Noise Covariance Identification for Filtering and Prediction

Ming Ge

Department of Electrical and Electronic Engineering
The Imperial College of Science, Technology and Medicine

This dissertation is submitted for the degree of
Doctor of Philosophy

February 2016
To my wife Ying Fan,
my parents and grandparents,
for their love, encouragement and support.
Declaration of Originality

As required by the college, I hereby confirm that this thesis is the result of my own work. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged through standard referencing practices of the discipline.

Ming Ge
February 2016
Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Ming Ge
February 2016
Acknowledgements

I would like to thank my PhD supervisor Dr Eric Kerrigan for his inspiration, guidance and patience through the course of my PhD. Eric is a tremendous supervisor, a cheerful and one of the smartest guy I know. He has given me the freedom to pursue the researching topic I am interested in, and always encourages me to become an independent scientific researcher. His great knowledge, insightful thoughts and valuable feedbacks have been inspiring me and keeping me on track of my goal. I feel honored to be his PhD student.

I would like to thank my colleagues in Eric’s research group: Paola Falugi, Shakil Ahmed, Edo Abraham, Bing Feng Ng, Juan Jerez, Andrea Suardi, Andrea Picciau, Stefano Longo, Zhe Feng, Mason Thammawichai, Luxin Zhang and Hari Vemuri for their feedback and suggestions during the weekly meeting and presentation. Particularly, I would like to thank Edo Abraham for providing real ocean elevation data and his help on ocean wave spectral and continuous-time bilinear wave prediction model, Paola Falugi for teaching me the ICLOCS and the IPOPT software and finally, Zhe Feng for providing many helpful suggestions in numerical optimization.

Words cannot express how grateful I am to my parents, grandparents, father and mother-in-law for all their love, supports and sacrifices.

Finally, I express my deepest gratitude to my beloved wife Ying Fan. Your love to me was what sustained me thus far. I love you, my darling.
Abstract

In this thesis, we introduce two different methods for determining noise covariance matrices in order to improve the stability and accuracy in state estimation and output prediction of discrete-time linear time varying (LTV) and nonlinear state space systems. The first method is based on the auto-covariance least squares (ALS) method, where the noise covariance matrices can be estimated by establishing a linear relationship between noise covariances and correlations of innovation sequence, hence solving a linear least squares problem.

For LTV systems, we propose a new ALS algorithm that does not involve any approximations in the formulation. 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, comparing to an existing method. In addition to the noise covariance estimates, our ALS algorithm can also provide the estimate of the initial state error covariance, which is required by most state estimation methods. For higher-order systems, we also provide a much faster and less memory demanding formulation by splitting large Kronecker products with sums of smaller Kronecker or Schur products.

For nonlinear systems, we have to approximate nonlinear parts as time-varying matrices by linearizing the nonlinear function around current state estimates. In addition to the extended Kalman Filter (EKF), our ALS algorithm also uses moving horizon estimation (MHE) to estimate the system state. MHE guarantees stability, is able to add state constraints and provides more accurate state estimates and local linearizations around the current state than the EKF.

The second method is based on expectation maximization (EM), where the noise covariance matrices are determined by recursively maximizing the likelihood of covariance matrices, given output measurements. In our method, the noise covariance matrices are estimated using a semi-definite programming (SDP) solver, so that the results are more accurate and guaranteed to be positive definite. We propose a new EM algorithm that, combined with MHE and full information estimation (FIE) rather than a Kalman-based filter/smoother, allows the addition of state constraints, provides stable and more accurate estimates, so that the performance of noise covariance estimation can be significantly improved.
Finally, we apply our noise covariance estimation methods to ocean wave prediction for the control of a wave energy converter (WEC), in order to approach optimal efficiency of wave energy extraction. We use a state space model representation for an autoregressive (AR) process, combined with noise covariance estimation, to simulate wave height forecasting based on data recorded at Galway Bay, Ireland. The simulation returns good wave predictions. Compared to existing wave prediction methods, our model has fewer parameters to tune and is able to provide more stable and accurate wave predictions by using a Kalman-based filter combined with the ALS or EM method.
# Table of contents

