(qN)^2 \times q^2}$ denotes a permutation matrix containing only zeros and ones so that $\big(I_N \otimes R\big)_s = \mathcal{I}_{N,q}(R)_s$ (Odelson et al., 2006). $(\cdot)_{ss}$ represents the column-wise stacked lower triangular elements of a symmetric matrix, hence it is possible to establish a relationship between $(\cdot)_s$ and $(\cdot)_{ss}$ as $(\hat{Q})_s = \mathscr{D}_r(\hat{Q})_{ss}$, where $\mathscr{D}_r \in \Re^{r^2 \times \frac{r(r+1)}{2}}$ is a full column rank duplication matrix, which contains only zeros and ones (Magnus and Neudecker, 1999, p. 57).

For a sequence of squared matrices $(A_k)_{k=1}^{M}$, the notation $\prod_{k=m_0}^{m_1}{}^{(b)} A_k$ with $m_0 \leq M$ and $m_1 \leq M$

represents backwards matrix multiplication of $(A_k)_{k=m_0}^{m_1}$, i.e.

$$\prod_{k=m_0}^{m_1} {}^{(b)} A_k := \begin{cases} A_{m_1} \times A_{m_1-1} \times \cdots \times A_{m_0} & \text{if } m_1 > m_0 \\ A_{m_1} & \text{if } m_1 = m_0 \\ I & \text{if } m_1 < m_0 \end{cases}$$

$\mathscr{M}_l^{r,c} \in \Re^{r\times c}$ is an auxiliary matrix containing only zeros and ones $\mathscr{M}_l^{r,c} := \begin{bmatrix}\mathbf{0}_{r\times(l-1)} & I_r & \mathbf{0}_{r\times(c-r-l+1)}\end{bmatrix}$. $x \sim \mathcal{N}(\mu, P)$ denotes a random vector variable x with a normal distribution with mean μ and covariance matrix P. $x \sim \mathcal{U}(a, b)$ denotes a uniformly distributed scalar random variable x within the interval $[a, b]$. $\mathscr{R}(A)$ denotes the range or column space of a matrix $A \in \Re^{n\times m}$, i.e. $\mathscr{R}(A) := \{Ax : x \in \Re^m\}$.

3. Linear Time-varying Systems

Consider the following discrete-time LTV model:

$$\begin{aligned} x_{k+1} &:= A_k x_k + G_k w_k, \\ y_k &:= C_k x_k + H_k v_k, \end{aligned} \tag{1}$$

where $A_k \in \Re^{n\times n}$ and $C_k \in \Re^{p\times n}$ are the dynamics and sensor matrices; $G_k \in \Re^{n\times r}$ and $H_k \in \Re^{p\times q}$ are time-varying full column rank matrices; $(x_k)_{k=1}^M$ are unknown state sequences; $(y_k)_{k=1}^M$ are given outputs; $(w_k)_{k=1}^M$ and $(v_k)_{k=1}^M$ are two unknown noise sequences, which affect the state and output, respectively.

Assumption 1: *The noise sequences $(w_k)_{k=1}^M$ and $(v_k)_{k=1}^M$ are two uncorrelated random variables having Gaussian (or normal) distributions $\mathcal{N}(0, Q)$ and $\mathcal{N}(0, R)$, respectively, with zero mean and unknown positive-definite covariance matrices Q and R.*

Assumption 2: *The pair $\begin{bmatrix}A_k & C_k\end{bmatrix}$ in the LTV system (1) is uniformly detectable (Anderson and Moore, 1981) for $1 \le k \le M$.*

Theorem 1: *(Anderson and Moore, 1981) If Assumption 2 holds and the sequences $(A_k)_{k=1}^M$, $(C_k)_{k=1}^M$ are bounded, then there exists a bounded filter gain sequence $(L_k)_{k=1}^M$ such that all trajectories of the closed-loop LTV system*

$$x_{k+1} := \bar{A}_k x_k := (A_k - A_k L_k C_k)\, x_k$$

are exponentially stable over a finite horizon $k = 1, \dots, M$.

Since the true noise covariance matrices Q and R are unknown, it is not possible to design a best linear unbiased estimator (BLUE) with a sequence of optimal filter gains $(L_k)_{k=1}^M$. Instead, we use a given sequence of stabilizing sub-optimal filter gains $(L_k^s)_{k=1}^M$ and any appropriate given initial state guess $\hat{x}_{1|0}$ to obtain the estimated state sequence

$$\hat{x}_k := \hat{x}_{k|k} := \hat{x}_{k|k-1} + L_k^s \left(y_k - \hat{y}_{k|k-1}\right), \tag{2a}$$

where $\hat{x}_{k+1|k}$ and $\hat{y}_{k|k-1}$ are the one-step ahead predicted state and output, respectively, given by

$$\hat{x}_{k+1|k} := A_k \hat{x}_k, \qquad \hat{y}_{k|k-1} := C_k \hat{x}_{k|k-1}. \tag{2b}$$

The state error terms are defined as $\varepsilon_k := x_k - \hat{x}_{k|k-1}$, for $k = 1, \ldots, M$, hence

$$\hat{x}_{k+1|k} = A_k(\hat{x}_{k|k-1} + L_k^s(y_k - \hat{y}_{k|k-1})) = A_k\hat{x}_{k|k-1} + A_kL_k^s(C_kx_k + H_kv_k - C_k\hat{x}_{k|k-1}).$$

Let $\bar{G}_k := \begin{bmatrix} G_k & -A_kL_k^sH_k \end{bmatrix}$ and $\bar{w}_k := \begin{bmatrix} w_k^\top & v_k^\top \end{bmatrix}^\top$, then

$$\varepsilon_{k+1} = \bar{A}_k\varepsilon_k + \bar{G}_k\bar{w}_k. \tag{3a}$$

Therefore, if Assumption 2 holds, Theorem 1 will ensure that $\mathbb{E}[\varepsilon_k] \to 0$ as $k \to \infty$.

We define the state space model of innovations as

$$z_k := C_k\varepsilon_k + H_kv_k. \tag{3b}$$

A necessary and sufficient condition for the optimality of a Kalman filter is that the innovation sequence $(z_k)_{k=1}^M$ be white Gaussian noise (Mehra, 1970), (Kailath, 1968). However, for a sub-optimal filter, $z_1,\ z_2,\ \ldots,\ z_M$ are correlated with each other, thus we could produce an auto-covariance matrix of $(z_k)_{k=1}^M$ that represents the similarity between the original signal and some time-lagged versions of itself. For any $k \in \{1, \ldots, M\}$, the auto-covariance of vector z_k with j time-lags is defined as:

$$\mathscr{C}_j(z_k) := \mathbb{E}[(z_{k+j} - \mu_{k+j})(z_k - \mu_k)^\top] = \mathbb{E}[z_{k+j}z_k^\top] - \mu_{k+j}\mu_k^\top$$

for $j = 0, 1, \ldots, N-1$, where N is the maximum number of time lags and $\mu_{k+j} := \mathbb{E}[z_{k+j}]$.

Assumption 3: *Given output measurements $(y_k)_{k=1}^{\bar{k}}$, $1 \leq \bar{k} \leq M$, there exists a smoothed initial state $\hat{x}_{1|\bar{k}}$, where*

$$\hat{x}_{1|\bar{k}} := \mathbb{E}\left[x_1|(y_k)_{k=1}^{\bar{k}}\right],$$

such that if let $\hat{x}_{1|0} = \hat{x}_{1|\bar{k}}$, then the expectation of the initial state error term ε_1 will be zero.

Because for any k, the state error term ε_{k+1} is a function of ε_1 and $(\bar{w}_k)_{k=1}^M$, Assumption 3 and (3b) ensure that

$$\forall\ k, j : \mu_{k+j} = 0 \quad \Longrightarrow \quad \mathscr{C}_j(z_k) = \mathbb{E}[z_{k+j}z_k^\top].$$

Let us pick a fragment of innovations $(z_{k+1})_{k=1}^{N_z}$, where $N_z = M_e - N + 1$. The auto-covariance with j time-lags is then given by

$$\mathscr{C}_j\left((z_{k+1})_{k=1}^{N_z}\right) := \begin{bmatrix}\mathscr{C}_j(z_2) & \cdots & \mathscr{C}_j(z_{M_e-N+2})\end{bmatrix} = \mathbb{E}\begin{bmatrix}z_{2+j}z_2^\top & \cdots & z_{M_e-N+2+j}z_{M_e-N+2}^\top\end{bmatrix},$$

where M_e is the estimation data length with $N \ll M_e \leq M$.

The auto-covariance matrix (ACM) of $(z_{k+1})_{k=1}^{N_z}$ can now be defined as

$$\mathcal{R} := \begin{bmatrix}\mathscr{C}_0^\top\left((z_{k+1}^\top)_{k=1}^{N_z}\right) & \cdots & \mathscr{C}_{N-1}^\top\left((z_{k+1}^\top)_{k=1}^{N_z}\right)\end{bmatrix}.$$

For $i = 0, \ldots, M_e - N$, we also define the matrix $\mathcal{R}_i$ as

$$\mathcal{R}_i := \mathbb{E}\begin{bmatrix}z_{2+i}^\top z_{2+i} & \cdots & z_{2+i}z_{N+1+i}^\top\end{bmatrix}^\top,$$

so that

$$\mathcal{R} = \begin{bmatrix} \mathcal{R}_0 & \mathcal{R}_1 & \cdots & \mathcal{R}_{M_e-N} \end{bmatrix}.$$

Note that the auto-covariance matrix $\mathcal{R}$ is a function of Q, R and P_1; see (8) below.

Since the innovation sequence obtained from LTV systems is generally non-stationary, we are unable to calculate the sample estimate of $\mathcal{R}$ using the same method as in Rajamani and Rawlings (2009). Instead, we define the sample estimate of $\mathcal{R}$ as

$$\bar{\mathcal{R}} := \begin{bmatrix} \bar{z}_2\bar{z}_2^\top & \cdots & \bar{z}_{M_e-N+2}\bar{z}_{M_e-N+2}^\top \\ \bar{z}_3\bar{z}_2^\top & \cdots & \bar{z}_{M_e-N+3}\bar{z}_{M_e-N+2}^\top \\ \vdots & \ddots & \vdots \\ \bar{z}_{N+1}\bar{z}_2^\top & \cdots & \bar{z}_{M_e+1}\bar{z}_{M_e-N+2}^\top \end{bmatrix}$$

where $\bar{z}_k$ represents the actual innovation terms calculated by

$$\bar{z}_k := y_k - \hat{y}_{k|k-1}. \tag{4}$$

We can now define an unconstrained least squares optimization problem to estimate the true covariance:

$$(P_1^*, Q^*, R^*) := \arg\min_{\hat{P}_1, \hat{Q}, \hat{R}} \left\| \mathcal{R}(\hat{P}_1, \hat{Q}, \hat{R}) - \bar{\mathcal{R}} \right\|_F^2. \tag{5}$$

Compared to time-invariant models, the estimated error covariance $P_k := \mathbb{E}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^\top]$ for an LTV model is the solution to a time-varying Riccati equation and does not reach a steady state value. Therefore, the state and measurement noise covariance Q and R cannot be estimated from a Lyapunov equation as in Odelson et al. (2006).

4. Solving The Optimization Problem (5)

In order to apply the Kalman filter, a guess of the initial state error covariance $P_1 := P_g \succ 0$ and guessed noise covariances Q_g and R_g have to be provided, hence the sub-optimal filter gains L_k^s can be recursively obtained from the Kalman filter equations (Humpherys, Redd, and West, 2012):

$$P_{k|k-1} := A_k P_{k-1} A_k^\top + G_k Q_g G_k^\top, \tag{6a}$$

$$P_k := (I - L_k^s C_k) P_{k|k-1}, \tag{6b}$$

$$L_k^s := \left(P_{k|k-1} C_k^\top\right)\left(C_k P_{k|k-1} C_k^\top + H_k R_g H_k^\top\right)^{-1}. \tag{6c}$$

Let us start from ε_1 and consider the evolution of (3a) and (3b). The innovation sequence $(z_k)_{k=1}^{M_e}$ can be shown to be given by

$$z = \tilde{\mathcal{V}}\left(\tilde{\mathcal{E}}\varepsilon_1 + \tilde{\mathcal{G}}\tilde{w}\right) + \tilde{\mathcal{H}}\tilde{v}, \tag{7}$$

where

$$z := \begin{bmatrix} z_2^\top & z_3^\top & \cdots & z_{M_e}^\top & z_{M_e+1}^\top \end{bmatrix}^\top, \quad \tilde{w} := \begin{bmatrix} \bar{w}_1^\top & \bar{w}_2^\top & \cdots & \bar{w}_{M_e-1}^\top & \bar{w}_{M_e}^\top \end{bmatrix}^\top,$$

$$\tilde{v} := \begin{bmatrix} v_2^\top & v_3^\top & \cdots & v_{M_e}^\top & v_{M_e+1}^\top \end{bmatrix}^\top, \quad \tilde{\mathcal{A}} := I_{nM_e} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \bigoplus_{k=2}^{M_e} \bar{A}_k & \mathbf{0} \end{bmatrix},$$

$$\tilde{\mathcal{C}} := \bigoplus_{k=2}^{M_e+1} C_k, \quad \tilde{\mathcal{V}} := \tilde{\mathcal{C}}\tilde{\mathcal{A}}^{-1}, \quad \tilde{\mathcal{G}} := \bigoplus_{k=1}^{M_e} \bar{G}_k, \quad \tilde{\mathcal{H}} := \bigoplus_{k=1}^{M_e} H_k, \quad \tilde{\mathcal{E}} := \begin{bmatrix} \bar{A}_1 \\ \mathbf{0} \end{bmatrix}.$$

It is possible to use the above expressions to show that (more details are given in Appendix A)

$$\mathcal{R}(P_1, Q, R) = \Gamma\big(I_{N_z} \otimes P_{1|0}\big)\bar{\Gamma}^\top + \Omega\big(I_{N_d} \otimes Q\big)\bar{\Omega}^\top + \Phi\big(I_{N_d} \otimes R\big)\bar{\Phi}^\top + \Psi\big(I_{N_z} \otimes R\big)\bar{\Psi}^\top, \tag{8}$$

where $N_d := \frac{(N_z+1)N_z}{2}$ and

$$\Gamma := \tilde{\mathcal{S}}\tilde{\mathcal{F}}, \quad \bar{\Gamma} := \tilde{\mathcal{S}}^d\tilde{\mathcal{F}}, \quad \Omega := \tilde{\mathcal{S}}\tilde{\mathcal{J}}, \quad \bar{\Omega} := \tilde{\mathcal{S}}^d\tilde{\mathcal{J}}, \quad \tilde{\mathcal{F}}_s := \tilde{\mathcal{V}}\tilde{\mathcal{E}}, \quad \tilde{\mathcal{F}} := I_{N_z} \otimes \tilde{\mathcal{F}}_s,$$

$$\bar{\Psi} := \bigoplus_{k=2}^{M_e-N+2} H_k \quad \Phi := \tilde{\mathcal{S}}\tilde{\mathcal{U}}, \quad \bar{\Phi} := \tilde{\mathcal{S}}^d\tilde{\mathcal{U}}, \quad \Psi := \begin{bmatrix} (\mathbf{1}_{1\times N_z} \otimes I_p)\bar{\Psi} \\ \tilde{\mathcal{P}}\tilde{\mathcal{O}} \end{bmatrix},$$

$$\tilde{\mathcal{P}}_i := \mathscr{M}_{p(i+1)+1}^{p(N-1),pM_e}, \quad \tilde{\mathcal{M}} := \mathscr{M}_1^{p,pN}, \quad \tilde{\mathcal{B}} := \tilde{\mathcal{V}}\bigoplus_{k=1}^{M_e} G_k, \quad \tilde{\mathcal{S}}_i := \mathscr{M}_{pi+1}^{pN,pM_e},$$

$$\tilde{\mathcal{J}}_i := \left(\mathscr{M}_1^{r(i+1),rM_e}\right)^\top, \quad \tilde{\mathcal{J}} := \bigoplus_{i=0}^{M_e-N} \tilde{\mathcal{B}}\tilde{\mathcal{J}}_i, \quad \tilde{\mathcal{U}}_i := \left(\mathscr{M}_1^{q(i+1),qM_e}\right)^\top,$$

$$\tilde{\mathcal{P}} := \begin{bmatrix} \tilde{\mathcal{P}}_0 & \tilde{\mathcal{P}}_1 & \cdots & \tilde{\mathcal{P}}_{M_e-N} \end{bmatrix}, \quad \tilde{\mathcal{S}} := \begin{bmatrix} \tilde{\mathcal{S}}_0 & \tilde{\mathcal{S}}_1 & \cdots & \tilde{\mathcal{S}}_{M_e-N} \end{bmatrix}, \quad \tilde{\mathcal{O}}_i := \left(\mathscr{M}_{q(i+1)+1}^{q,qM_e}\right)^\top,$$

$$\tilde{\mathcal{D}} := -\tilde{\mathcal{V}}\bigoplus_{k=1}^{M_e} A_k L_k^s H_k, \; \tilde{\mathcal{U}} := \bigoplus_{i=0}^{M_e-N} \tilde{\mathcal{D}}\tilde{\mathcal{U}}_i, \; \tilde{\mathcal{O}} := \bigoplus_{i=0}^{M_e-N} \tilde{\mathcal{D}}\tilde{\mathcal{O}}_i, \; \tilde{\mathcal{S}}^d := \bigoplus_{i=0}^{M_e-N} \tilde{\mathcal{M}}\tilde{\mathcal{S}}_i$$

In order to fit problem (5) into a standard linear least-squares formulation, matrix $\mathcal{R}$ must be vectorized, which is the column-wise stacking of a matrix into a vector. Hence, the vectorized matrix $(\mathcal{R})_s$ can be expressed as

$$(\mathcal{R})_s = (\bar{\Gamma} \otimes \Gamma)\mathcal{I}_{N_z,n}(P_{1|0})_s + (\bar{\Omega} \otimes \Omega)\mathcal{I}_{N_d,r}(Q)_s + \left[(\bar{\Phi} \otimes \Phi)\mathcal{I}_{N_d,q} + (\bar{\Psi} \otimes \Psi)\mathcal{I}_{N_z,q}\right](R)_s.$$

Considering the dimension and structure of matrices $\bar{\Gamma}$, Γ, $\bar{\Omega}$, Ω, $\bar{\Phi}$, Φ, $\bar{\Psi}$ and Ψ, calculating the Kronecker product of these matrices directly will be extremely slow and require significant amounts of computer memory. Alternatively, one could parallelize the computation of each vector $(\mathcal{R}_i)_s$ and combine them together to form the vector $(\mathcal{R})_s$. The vectorized matrix $(\mathcal{R}_i)_s$ is given by Rajamani and Rawlings (2007)

$$(\mathcal{R}_i)_s = (\bar{\Gamma}_i \otimes \Gamma_i)\mathcal{I}_{1,n}(P_{1|0})_s + (\bar{\Omega}_i \otimes \Omega_i)\mathcal{I}_{i+1,r}(Q)_s + [(\bar{\Phi}_i \otimes \Phi_i)\mathcal{I}_{i+1,q} + H_{i+2} \otimes \Psi_i](R)_s, \tag{9}$$

where $\Gamma_i := \tilde{\mathcal{S}}_i\tilde{\mathcal{F}}_s, \; \bar{\Gamma}_i := \tilde{\mathcal{M}}\Gamma_i, \; \Omega_i := \tilde{\mathcal{S}}_i\tilde{\mathcal{B}}\tilde{\mathcal{J}}_i,$

$$\bar{\Omega}_i := \tilde{\mathcal{M}}\Omega_i, \; \Phi_i := \tilde{\mathcal{S}}_i\tilde{\mathcal{D}}\tilde{\mathcal{U}}_i, \; \bar{\Phi}_i := \tilde{\mathcal{M}}\Phi_i, \; \Psi_i := \begin{bmatrix} H_{i+2}^\top & \left(\tilde{\mathcal{P}}_i\tilde{\mathcal{D}}\tilde{\mathcal{O}}_i\right)^\top \end{bmatrix}^\top.$$