List of figures .......................................................... xvii

List of tables ........................................................... xix

Nomenclature ............................................................ xxi

1 Introduction .......................................................... 1
  1.1 Motivation ........................................................ 1
  1.2 Why Noise Covariance Identification? ......................... 2
  1.3 Contributions ..................................................... 3
  1.4 Thesis Layout ..................................................... 5

2 Review on State Estimation and Noise Covariance Identification 7
  2.1 Kalman Filter ...................................................... 7
  2.2 Kalman Smoother .................................................. 11
  2.3 Steady State Kalman Filter ..................................... 13
  2.4 Extended Kalman Filter ........................................ 14
  2.5 Full Information Estimation ................................... 15
  2.6 Moving Horizon Estimation .................................... 21
  2.7 Relations Between the Kalman Filter/Smoother and Full Information Estimation 23
  2.8 Auto-covariance Least Squares for LTI Systems ............... 35
  2.9 Expectation Maximization for Linear Systems ................ 38
  2.10 Conclusions ..................................................... 42

3 Noise Covariance Identification for Linear Time-Varying Systems using the
   Auto-covariance Least Squares Method 45
  3.1 Auto-covariance Least Squares for LTV Systems ............... 46
  3.2 Solving the Optimization Problem ............................. 49
  3.3 Properties of the ALS Estimate and Discussion ............... 51
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4 Memory Allocation for the ALS Estimation</td>
<td>54</td>
</tr>
<tr>
<td>3.4.1 Full Matrix or Sparse Matrix?</td>
<td>55</td>
</tr>
<tr>
<td>3.4.2 Modified Method 1</td>
<td>56</td>
</tr>
<tr>
<td>3.4.3 Modified Method 2</td>
<td>56</td>
</tr>
<tr>
<td>3.4.4 Numerical Tests</td>
<td>57</td>
</tr>
<tr>
<td>3.5 Numerical Examples</td>
<td>57</td>
</tr>
<tr>
<td>3.5.1 Linear Time-Varying System 1</td>
<td>59</td>
</tr>
<tr>
<td>3.5.2 Linear Time-Varying System 2</td>
<td>64</td>
</tr>
<tr>
<td>3.6 Conclusions</td>
<td>69</td>
</tr>
<tr>
<td>4 Noise Covariance Identification for Nonlinear Systems using Auto-covariance Least Squares and Expectation Maximization Method</td>
<td>73</td>
</tr>
<tr>
<td>4.1 Auto-covariance Least Squares Method with the Moving Horizon Estimation</td>
<td>74</td>
</tr>
<tr>
<td>4.1.1 Estimating States Using an Extended Kalman Filter</td>
<td>75</td>
</tr>
<tr>
<td>4.1.2 Estimating States Using Moving Horizon Estimation</td>
<td>75</td>
</tr>
<tr>
<td>4.2 Expectation Maximization Method with the Full Information Estimation</td>
<td>75</td>
</tr>
<tr>
<td>4.3 Numerical Examples</td>
<td>82</td>
</tr>
<tr>
<td>4.3.1 Nonlinear System 1</td>
<td>82</td>
</tr>
<tr>
<td>4.3.2 Nonlinear System 2</td>
<td>84</td>
</tr>
<tr>
<td>4.3.3 Nonlinear System 3</td>
<td>90</td>
</tr>
<tr>
<td>4.4 Conclusions</td>
<td>96</td>
</tr>
<tr>
<td>5 Ocean Wave Forecasting</td>
<td>99</td>
</tr>
<tr>
<td>5.1 Ocean Wave Spectrum, Cut-off Frequency and Prediction Horizon</td>
<td>99</td>
</tr>
<tr>
<td>5.2 Autoregressive Prediction Model</td>
<td>102</td>
</tr>
<tr>
<td>5.3 Performance of Wave Prediction</td>
<td>105</td>
</tr>
<tr>
<td>5.3.1 Cut-off Frequency and the Length of the AR Parameter Vector</td>
<td>105</td>
</tr>
<tr>
<td>5.3.2 Robustness of AR Prediction Models</td>
<td>105</td>
</tr>
<tr>
<td>5.3.3 Which AR Prediction Model is Better?</td>
<td>107</td>
</tr>
<tr>
<td>5.3.4 Prediction Performance Comparison</td>
<td>109</td>
</tr>
<tr>
<td>5.4 Conclusions</td>
<td>111</td>
</tr>
<tr>
<td>6 Conclusions</td>
<td>113</td>
</tr>
<tr>
<td>6.1 Contributions</td>
<td>113</td>
</tr>
<tr>
<td>6.2 Future Works</td>
<td>116</td>
</tr>
<tr>
<td>References</td>
<td>119</td>
</tr>
</tbody>
</table>
### Table of contents