Let $\bar{b} := (\bar{\mathcal{R}})_s$. The original optimization problem (5) can now be rearranged into an unconstrained least-squares problem with decision variables $(P_{1|0})_{ss}$, $(Q)_{ss}$ and $(R)_{ss}$:

$$\min_{\vartheta} \left\| \underbrace{\begin{bmatrix} \mathscr{A}_0 \\ \vdots \\ \mathscr{A}_{M_e-N} \end{bmatrix}}_{\mathscr{A}} \underbrace{\begin{bmatrix} (P_{1|0})_{ss} \\ (Q)_{ss} \\ (R)_{ss} \end{bmatrix}}_{\vartheta} - \underbrace{\begin{bmatrix} \bar{b}_0 \\ \vdots \\ \bar{b}_{M_e-N} \end{bmatrix}}_{\bar{b}} \right\|_2^2 \tag{10a}$$

where, for $i = 0, \dots, M_e - N$, $\bar{b}_i := (\bar{\mathcal{R}}_i)_s$ and

$$\mathscr{A}_i := \left[(\bar{\Gamma}_i \otimes \Gamma_i)\mathcal{I}_{1,n}\mathscr{D}_n \quad (\bar{\Omega}_i \otimes \Omega_i)\mathcal{I}_{i+1,r}\mathscr{D}_r \quad \big((\bar{\Phi}_i \otimes \Phi_i)\mathcal{I}_{i+1,q} + I_q \otimes \Psi_i \big)\mathscr{D}_q \right]. \tag{10b}$$

5. Memory Allocation for the ALS Estimation

Although we have split $(\mathcal{R})_s$ into several smaller portions $(\mathcal{R}_i)_s$ in (9), calculating each $\mathscr{A}_i$ in (10b) will exhaust large amounts of computer memory as i goes to $M_e - N$, because of the Kronecker products in (9) and (10b). Thus, in order to apply our ALS algorithm for high order systems, one has to carefully modify (9) so that the memory can be used efficiently.

5.1 *Full Matrix or Sparse Matrix?*

By default, MATLAB represents a number using floating-point in double-precision. Thus for any full matrix $B \in \Re^{r \times c}$, each entry will use 8 bytes of memory, so the matrix B requires $r \times c \times 8$ bytes of memory. For a sparse matrix with n_0 non-zero entries, MATLAB represents matrices using compressed sparse column (CSC) format, so the total memory requirement M_r is calculated by

$$M_r = \underbrace{(1+c) \times 8}_{\text{IA}} + \underbrace{n_0 \times 8}_{\text{JA}} + \underbrace{n_0 \times 8}_{\text{A}},$$

where the first index term IA represents the accumulative number of non-zero entries on each column, from left to right; the index always starts with an extra 0. The second index term JA represents the location of non-zero entries on each row. The last term A represents the corresponding non-zero entries.

Proposition 1: *When using a CSC format, for a matrix with non-zero entries greater than 50%, representing it as a sparse matrix will cost more memory than representing it as a full matrix.*

Proof. For a matrix $B \in \Re^{r \times c}$, define p_n as the percentage of non-zero entires, then

$$M_r = (1+c) \times 8 + (p_n \times r \times c) \times 16,$$

hence, we have

$$\begin{aligned} (r \times c) \times 8 &= (1+c) \times 8 + (p_n \times r \times c) \times 16 \\ p_n &= 50\% - \frac{(1+c)}{(r \times c) \times 2}, \end{aligned}$$

thus, if non-zero entries greater than 50%, save a sparse matrix will require more memory than a full matrix. □

Due to the structures of matrices Γ_i, Ω_i, Φ_i and Ψ_i in (10b), the non-zero entries of these matrices will be much greater than 50%, thus, it is sensible to save them as full matrices rather than sparse matrices.

5.2 *Modified Method 1*

The computational complexity and memory allocation can be improved by rewriting (9) as Bernstein (2009, Prop. 7.1.6)

$$\begin{aligned}(\mathcal{R}_i)_s = &\left[\sum_{j=1}^{i+1}\left(\bar{\Gamma}_i\zeta_j^{\bar{\Gamma}}\right)\otimes\left(\Gamma_i\zeta_j^{\Gamma}\right)\right]\mathscr{D}_n(P_{1|0})_{ss} + \left[\sum_{j=1}^{i+1}\left(\bar{\Omega}_i\zeta_j^{\bar{\Omega}}\right)\otimes\left(\Omega_i\zeta_j^{\Omega}\right)\right]\mathscr{D}_r(Q)_{ss}\\ &+\left[\sum_{j=1}^{i+1}\left(\bar{\Phi}_i\zeta_j^{\bar{\Phi}}\right)\otimes\left(\Phi_i\zeta_j^{\Phi}\right) + H_{i+2}\otimes\Psi_i\right]\mathscr{D}_q(R)_{ss},\end{aligned}\tag{11}$$

where $\mathcal{I}_{i+1,n}$, $\mathcal{I}_{i+1,r}$ and $\mathcal{I}_{i+1,q}$ have been decomposed as

$$\begin{aligned}&\mathcal{I}_{i+1,n} = \sum_{j=1}^{i+1}\zeta_j^{\bar{\Gamma}}\otimes\zeta_j^{\Gamma}, && \mathcal{I}_{i+1,r} = \sum_{j=1}^{i+1}\zeta_j^{\bar{\Omega}}\otimes\zeta_j^{\Omega}, && \mathcal{I}_{i+1,q} = \sum_{j=1}^{i+1}\zeta_j^{\bar{\Phi}}\otimes\zeta_j^{\Phi},\\ &\zeta_j^{\bar{\Gamma}},\ \zeta_j^{\Gamma}\in\Re^{n(i+1)\times n}, && \zeta_j^{\bar{\Omega}},\ \zeta_j^{\Omega}\in\Re^{r(i+1)\times r}, && \zeta_j^{\bar{\Phi}},\ \zeta_j^{\Phi}\in\Re^{q(i+1)\times q}.\end{aligned}$$

Because $P_{1|0}$, Q and R are all symmetric matrices, from a memory efficiency point of view, we should involve $\mathscr{D}_n$, $\mathscr{D}_r$ and $\mathscr{D}_q$ in the decomposition as well. However, we are unable to decompose any of $\mathcal{I}_{i+1,n}\mathscr{D}_n$, $\mathcal{I}_{i+1,r}\mathscr{D}_r$ or $\mathcal{I}_{i+1,q}\mathscr{D}_q$ into a sum of $i+1$ Kronecker product terms, thus involving $\mathscr{D}_n$, $\mathscr{D}_r$ and $\mathscr{D}_q$ in the decomposition using Bernstein (2009, Prop. 7.1.6) will be computationally inefficient.

5.3 *Modified Method 2*

Instead of using Bernstein (2009, Prop. 7.1.6), a sum Schur/Hadamard products could be used to improve the memory efficiency of our ALS algorithm, such as

$$\begin{aligned}(\mathcal{R}_i)_s = &\left\{\sum_{j=1}^{i+1}\left[\left(\bar{\Gamma}_i\xi_j^{\bar{\Gamma}}\right)\otimes\mathbf{1}_{pN,1}\right]\circ\left[\mathbf{1}_{p,1}\otimes\left(\Gamma_i\xi_j^{\Gamma}\right)\right] + \left[\left(\bar{\Gamma}_i\tilde{\xi}_j^{\bar{\Gamma}}\right)\otimes\mathbf{1}_{pN,1}\right]\circ\left[\mathbf{1}_{p,1}\otimes\left(\Gamma_i\tilde{\xi}_j^{\Gamma}\right)\right]\right\}(P_{1|0})_{ss}+\\ &\left\{\sum_{j=1}^{i+1}\left[\left(\bar{\Omega}_i\xi_j^{\bar{\Omega}}\right)\otimes\mathbf{1}_{pN,1}\right]\circ\left[\mathbf{1}_{p,1}\otimes\left(\Omega_i\xi_j^{\Omega}\right)\right] + \left[\left(\bar{\Omega}_i\tilde{\xi}_j^{\bar{\Omega}}\right)\otimes\mathbf{1}_{pN,1}\right]\circ\left[\mathbf{1}_{p,1}\otimes\left(\Omega_i\tilde{\xi}_j^{\Omega}\right)\right]\right\}(Q)_{ss}+\\ &\left\{\sum_{j=1}^{i+1}\left[\left(\bar{\Phi}_i\xi_j^{\bar{\Phi}}\right)\otimes\mathbf{1}_{pN,1}\right]\circ\left[\mathbf{1}_{p,1}\otimes\left(\Phi_i\xi_j^{\Phi}\right)\right] + \left[\left(\bar{\Phi}_i\tilde{\xi}_j^{\bar{\Phi}}\right)\otimes\mathbf{1}_{pN,1}\right]\circ\left[\mathbf{1}_{p,1}\otimes\left(\Phi_i\tilde{\xi}_j^{\Phi}\right)\right]+\right.\\ &\left.\left[\left(H_{i+2}\xi^{H}\right)\otimes\mathbf{1}_{pN,1}\right]\circ\left[\mathbf{1}_{p,1}\otimes\left(\Psi_i\xi^{\Psi}\right)\right] + \left[\left(H_{i+2}\tilde{\xi}^{H}\right)\otimes\mathbf{1}_{pN,1}\right]\circ\left[\mathbf{1}_{p,1}\otimes\left(\Psi_i\tilde{\xi}^{\Psi}\right)\right]\right\}(R)_{ss},\end{aligned}\tag{12}$$

where

$$\begin{aligned} &\xi_j^{\bar{\Gamma}},\ \xi_j^{\Gamma},\ \tilde{\xi}_j^{\bar{\Gamma}},\ \tilde{\xi}_j^{\Gamma} \in \Re^{n(i+1)\times \frac{n(n+1)}{2}}, && \xi_j^{\bar{\Omega}},\ \xi_j^{\Omega},\ \tilde{\xi}_j^{\bar{\Omega}},\ \tilde{\xi}_j^{\Omega} \in \Re^{r(i+1)\times \frac{r(r+1)}{2}}, \\ &\xi_j^{\bar{\Phi}},\ \xi_j^{\Phi},\ \tilde{\xi}_j^{\bar{\Phi}},\ \tilde{\xi}_j^{\Phi} \in \Re^{q(i+1)\times \frac{q(q+1)}{2}}, && \xi^{H},\ \xi^{\Psi},\ \tilde{\xi}^{H},\ \tilde{\xi}^{\Psi} \in \Re^{q\times \frac{q(q+1)}{2}}, \end{aligned}$$

are matrices that only contain zeros and ones. Note that, if one assumes that $P_{1|0}$ Q and R are diagonal matrices, then all $\tilde{\xi}$ matrices are equal to zero.

5.4 *Numerical Tests*

According to Appendix A, we have for $i = 0, \cdots, M_e - N$

$$\begin{gathered} \Omega_i \in \Re^{pN\times r(i+1)}, \quad \Phi_i \in \Re^{pN\times p(i+1)}, \quad \Gamma_i \in \Re^{pN\times n}, \quad \Psi_i \in \Re^{pN\times q}, \\ \bar{\Omega}_i \in \Re^{p\times r(i+1)}, \quad \bar{\Phi}_i \in \Re^{p\times p(i+1)}, \quad \bar{\Gamma}_i \in \Re^{p\times n}. \end{gathered}$$

Since only the size of Ω_i, $\bar{\Omega}_i$, Φ_i and $\bar{\Phi}_i$ will grow with i and, in practice $p \ll r$, we only focus on monitoring how the size of Ω_i and $\bar{\Omega}_i$ vary with i and r.

Figures 1 and 2 are the plots of memory requirements and computational time for calculating the part

$$\Omega_Q := (\bar{\Omega}_i \otimes \Omega_i)\mathcal{I}_{i+1,r}\mathcal{D}_r \tag{13}$$

in (10b), respectively, using (13), the memory efficient forms (11) and (12) with $N = 50$. All results are based on an Intel Xeon E5-2699v3 18-core CPU at 3.0 GHz and 128 GB DDR4 memory. Figure 1 illustrates that, by using the memory efficient form (11) and (12), the maximum memory requirement reduces from 84.94 GB to just 372.5 and 118.1 MB, respectively. Figure 2 shows that for a larger Q, using memory efficient methods 1 and 2 will be faster than using (13).

6. Properties of the ALS Estimate and Discussion

The correlation between $\bar{z}_k$ and $\bar{z}_{k+j}$ will keep decreasing and eventually become uncorrelated as the time lag j increases. Hence, the number of time lags N can be determined by looking at the plot of the autocorrelation function of the innovation sequence $(\bar{z}_k)_{k=1}^{M}$ against the time-lagged variable, where for all $j > N$ the correlations between $\bar{z}_k$ and $\bar{z}_{k+j}$ are negligible. As discussed in Section 3, the matrix $\mathscr{A}$ in (10a) cannot be constructed using steady state solutions from the Riccati equation; therefore, the accuracy of the estimate and computational complexity is a function of M_e. In theory, we should use all given information by letting $M_e = M$, however, if M is too large, the whole estimation process will require a lot of time and computer memory. More results and discussion about the effect of varying M_e will be given in Section 8.1.

For LTV systems, the Kalman smoother (Rauch, Tung, and Striebel, 1965) can be rearranged as a convex QP with linear constraints (1). Thus, by using the method of Lagrange multipliers, it is straightforward to prove that for a sufficiently large P_g, Assumption 3 will be fulfilled regardless the choice of covariances Q_g and R_g.

Recall the matrix $\mathscr{A}$ in the auto-covariance least squares (ALS) problem (10a):

$$\mathscr{A} = [\tilde{\mathscr{A}_1} \quad \tilde{\mathscr{A}_2} \quad \tilde{\mathscr{A}_3}] \in \Re^{p^2NN_z\times(n^2+r^2+q^2)}$$

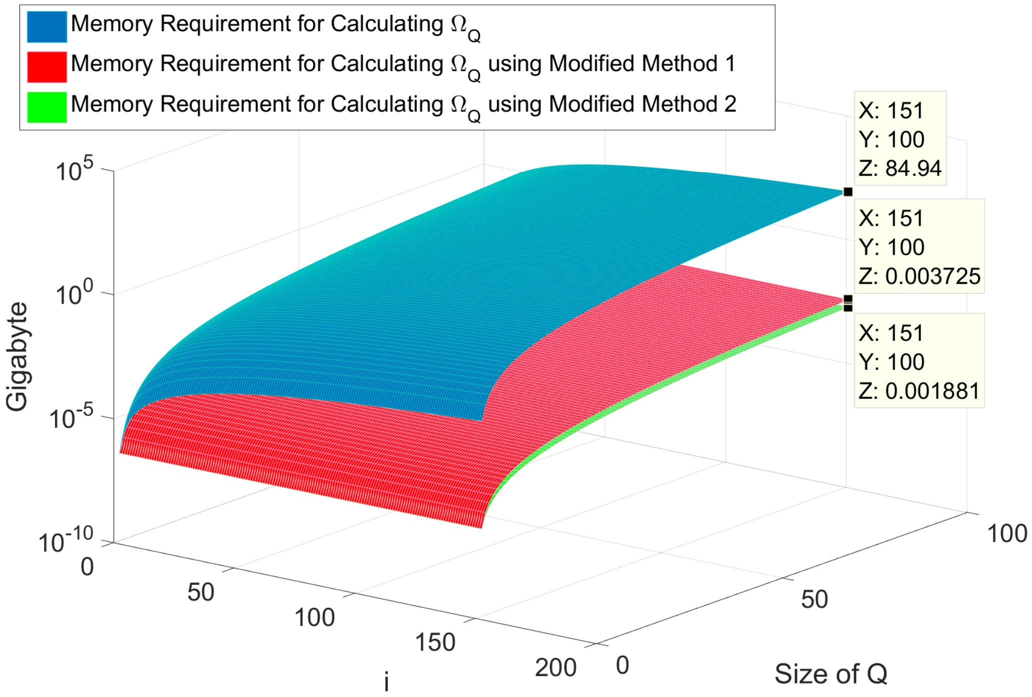

Figure 1. Memory requirement for calculating Ω_Q

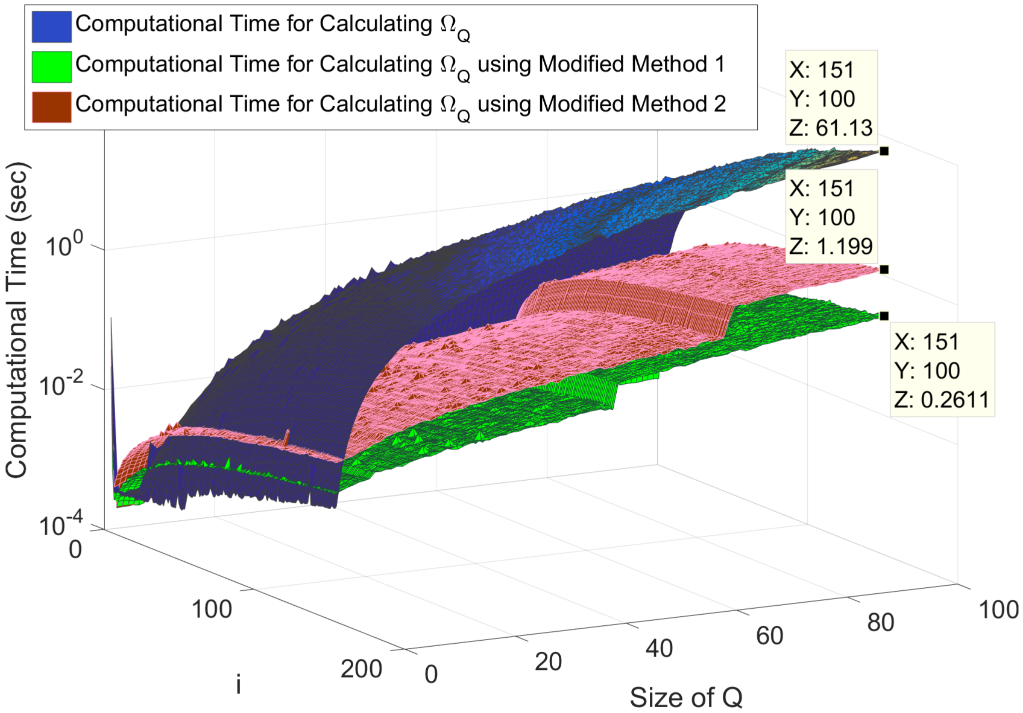

Figure 2. Computational time for calculating Ω_Q

where $\tilde{\mathscr{A}_3} := \tilde{\mathscr{A}_4} + \tilde{\mathscr{A}_5}$,

$$\tilde{\mathscr{A}_1} := (\bar{\Gamma} \otimes \Gamma)\mathcal{I}_{N_z,n}, \qquad \tilde{\mathscr{A}_2} := (\bar{\Omega} \otimes \Omega)\mathcal{I}_{N_d,r},$$
$$\tilde{\mathscr{A}_5} := (\bar{\Psi} \otimes \Psi)\mathcal{I}_{N_z,q}, \qquad \tilde{\mathscr{A}_4} := (\bar{\Phi} \otimes \Phi)\mathcal{I}_{N_d,q}.$$

Assumption 4: *The number of rows of matrix $\mathscr{A}$ is greater than the number of columns (i.e. $NN_z \geq n^2$), $\mathscr{R}(\tilde{\mathscr{A}_1}) \cap \mathscr{R}(\tilde{\mathscr{A}_2}) \cap \mathscr{R}(\tilde{\mathscr{A}_3}) = \{0\}$ and $\mathscr{R}(\tilde{\mathscr{A}_4}) \cap \mathscr{R}(\tilde{\mathscr{A}_5}) = \{0\}$.*

Assumption 5: *P_g, Q_g and R_g are positive definite, A_k is nonsingular, C_k is a full row rank matrix, G_k and H_k are full column rank matrices for all k.*

Theorem 2: *For the discrete-time LTV system* (1)*, if Assumptions 4 and 5 hold, then matrix $\mathscr{A}$ is full rank and* (5) *has a unique solution.*

Proof. See Appendix A. □

Note that, if Assumptions 4 and/or 5 are not satisfied, one has to manually check the rank of matrix $\mathscr{A}$ and make sure matrix $\mathscr{A}$ is of full column rank.