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Proof of non-singularity of matrix $\mathcal{H}_{S_k}$ in Theorem 2.4</td>
<td>123</td>
</tr>
<tr>
<td>B</td>
<td>Proof of monotonicity of $\mathcal{L}_\ell(\mathcal{G}_i</td>
<td>Y_{1:M})$ in the EM method</td>
</tr>
<tr>
<td>C</td>
<td>“Lag-n” smoothed state error covariance</td>
<td>129</td>
</tr>
<tr>
<td>D</td>
<td>Derivation of (3.5) and (3.6)</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>D.1 Derivation of (3.5)</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>D.2 Derivation of (3.6)</td>
<td>132</td>
</tr>
<tr>
<td>E</td>
<td>Proof of Proposition 5.1</td>
<td>135</td>
</tr>
</tbody>
</table>
List of figures

2.1 Flow chart of moving horizon estimation .......................... 24
3.1 Memory requirement for calculating $\Omega_Q$ .......................... 58
3.2 Computational time for calculating $\Omega_Q$ .......................... 58
3.3 Time taken vs. the number of CPU cores for setting up the ALS algorithm .... 59
3.4 Eigenvalues of matrix $\prod_{k=2}^{2+N_k(b)} \bar{A}_k$ as $N_k$ increases .......... 60
3.5 Autocorrelation function of the innovation sequence ............... 61
3.6 Estimation of noise covariances $Q^*$ ............................... 62
3.7 Estimation of noise covariances $Q^*$ and $R^*$ .......................... 62
3.8 Output measurements $y_k$ ............................................. 65
3.9 Eigenvalues of matrix $\prod_{k=2}^{2+N_k(b)} \bar{A}_k$ as $N_k$ increases .......... 66
3.10 Estimation of noise covariances $Q^*$ ............................... 67
3.11 Estimation of noise covariances $Q^*$ and $R^*$ .......................... 67
3.12 Estimation of noise covariances $Q^*$ ............................... 68
3.13 State estimation using the existing ALS algorithm ............... 70
4.1 Time taken vs. the number of CPU cores for computing the ALS algorithm 83
4.2 Estimation of noise covariances $Q^*$ and $R^*$ .......................... 84
4.3 Estimation of noise covariances $Q^*$ using algorithm 4.1 .............. 86
4.4 Estimation of noise covariances $Q^*$ and $R^*$ using algorithm 4.1 ........ 86
4.5 Estimation of noise covariances $Q^*$ using algorithm 4.1 .............. 87
4.6 Noise covariance estimation using algorithm 4.2 .................... 87
4.7 Noise covariance estimation using fast algorithm 4.2 ............... 88
4.8 Noise covariance estimation using the EM+EKS ..................... 88
4.9 Log-likelihood function $\Omega$ versus number of iterations ............. 89
4.10 Estimates of the state trajectory using $Q_1$ and $R_1$ ................... 93
4.11 Estimates of the state noise covariance ........................... 93
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13</td>
<td>Estimates of output noise covariance</td>
<td>94</td>
</tr>
<tr>
<td>4.14</td>
<td>Log-likelihood function $\Omega$ versus number of iterations</td>
<td>95</td>
</tr>
<tr>
<td>5.1</td>
<td>Spectrum of measured ocean waves</td>
<td>100</td>
</tr>
<tr>
<td>5.2</td>
<td>Smoothed ocean wave measurements (cut-off frequency: 1.3 rad/s)</td>
<td>101</td>
</tr>
<tr>
<td>5.3</td>
<td>Percentage of fitness vs cut-off frequency and size of AR parameters using (5.2)</td>
<td>106</td>
</tr>
<tr>
<td>5.4</td>
<td>Percentage of fitness vs cut-off frequency and size of AR parameters using (5.6)</td>
<td>106</td>
</tr>
<tr>
<td>5.5</td>
<td>Percentage of fitness vs $m$ for estimating the initial state $\hat{x}_m$</td>
<td>107</td>
</tr>
<tr>
<td>5.6</td>
<td>Robustness tests for AR prediction with models</td>
<td>108</td>
</tr>
<tr>
<td>5.7</td>
<td>Log-likelihood function $\Omega$ versus number of iterations</td>
<td>110</td>
</tr>
<tr>
<td>5.8</td>
<td>Eigenvalues of matrix $\prod_{k=2}^{2+N_k(b)} \tilde{A}_k$ as $N_k$ increases</td>
<td>110</td>
</tr>
<tr>
<td>5.9</td>
<td>Wave prediction performances for different AR prediction models</td>
<td>111</td>
</tr>
<tr>
<td>5.10</td>
<td>Wave prediction performances for different AR prediction models</td>
<td>112</td>
</tr>
<tr>
<td>6.1</td>
<td>Two wave forecasting approaches [15]</td>
<td>118</td>
</tr>
</tbody>
</table>
### List of tables

3.1 Noise covariance estimation results using different $M_e$ ................. 63  
3.2 Performance comparison between two ALS algorithms .................. 63  
3.3 Comparison amongst different noise covariance estimation algorithms .. 64  
3.4 Noise covariance estimation results using different $M_e$ ................. 68  
3.5 Performance comparison between two ALS algorithms .................. 69  
3.6 Comparison of different noise covariance estimation algorithms .......... 70  
4.1 Noise covariance estimation results using different $M_e$ ................. 89  
4.2 Noise covariance estimation results ........................................... 89  
4.3 Noise covariance estimation results ........................................... 91  
4.4 Robustness tests for EM+EKS based noise covariance estimation ....... 92  
4.5 Noise covariance estimation results ........................................... 96  
4.6 Noise covariance estimation results ........................................... 97  
5.1 Ocean wave prediction results with model (5.2) .......................... 108  
5.2 Ocean wave prediction results with model (5.6) .......................... 109  
6.1 Advantages and disadvantages of MHE and particle filter ............... 118
Nomenclature

Roman Symbols

$\cdot^\dagger$ The Moore-Penrose generalized inverse of a matrix, such that $XX^\dagger X = X$. If the matrix is nonsingular, then $(\cdot)^\dagger = (\cdot)^{-1}$

$\cdot_s$ The column-wise vectorization of a matrix, such that $(ABC)_s = (C^T \otimes A)(B)_s$

$\cdot_{ss}$ The column-wise stacked lower triangular elements of a symmetric matrix, such that $(\hat{Q})_{ss} = \mathcal{D}_r(\hat{Q})_{ss}$