When dealing with a small sample of measurements, inappropriate choices of M_e or significant model error, the ALS estimate of the covariances may not be positive-definite and such estimates are physically meaningless (Odelson et al., 2006). This problem can be solved by adding positive-definite constraints to the linear least squares problem (10a) to get

$$\begin{aligned} \min_{\theta} & \left\| \underbrace{\begin{bmatrix} \mathscr{A}_0 \\ \vdots \\ \mathscr{A}_{M_e-N} \end{bmatrix}}_{\mathscr{A}} \underbrace{\begin{bmatrix} (\hat{P}_1)_{ss} \\ (\hat{Q})_{ss} \\ (\hat{R})_{ss} \end{bmatrix}}_{\theta} - \underbrace{\begin{bmatrix} \bar{b}_0 \\ \vdots \\ \bar{b}_{M_e-N} \end{bmatrix}}_{\bar{b}} \right\|_2^2 \\ \text{s.t.} & \quad \left(\hat{P}_1,\ \hat{Q},\ \hat{R}\right) \succ 0. \end{aligned} \tag{14}$$

Note that, in practice, G_k and H_k are usually constant ($G_k = G$ and $H_k = H$) and rarely known, hence $GQG^\top$ and $HQH^\top$ are usually estimated, thus $\hat{Q}$ and $\hat{R}$ matrices may have off-diagonal entries.

A different approach for estimating the noise covariances for time-varying and nonlinear systems is provided in Rajamani and Rawlings (2007), which assumes that there exists a k_0 with $1 < k_0 < M$ such that

$$\mathbb{E}\{\varepsilon_{k_0}\} = 0 \quad \text{and} \quad \lim_{N_k \to \infty} \left(\prod_{k=k_0}^{k_0+N_k}{}^{(b)} \bar{A}_k \right) \varepsilon_{k_0} = 0. \tag{15}$$

Once k_0 and N_k are determined, Rajamani and Rawlings (2007) constructs the ALS problem only based on $\{A_k, G_k, C_k, H_k, y_k\}_{k=k_0+N_k}^{M}$. The advantage of the algorithm in Rajamani and Rawlings (2007) is that the number of decision variables in the objective function is reduced from three to just two vectorized matrices: $(Q)_{ss}$ and $(R)_{ss}$. Hence, the computational effort of solving the auto-covariance least squares is reduced.

In fact, both statements in (15) will hold as long as (1) is uniformly detectable; however, if the available output measurements $(y_k)_{k=1}^{M}$ is limited, the statements (15) may not hold until $k_0 + N_k$ is close to M, which could significantly affect the performance of covariance identification. Even if M is long enough, in order to ignore the term $\hat{P}_{k_0}$ from the decision variables, the computational complexity to ensure

$$\left(\prod_{k=k_0}^{k_0+N_k}{}^{(b)} \bar{A}_k\right) \hat{P}_{k_0} \left(\prod_{k=k_0}^{k_0+N_k}{}^{(b)} \bar{A}_k\right)^{\top} \approx 0 \tag{16}$$

mainly depends on the value of N_k, dimension and sparsity of matrices $\bar{A}_k$.

Our method sets $k_0 = 1$ by letting $\hat{x}_{1|0} = \hat{x}_{1|M}$ and treats the term $(\hat{P}_1)_{ss}$ as a decision variable in the ALS estimation problem (14), hence our formulation does not involve any approximations, has less parameters to determine (only N, M_e, no need for N_k and k_0) and is able to estimate noise covariances Q, R as well as the initial state error covariance P_1 of a uniformly detectable LTV system. Given an appropriate choice of N_k that satisfies (16) and $N_k \leq M_e - N$, if let $k_0 = 1$, then the existing ALS algorithm provided in Rajamani and Rawlings (2007) is given by

$$\begin{aligned} \min_{(\hat{Q})_{ss},\ (\hat{R})_{ss}} & \left\| \begin{bmatrix} \hat{\mathscr{A}}_{N_k} \\ \vdots \\ \hat{\mathscr{A}}_{M_e-N} \end{bmatrix} \begin{bmatrix} (\hat{Q})_{ss} \\ (\hat{R})_{ss} \end{bmatrix} - \begin{bmatrix} \bar{b}_{N_k} \\ \vdots \\ \bar{b}_{M_e-N} \end{bmatrix} \right\|_2^2 \\ \text{s.t.} & \quad \left(\hat{Q},\ \hat{R}\right) \succ 0, \end{aligned} \tag{17}$$

where for $i = N_k, \ldots, M_e - N$,

$$\hat{\mathscr{A}_i} := \left[(\bar{\Omega}_i \otimes \Omega_i)\mathcal{I}_{i+1,r}\mathscr{D}_r \quad \left((\bar{\Phi}_i \otimes \Phi_i)\mathcal{I}_{i+1,q} + I_q \otimes \Psi_i\right)\mathscr{D}_q \right].$$

7. Nonlinear Systems

Consider the following discrete-time nonlinear state space model:

$$\begin{aligned} x_{k+1} &:= f(x_k) + G_k w_k \\ y_k &:= h(x_k) + H_k v_k \end{aligned} \tag{18}$$

where $x_k \in \mathbb{X}_k \subseteq \Re^n$ is the state, $y_k \in \Re^p$ is the measurement, w_k and v_k satisfy Assumption 1.

Assumption 6: *The discrete-time nonlinear model* (18) *is uniformly observable (Moraal and Grizzle, 1995). The nonlinear functions $f(\cdot)$ and $h(\cdot)$ are twice differentiable and there exists a stable state observer for* (18) *with nonempty feasible region.*

If we linearize the nonlinear functions $f(\cdot)$ and $h(\cdot)$ around the current estimate $\hat{x}_k$ (Simon, 2006, p. 408), i.e.

$$A_k := \left.\tfrac{\partial f(\cdot)}{\partial x_k}\right|_{x_k=\hat{x}_k}, \quad C_k := \left.\tfrac{\partial h(\cdot)}{\partial x_k}\right|_{x_k=\hat{x}_k}, \tag{19a}$$

then, A_k and C_k are linearized matrices, hence we have

$$\begin{aligned} \varepsilon_{k+1} &\approx \bar{A}_k \varepsilon_k + \bar{G}_k \bar{w}_k \\ z_k &\approx C_k \varepsilon_k + H_k v_k. \end{aligned} \tag{20}$$

Given the estimated state $\hat{x}_k$, the time-varying matrices A_k, G_k, C_k and H_k, the suboptimal Kalman filter gains L_k^s and innovation $\bar{z}_k$ can all be determined using (19), (6) and (4), respectively. If Assumption 3 is fulfilled, then the noise covariances for nonlinear systems can be estimated by establishing and solving the ALS problem (14) using all the information derived in Section 4. In the following sections, we discuss the properties of using the EKF and FIE (MHE) to estimate the true system state sequence $(x_k)_{k=1}^{M}$.

7.1 *Estimating States Using an Extended Kalman Filter*

Assumption 3 requires the estimation errors $(\varepsilon_k)_{k=1}^{M}$ to be bounded and the expectation of ε_k equals to zero for all k. Unfortunately, for nonlinear state estimation using the EKF, Assumption 3 generally does not hold, even with true noise covariances Q and R.

Because the guessed noise covariance matrices Q_g and R_g are both inaccurate (due to model mismatch), if the guessed initial state $\hat{x}_{1|0}$ is not close to the true initial state, estimation errors $(\varepsilon_k)_{k=1}^{M}$ may not be bounded for the EKF. Thus, additional conditions are needed to improve the stability and convergence of the EKF (Reif, Günther, Yaz, and Unbehauen, 1999), including observability, small initial estimation error, small noise terms and no model mismatch.

7.2 *Estimating States Using Moving Horizon Estimation*

In order to improve the state estimation for nonlinear systems, one may have to use FIE, rather than EKF to estimate the unknown system states. Because the EKF equations can be derived by minimizing the same objective function as FIE by a single Newton step (Humpherys et al., 2012), due to optimality, the FIE will provide smaller error norms $\|\varepsilon_k\|_{k=1}^{M}$ than the EKF.

The states $(\hat{x}_{k|\bar{k}})_{k=1}^{\bar{k}}$ for (18) can be determined by recursively solving the FIE, for $\bar{k} = 2, \ldots, M$:

$$X_{1,\bar{k}}^{*} := \arg\min_{X_{1,\bar{k}}} \|\varepsilon_1\|_{P_g^{-1}}^2 + \sum_{k=1}^{\bar{k}-1} \|w_k\|_{Q_g^{-1}}^2 + \sum_{k=1}^{\bar{k}} \|v_k\|_{R_g^{-1}}^2 \tag{21a}$$

$$\text{s.t.} \quad x_{k+1} = f(x_k) + G_k w_k, \ k = 1, \ldots, \bar{k}-1 \tag{21b}$$

$$y_k = h(x_k) + H_k v_k, \ k = 1, \ldots, \bar{k} \tag{21c}$$

$$x_k \in \mathbb{X}_k, \ k = 1, \ldots, \bar{k} \tag{21d}$$

where $X_{1,\bar{k}} := \begin{bmatrix} x_1^\top & v_1^\top & w_1^\top & x_2^\top & \cdots & w_{\bar{k}-1}^\top & x_{\bar{k}}^\top & v_{\bar{k}}^\top \end{bmatrix}^\top$ and the optimal of $X_{\bar{k}}$ is given by

$$X_{1,\bar{k}}^{*} =: \begin{bmatrix} \hat{x}_{1|\bar{k}}^\top & \hat{v}_{1|\bar{k}}^\top & \hat{w}_{1|\bar{k}}^\top & \cdots & \hat{w}_{\bar{k}-1|\bar{k}}^\top & \hat{x}_{\bar{k}}^\top & \hat{v}_{\bar{k}}^\top \end{bmatrix}^\top ;$$

$(\hat{x}_{k|\bar{k}})_{k=1}^{\bar{k}}$, $(\hat{w}_{k|\bar{k}})_{k=1}^{\bar{k}-1}$ and $(\hat{v}_{k|\bar{k}})_{k=1}^{\bar{k}}$ are the estimates of the state and system noise sequence $(x_k)_{k=1}^{\bar{k}}$, $(w_k)_{k=1}^{\bar{k}-1}$ and $(v_k)_{k=1}^{\bar{k}}$, respectively, given outputs.