$\bigoplus$ The matrix direct sum, such that

$$
\bigoplus_{k=1}^{N} G_k := \text{diag}(G_1, \ldots, G_N) = \begin{bmatrix}
G_1 & \ldots & 0 \\
\vdots & \ddots & \vdots \\
0 & \ldots & G_N
\end{bmatrix},
$$

where $G_k$ can be any size

$0_{r,c}$ An $r \times c$ matrix with all entries equal to zero, $0_r = 0_{r,r}$

$1_{r,c}$ An $r \times c$ matrix with all entries equal to one, $1_r = 1_{r,r}$

$\delta(x - m)$ The Dirac delta function, such that

$$
\delta(x - m) = \begin{cases}
+\infty, & x = m \\
0, & x \neq m
\end{cases}
$$

and

$$
\int_{-\infty}^{\infty} \delta(x - m) dx = 1
$$

log$\cdot$ The natural logarithm function of a random variable

log $|\cdot|$ The natural logarithm of the determinant of a squared matrix or ("logdet")

$\mathbb{C}\{X,Y\}$ The covariance between two joint distributed random variables $X$ and $Y$ is defined as $\mathbb{C}\{X,Y\} = \mathbb{E}\{(X - \mathbb{E}\{X\})(Y - \mathbb{E}\{Y\})\}$
$E\{X\}$ If the probability distribution of a random variable $X$ admits a probability density function $p(X)$, then $E\{X\}$ denotes the expected value of $X$, such that

$$E\{X\} = \int_{-\infty}^{\infty} Xp(X)dX$$

$I_{N,q}$ An $(qN)^2 \times q^2$ permutation matrix containing only zeros and ones so that $(I_N \otimes R)_s = I_{N,q}(R)_s$

$L(\theta \mid y)$ Given outcomes $y$ of a statistical model, $L(\theta \mid y)$ denotes the likelihood function of parameters $\theta$ in such a model, which is equal to the probability density function of those outcomes $y$ given parameter $\theta$, that is $L(\theta \mid y) = p(y \mid \theta)$

$L_\ell(\cdot)$ The logarithm likelihood function of parameters of a statistical model $L_\ell(\cdot) = \log(L(\cdot))$

$N(\mu, P)$ Normal distribution with mean $\mu$ and covariance matrix $P$

$\mathcal{P}_r$ An $r^2 \times \frac{r(r+1)}{2}$ full column rank duplication matrix, which contains only zeros and ones

$\mathcal{H}(f(x))$ the Jacobian matrix of a scalar-valued function $f(x)$ respect to $x$

$\mathcal{J}(f(x))$ the Jacobian matrix of a vector-valued function $f(x)$ respect to $x$

$\mathcal{M}_{IC}^r$ An $r \times c$ auxiliary matrix containing only zeros and ones

$$\mathcal{M}_{IC}^r := \begin{bmatrix} 0_{r\times(l-1)} & I_r & 0_{r\times(c-r-l+1)} \\ \end{bmatrix}$$

$\mathcal{R}(A)$ The range or column space of a matrix $A \in \mathbb{R}^{n \times m}$, i.e. $\mathcal{R}(A) := \{Ax : x \in \mathbb{R}^m\}$

$|\cdot|$ or $\det(\cdot)$ The determinant of a squared matrix

$\otimes$ The standard Kronecker product, if matrices $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{p \times q}$, then

$$A \otimes B := \begin{bmatrix} A_{\{1,1\}}B & A_{\{1,2\}}B & \cdots & A_{\{1,m\}}B \\
\vdots & \vdots & \ddots & \vdots \\
A_{\{n,1\}}B & A_{\{n,2\}}B & \cdots & A_{\{n,m\}}B \\
\end{bmatrix} \in \mathbb{R}^{np \times mq}.$$