For FIE, the problem size grows with time as the estimator processes more data; as a result, the problem complexity scales at least linearly with M. In order to make the estimation problem tractable, we need to limit the problem size by using MHE (Rawlings and Mayne, 2009), such that

$$\Theta_{\bar{k}}^{*} = \min_{X_{k_s,\bar{k}}} \|\varepsilon_{k_s}\|_{P_{k_s}^{-1}}^2 + \sum_{k=k_s}^{\bar{k}-1} \|w_k\|_{Q_g^{-1}}^2 + \sum_{k=k_s+1}^{\bar{k}} \|v_k\|_{R_g^{-1}}^2 \tag{22a}$$

$$\text{s.t.} \quad x_{k+1} = f(x_k) + G_k w_k, \ k = k_s, \ldots, \bar{k}-1 \tag{22b}$$

$$y_k = h(x_k) + H_k v_k, \ k = k_s, \ldots, \bar{k} \tag{22c}$$

$$x_k \in \mathbb{X}_k, \ k = k_s, \ldots, \bar{k} \tag{22d}$$

Algorithm 1 Noise Covariance Estimation Algorithm using ALS combined with MHE

```
Given $\bar{k}$, $\bar{k} \leq M$, $(y_k)_k^{\bar{k}}$, $x_{1|0}$, $P_g$, $Q_g$, $R_g$, determine the smoothed initial state $\hat{x}_{1|\bar{k}}$ by recursively calculating (22), for $k = 1, \cdots, \bar{k}$.
Set $x_{1|0} = \hat{x}_{1|\bar{k}}$, then recursively determine the filtered states $(\hat{x}_k)_k^M$ using the MHE (22) and the algorithm provided in Rao et al. (2003).
Calculate $(A_k,\ G_k,\ H_k,\ C_k)_{k=1}^M$ using (19) and $(\hat{x}_k)_k^M$.
Calculate the innovation sequence $(z_k)_{k=1}^M$ using $(\hat{x}_k)_k^M$.
Estimate $P_{1|0}$, $Q$ and $R$ using the ALS algorithm (14).
```

where $k_s := \max\{\bar{k} - N_l, 0\} + 1$; N_l is a positive integer called the horizon length and $N_l \leq M$. The MHE is equivalent to FIE if $\bar{k} = N_l$; for $\bar{k} \leq N_l$, $\hat{x}_1 := \tilde{x}_1$ and $P_{k_s} := P_g$. $\varepsilon_{k_s}^\top P_{k_s}^{-1} \varepsilon_{k_s}$ is called arrival cost.

A stability analysis-based algorithm for determining the arrival cost without any linearization or approximation is given in Rao, Rawlings, and Mayne (2003, p. 252), which establishes an accompanying MHE problem (Rao et al., 2003, p. 252):

$$
\begin{aligned}
\Theta_{\bar{k}}^{0*} = \min_{X^0_{k_s,\bar{k}}} + \sum_{k=k_s}^{\bar{k}-1} \|w_k\|^2_{Q_g^{-1}} + \sum_{k=k_s+1}^{\bar{k}} \|v_k\|^2_{R_g^{-1}} \\
\text{s.t.} \quad (22\text{b}),\ (22\text{c}),\ (22\text{d})
\end{aligned}
$$

where $X^0_{k_s,\bar{k}} := X_{k_s,\bar{k}}$ and the optimal of $X^0_{\bar{k}}$ is given by

$$
X^{0*}_{1,\bar{k}} =: \begin{bmatrix} \hat{x}^{0\top}_{1|\bar{k}} & \hat{v}^{0\top}_{1|\bar{k}} & \hat{w}^{0\top}_{1|\bar{k}} & \cdots & \hat{w}^{0\top}_{\bar{k}-1|\bar{k}} & \hat{x}^{0\top}_{\bar{k}} & \hat{v}^{0\top}_{\bar{k}} \end{bmatrix}^\top .
$$

The weight P_{k_s} in the arrival cost is defined by $P_{k_s} := \gamma_{k_s} P_g$, where $\gamma_{k_s} \in [0,\ 1]$ and is determined by finding the maximum possible γ_{k_s} that satisfies the following stability condition:

$$
\gamma_{k_s}^{-1} \|\varepsilon_{k_s}\|^2_{P_g^{-1}} + \Theta^*_{\bar{k}-N_l} \leq \sum_{k=0}^{m} \Theta^{0*}_{k_s - kN_l} + \|\varepsilon_1\|^2_{P_g^{-1}},
$$

where m is the quotient of k_s divided by N_l.

Note that, for the MHE algorithm stated above, if Assumption 6 holds, $\mathbb{X}_1$ is a compact set and the disturbances w_k and v_k are all bounded, then the state estimation errors $\|x_k - \hat{x}_k\|_{k=1}^M$ are guaranteed to be bounded for all k greater than the horizon length N_l (Rao et al., 2003, Prop. 3.8). Therefore, given all necessary information, the FIE will provide a stable and more accurate state estimate compared to using an EKF. Algorithm 1 shows the procedure of estimating noise covariances using the ALS combined with the MHE.

8. Numerical Examples

We present one LTV and one nonlinear example to investigate the performance of our new ALS method. The first LTV example contains 50 states, which is used to examine the performance of our ALS formulation for a large scale system; the nonlinear example analyses the difference in performance between MHE and EKF. All results in this paper are based on the MOSEK 7.0 SDP solver.

Table 1. Noise Covariance Estimation Results using Different M_e

M_e	100	200	300	400	500
$\bar{p}_Q$ (%)	44.12	28.07	21.75	19.63	18.16
$\bar{p}_R$ (%)	72.52	28.91	52.54	57.84	59.94
T (s)	7.51	8.97	11.23	13.84	18.21

8.1 *Linear Time-Varying System*

As a renewable and sustainable energy resource, ocean wave energy farms use the Wave Energy Converters (WECs) to extract energy from oscillations of the ocean waves, thus the optimal control of WECs must require knowledge of future wave elevation. An auto-regressive (AR) based model is often used to predict ocean wave heights (Fusco, 2009). Let us consider the evolution of an AR parameter vector (the state) that follows a random walk and the output matrix C_k varies with time (Fusco, 2009):

$$\begin{aligned} x_{k+1} &:= Ax_k + w_k, \\ y_k &:= C_k x_k + v_k, \end{aligned} \tag{23}$$

where $A = I_{50}$, w_k and v_k satisfy Assumption 1 and the time-varying output matrix $C_k \in \Re^{1\times 50}$ consists of historical ocean wave measurements[1] $(\eta_k)_{k=1}^{3022}$, for which

$$C_k := \begin{bmatrix} \eta_{k+50} & \eta_{k+49} & \cdots & \eta_k \end{bmatrix}.$$

We randomly pick an initial states $x_1 \sim \mathcal{N}(0, 0.2 \times I_{50})$, then generate output measurements $(y_k)_{k=1}^{3022}$ based on noise covariances $Q = 1 \times 10^{-6} \times I_{50}$ and $R = 1 \times 10^{-6}$. By guessing an initial state error covariance $P_1 = 1000 \times I_{50}$ and the guessed initial state $x_1 \sim \mathcal{N}(0, 2 \times I_{50})$, the sub-optimal filter gains $(L_k^s)_{k=1}^{3022}$ and the state error covariance $(P_k)_{k=1}^{3022}$ can be obtained from (6) with the guessed noise covariance matrices $Q_g = 0.001 \times I_{50}$ and $R_g = 0.001$.

We re-generate output measurements using different initial state and noise sequences and repeat the simulation 200 times with $N = 20$ and $M_e = 500$. Figure 3 is a scatter plot of 200 estimates of the noise covariances $\|Q^*\|_F$ and R^*, as well as the average of all 200 estimates. Table 1 shows the mean error percentage of 200 estimations of Q and R such that

$$\bar{p}_Q := \tfrac{\bar{e}_Q}{\|Q\|_F}, \quad \bar{p}_R := \tfrac{\bar{e}_R}{\|R\|_F},$$

where

$$\bar{e}_Q := \left\| Q - \tfrac{1}{200} \textstyle\sum_{i=1}^{200} Q_i^* \right\|_F, \; \bar{e}_R := \left\| R - \tfrac{1}{200} \textstyle\sum_{i=1}^{200} R_i^* \right\|_F$$

and the average time taken T for determining the noise covariances with different choices of M_e. Note that, as M_e increases, the dimensions of matrix $\mathscr{A}$ are increasing, hence the times taken for determining the noise covariances are getting larger. The best estimate of Q^* and R^* is gained at $M_e = 200$.

[1] Recorded at Galway at 5:20 on the 10th of February 2005 with sampling frequency 2.56Hz.

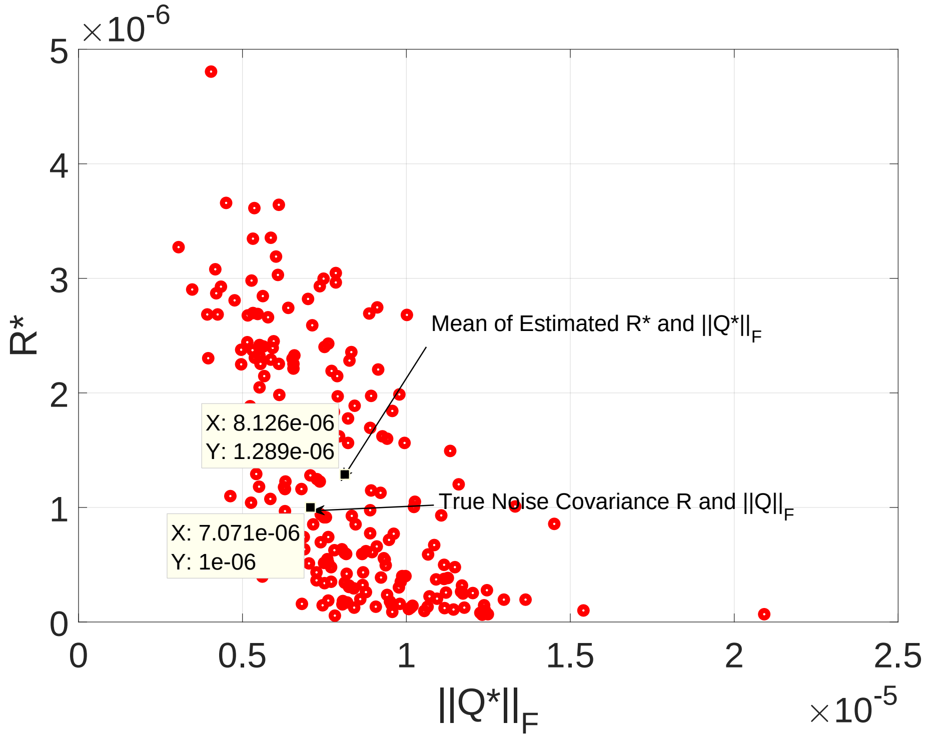

Figure 3. Estimation of Noise Covariances Q^* and R^*

8.2 *Nonlinear System*

Let us now consider tracking a sinusoidal wave whose amplitude, phase and frequency follow a random walk:

$$\underbrace{\begin{bmatrix} a_{k+1} \\ b_{k+1} \\ c_{k+1} \end{bmatrix}}_{x_{k+1}} := \underbrace{\begin{cases} +a_k \cos(c_k T_s) + b_k \sin(c_k T_s) \\ -a_k \sin(c_k T_s) + b_k \cos(c_k T_s) \\ +c_k \end{cases}}_{f(x_k)} + \underbrace{\begin{bmatrix} w_k^a \\ w_k^b \\ w_k^c \end{bmatrix}}_{w_k}$$

$$y_k := \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_k}_{h(x_k)} + v_k$$

where $T_s = 0.1\,\mathrm{s}$ is the sampling time and w_k and v_k satisfy Assumption 1. The time-varying frequency c_k is now part of the unknown state x_k, which transfers our first LTV example into a nonlinear example. We now randomly pick an initial state from a uniform distribution, such that

$$x_1 := \begin{bmatrix} x_1^1 \sim \mathcal{U}(2,4) & x_1^2 \sim \mathcal{U}(-4,-2) & x_1^3 \sim \mathcal{U}(0.3,0.7) \end{bmatrix}^\top,$$

then generate output measurements $(y_k)_{k=1}^{1000}$ based on noise covariance matrices

$$Q = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix} \times 10^{-4}, \; R = 1 \times 10^{-4}.$$

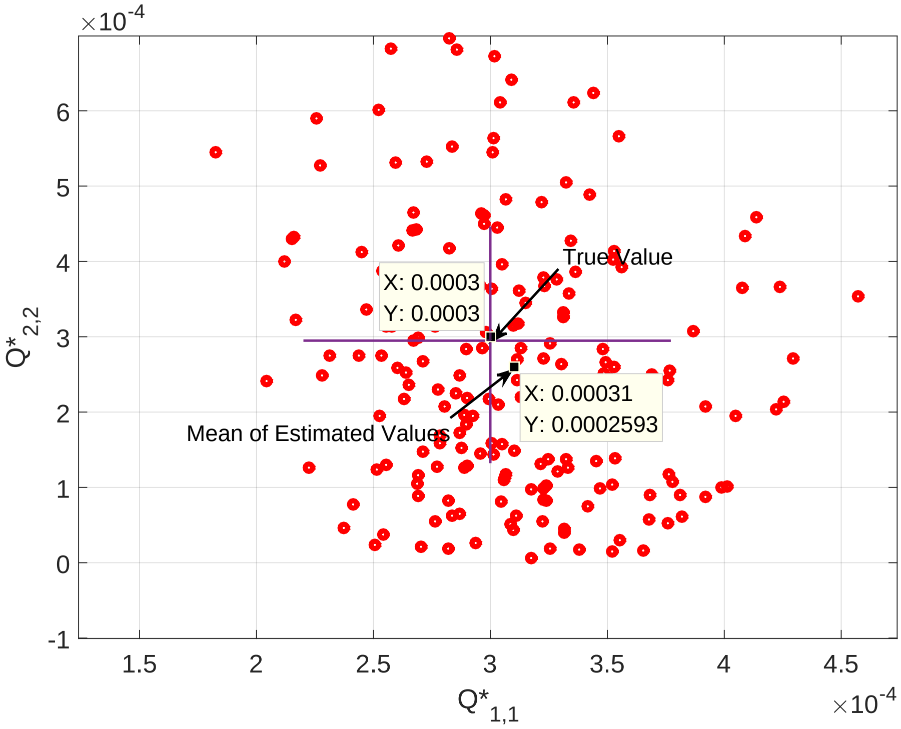

Figure 4. Estimation of noise covariances Q^* using algorithm 1

Table 2. Noise covariance estimation results using different M_e

M_e	200	400	600	800	1000
$\bar{p}_Q$ (%)	282.8	57.28	36.68	30.80	26.93
$\bar{p}_R$ (%)	11.32	7.51	4.91	2.70	3.99
T (s)	0.487	4.747	26.04	81.65	115.9

Table 3. Noise Covariance Estimation Error Analysis

	$\bar{p}_Q$(%)	$\bar{p}_R$(%)	T (s)
ALS-MHE (Section 7.2)	26.93	3.99	185.9
ALS-EKF (Section 7.1)	389×10^5	172.9	6.411

The guessed initial state error covariance is $P_{1|0} = P_g = 0.1 \times I_3$ and the guessed initial state is $\hat{x}_{1|0} = \begin{bmatrix} 15 & -15 & 15 \end{bmatrix}^\top$, the guessed noise covariance matrices are set to $Q_g = I_3$ and $R_g = 1$. Similar to the LTV example in Section 8.1, we repeat the simulation 200 times using both the MHE and EKF methods mentioned in Sections 7.1 and 7.2 with $N = 150$, $M_e = 1000$ and horizon length $H_l = 300$ for MHE.

Figures 4 to 6 are scatter plots of 200 estimates of the noise covariances Q^* and R^*, as well as the average of all 200 estimates, using ALS based Algorithm 1. Table 2 presents the mean error percentage $\bar{p}_Q$, $\bar{p}_R$ using different M_e and the average time taken T for determining the noise covariances. Table 2 clearly shows that as M_e goes to M, the error percentage $\bar{p}_Q$ and the variance of e_Q are getting smaller, but the time taken T dramatically increases. Table 3 compares the performance between the ALS-MHE and ALS-EKF method introduced in Section 7.

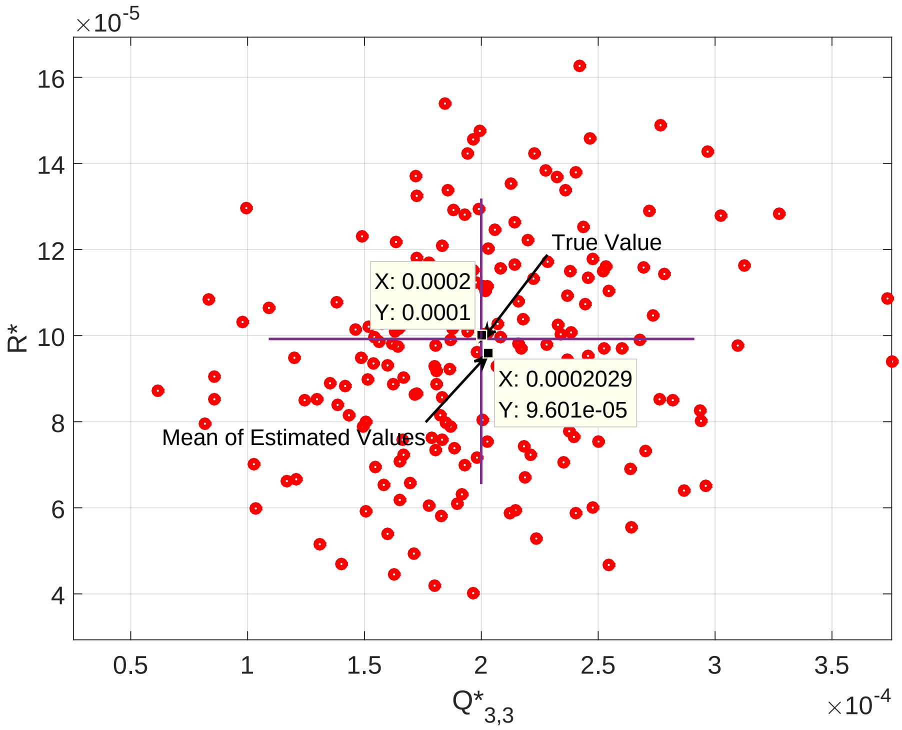

Figure 5. Estimation of noise covariances Q^* and R^* using algorithm 1

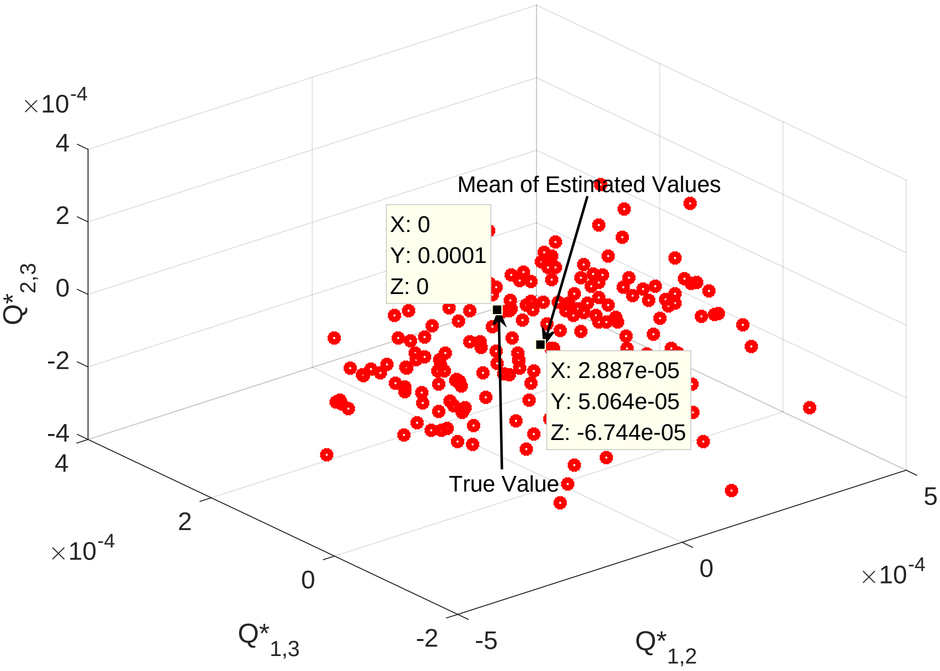

Figure 6. Estimation of noise covariances Q^* using algorithm 1

9. Conclusions

We have developed a noise covariance estimation algorithm for large scale time-varying and nonlinear systems based on a constrained (positive-definite) auto-covariance least-squares method. We used one LTV and one nonlinear example to investigate the performance of the algorithm. All numerical examples indicated that better noise covariance estimates rely on appropriate choices of algorithm parameters, such as M_e and N. The value of N can be determined by checking the autocorrelation plot; however, how to choose M_e still remains an open question. Compared with Rajamani and Rawlings (2007), our method added a useful decision variable P_1 in the optimization problem, so that our algorithm does not involve any approximations in LTV system; our new ALS algorithm has fewer parameters to determine and can provide more accurate noise covariance estimation even when the historical output measurement window is not sufficiently long for (15) to be satisfied. By rearranging the original ALS formulation, Figures 1 and 2 have shown that the new ALS algorithm is 60 times faster than the original ALS formulation (10a), using only 0.001% of the original memory. We also proposed to combine the auto-covariance least squares method with moving horizon estimation for noise covariance identification in nonlinear systems. The overall computation time of constructing and solving the optimization problem can be significantly reduced by using parallel implementations and efficient SDP solvers.

For future work, it should be possible to develop an iterative ALS method (similar to the expectation maximization algorithm), so noise covariances can be recursively identified and hopefully the results can become more accurate after several iterations. The efficiency of nonlinear noise covariance estimation can be improved by speeding up the MHE based state estimation. A possible approach is given in Rawlings and Mayne (2009, p. 350), which combines the MHE with a particle filter (Moral, 1996). The MHE has good estimation accuracy but is quite slow, whereas the particle filter can provide fast estimation but poor robustness for unmodeled disturbances and poor accuracy for higher-order systems. One of the possible ways to combine advantages of both MHE and particle filtering is given in Rawlings and Mayne (2009, p. 350), which uses particle filter to perform fast online state estimation while a computationally expensive MHE is underway. As soon as the MHE has finished optimization, particle filter samples will be located/relocated based on the MHE results, so that particle filter estimates can be recovered from any divergence.

Appendix A. Proof of Theorem 2

Assumption 5 ensures that $G_k Q G_k^\top \succeq 0$ in (6a), so that, by the definition of positive definite matrices, if $P_{k-1} \succ 0$, then $P_{k|k-1} \succ 0$.

$$P_{k+1} = \left(I_n - L_{k+1}C_{k+1}\right) P_{k+1|k} \left(I_n - L_{k+1}C_{k+1}\right)^\top + L_{k+1}H_{k+1}RH_{k+1}^\top {L_{k+1}}^\top,$$

ensures that if $P_{k|k-1} \succ 0$, then $P_k \succ 0$. Hence, by Assumption 5 and Bernstein (2009, Corollary 8.3.6)

$$\bar{A}_k := A_k \left(I - L_k^s C_k\right) = A_k P_k P_{k|k-1}^{-1},$$

is a full rank square matrix $\forall k$.

Assumptions 5 and Bernstein (2009, Fact 2.10.3) ensure that the suboptimal Kalman filter gain $L_k^s \in \Re^{n \times p}$ in (6c) is full rank $\forall k$. According to the definition of the permutation matrix, $\mathcal{I}$ has full column rank.

From the definition of $\mathscr{M}_l^{r,c}$, the following matrices are all full rank:

$$\begin{array}{ccc} \tilde{\mathcal{P}}_i \in \Re^{p(N-1)\times pM_e},\ \forall i; & \tilde{\mathcal{S}}_i \in \Re^{pN\times pM_e},\ \forall i; & \tilde{\mathcal{O}}_i \in \Re^{qM_e\times p},\ \forall i; \\ \tilde{\mathcal{U}}_i \in \Re^{qM_e\times p(i+1)},\ \forall i; & \tilde{\mathcal{J}}_i \in \Re^{rM_e\times r(i+1)},\ \forall i; & \tilde{\mathcal{P}} \in \Re^{p(N-1)\times pM_eN_z}; \\ \tilde{\mathcal{S}} \in \Re^{pN\times pM_eN_z}; & \tilde{\mathcal{S}}^d \in \Re^{pN_z\times pM_eN_z}. & \end{array}$$

$\text{rank}(\bar{A}_k) = n$ ensures $\text{rank}(\tilde{\mathcal{E}}) = n$. Assumption 5 and Bernstein (2009, Fact 2.11.13) will ensure the following matrices are full rank:

$$\begin{array}{c} \tilde{\mathcal{C}} \in \Re^{pM_e\times nM_e},\ \bar{\Psi} \in \Re^{pN_z\times qN_z},\ \tilde{\mathcal{A}} \in \Re^{nM_e\times nM_e} \\ \left(\bigoplus_{k=2}^{M_e+1} G_k\right) \in \Re^{nM_e\times rM_e},\ \left(\bigoplus_{k=2}^{M_e+1} A_k L_k^s H_k\right) \in \Re^{nM_e\times qM_e}. \end{array}$$

Hence, by using Bernstein (2009, Fact 2.10.3) and Bernstein (2009, Fact 7.4.23), we have the following full rank matrices:

$$\begin{array}{ccc} \tilde{\mathcal{V}} \in \Re^{pM_e\times nM_e}, & \tilde{\mathcal{B}} \in \Re^{pM_e\times rM_e}, & \tilde{\mathcal{J}} \in \Re^{pM_eN_z\times rN_d}, \\ \tilde{\mathcal{F}} \in \Re^{pM_eN_z\times nN_z}, & \tilde{\mathcal{D}} \in \Re^{pM_e\times qM_e}, & \tilde{\mathcal{U}} \in \Re^{pM_eN_z\times qN_d}. \end{array}$$

Therefore, by using Bernstein (2009, Fact 2.10.3) again, the following matrices are full rank:

$$\begin{array}{cccc} \Gamma \in \Re^{pN\times nN_z}, & \bar{\Gamma} \in \Re^{pN_z\times nN_z}, & \Omega \in \Re^{pN\times rN_d}, & \tilde{\mathcal{O}} \in \Re^{pM_eN_z\times qN_z} \\ \bar{\Omega} \in \Re^{pN_z\times rN_d}, & \Phi \in \Re^{pN\times qN_d}, & \bar{\Phi} \in \Re^{pN_z\times qN_d}, & \Psi \in \Re^{pN\times qN_z}. \end{array}$$

Finally, according to Bernstein (2009, Fact 2.10.3) and Bernstein (2009, Fact 7.4.23) we have

$$\begin{array}{cccc} \tilde{\mathscr{A}_1} \in \Re^{p^2NN_z\times n^2}, & \tilde{\mathscr{A}_2} \in \Re^{p^2NN_z\times r^2}, & \tilde{\mathscr{A}_4} \in \Re^{p^2NN_z\times q^2}, & \tilde{\mathscr{A}_5} \in \Re^{p^2NN_z\times q^2}, \\ \text{rank}(\tilde{\mathscr{A}_1}) = n^2, & \text{rank}(\tilde{\mathscr{A}_2}) = r^2, & \text{rank}(\tilde{\mathscr{A}_4}) = q^2, & \text{rank}(\tilde{\mathscr{A}_5}) = q^2. \end{array}$$

Assumption 4 and Bernstein (2009, Fact 2.11.11) will ensure that

$$\tilde{\mathscr{A}_3} := (\tilde{\mathscr{A}_4} + \tilde{\mathscr{A}_5}) \in \Re^{p^2NN_z\times q^2}, \quad \text{rank}(\tilde{\mathscr{A}_3}) = q^2.$$

Assumption 4 and Bernstein (2009, Fact 2.11.9) will ensure the rank of $\tilde{\mathscr{A}}$ equals to $n^2 + r^2 + q^2$, which is a full rank matrix.

References

B. D. O. Anderson and J. B. Moore. Detectability and stabilizability of time-varying discrete-time linear systems. *SIAM J. Control and Optimization*, 19(1):20–32, 1981.

Dennis S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory* (2$^{\text{nd}}$ *Edition*). Princeton University Press, 41 William Street, Princeton, New Jersey, U.S., 2009.

Francesco Fusco. Short-term wave forecasting as a univariate time series problem. Technical Report 2009-03, National University of Ireland, Dublin, Ireland, December 2009.

Jeffrey Humpherys, Preston Redd, and Jeremy West. A fresh look at the Kalman filter. *SIAM Review*, 54 (4):801–823, 2012.

Thomas Kailath. An innovations approach to least-squares estimation Part I: Linear filtering in additive white noise. *IEEE Transactions on Automatic Control*, 13(6):646–655, 1968.

Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics.* John Wiley & Sons, New York, U.S., 1999.

R. K. Mehra. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control*, 15(2):175–184, 1970.

R. K. Mehra. Approaches to adaptive filtering. *IEEE Transactions on Automatic Control*, 15(12):903–908, 1972.

P. E. Moraal and J. W. Grizzle. Observer design for nonlinear systems with discrete-time measurement. *IEEE Transactions on Automatic Control*, 40(3):395–404, 1995.

Pierre Del Moral. Nonlinear filtering: Interacting particle resolution. *Markov Processes and Related Fields*, 2(4):555–580, 1996.

B. J. Odelson, M. R. Rajamani, and J. B. Rawlings. A new autocovariance least squares method for estimating noise covariance. *Automatica*, 42(2):532–540, 2006.

M. R. Rajamani and J. B. Rawlings. Application of a new data-based covariance estimation technique to a nonlinear industrial blending drum. Technical Report 2007-03, University of Wisconsin-Madison, Madison, Wisconsin, U.S., September 2007.

M. R. Rajamani and J. B. Rawlings. Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. *Automatica*, 45(1):142–148, 2009.

Christopher. V. Rao, James. B. Rawlings, and David Q. Mayne. Constrained state estimation for nonlinear discrete-time system: Stability and moving horizon approximation. *IEEE Transactions on Automatic Control*, 48(2):246–258, 2003.

H. E. Rauch, F. Tung, and C. T. Striebel. AIAA journal. *Maximum Likelihood Estimates of Linear Dynamic System*, 3(80):1445–1450, 1965.

James B. Rawlings and Bhavik R. Bakshi. Particle filtering and moving horizon estimation. *Computer and Chemical Engineering*, 30(2006):1529–1541, 2006.

James B. Rawlings and David Q Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin, U.S., 2009.

Konrad Reif, Stefan Günther, Engin Yaz, and Rolf Unbehauen. Stochastic stability of the discrete-time extended Kalman filter. *IEEE Transactions on Automatic Control*, 44(4):714–728, 1999.

Dan Simon. *Optimal State Estimation: Kalman, H_∞ and Nonlinear Approaches*. John Wiley & Son, Inc., New Jersey, U.S., 2006.