$\|\cdot\|$ Euclidean norm of a vector

$\|\cdot\|_F$ Frobenius norm of a matrix
∥x∥₂² Linear least squares (LLS) of vector x, such that ∥x∥₂² = ∥x∥₂²
∥x∥₀ Weighted least squares of vector x, which is equal to xᵀWx
tr(·) The trace of a square matrix and ∥x∥₂² = tr(Axxᵀ)
A ◦ B The Schur or Hadamard product of matrices A and B, if matrices A ∈ ℜⁿˣᵐ and B ∈ ℜⁿˣᵐ, then (A ◦ B) ∈ ℜⁿˣᵐ is the entry-wise product of two matrices A and B, such that
(A ◦ B)_{i,j} := A_{i,j} × B_{i,j} = (B ◦ A).
Iₙ An n × n identity matrix
p(X) The marginal probability density function of random variable X, such that
p(X) = ∫₀⁺∞ p(X,Y)dY
p(X,Y) Given two continuous random variables X and Y, p(X,Y) denotes the joint probability density function of X and Y, which is the probability density function of both events occurring at the same time
p(X | Y) The conditional probability density function of event X given Y such that p(X | Y) = p(X,Y) / p(Y); for a well-defined p(X | Y), we assume that the marginal probability density function of Y, p(Y) is non-zero
P ≪ 0 P is a negative-definite symmetric matrix
P ≫ 0 P is a positive-definite symmetric matrix
P ≳ 0 P is a positive-semidefinite symmetric matrix
x ∝ y The variable x is directly proportional to the variable y (the ratio \( \frac{x}{y} \) is constant)
x_a:b A sequence of vectors xᵢ from a to b, a ≤ b, such that x_a:b := (x_k)_{k=a}^b = (x_a, ..., x_b)
y_a:b A sequence of vectors yᵢ from a to b, a ≤ b, such that y_a:b := (y_k)_{k=a}^b = (y_a, ..., y_b)
Πᵏ⁺ᴺ⁺(b)ₖ=₀ A_k For a sequence of squared matrices (A_k)ₖ=₁, the notation Πⁿ⁺ᴺ⁺(b)ₖ=₀ A_k with m₀ ≤ M and m₁ ≤ M represents backwards matrix multiplication of (A_k)ₖ=m₀, i.e
\[
Πⁿ⁺ᴺ⁺(b)ₖ=₀ A_k := \begin{cases} 
A_{m₁} × A_{m₁−1} × ⋯ × A_{m₀} & \text{if } m₁ > m₀ \\
A_{m₁} & \text{if } m₁ = m₀ \\
I & \text{if } m₁ < m₀ 
\end{cases}
\]
Chapter 1

Introduction

I leave no trace of wings in the air, but I am glad I have had my flight.

Fireflies
Rabindranath Tagore

This thesis focuses on developing noise covariance estimation algorithms for improving the accuracy of state estimation and output prediction of linear time-varying (LTV) and nonlinear state space systems. This thesis also applies the noise covariance estimation algorithms to ocean wave forecasting using the autoregressive (AR) wave forecasting model for the control of wave energy converters (WECs).

1.1 Motivation

A wave energy converter (WEC) is a device for capturing wave power directly from surface waves or from pressure fluctuations below the surface [12]. In 2011, renewable energy resources, including solar, wind, geothermal as well as biofuel, contributed about 8.2% of the world’s total energy generation and the number is still increasing [46]. As a new renewable and sustainable energy resource and major competitor of offshore wind power, ocean waves have the highest energy density per unit area of all renewable resources [10]. The total wave power that can be generated around the coasts of the world is of the order of 1 TW, similar to current global electricity consumption [10, 13]. In 2008, the first wave power farm was opened in Portugal. Since then, the United States, Russia and many European countries have launched their own wave power farms to harvest energy from the ocean.

Real-time control of WECs requires knowledge of future incident wave elevations in order to approach optimal efficiency of wave energy extraction. The energy conversion in
most WECs are based either on relative oscillation between bodies or on oscillating pressure distributions within fixed or moving chambers [15]. Therefore, it is important to know the wave elevations before applying any future control techniques (i.e. latching or declutching control) in order to enable efficient power absorption over a wide range of wave conditions.

In [15], an autoregressive (AR) based ocean wave prediction model was introduced, which assumes that the current wave height depends linearly on a number of past wave heights. The linear relationship between the current and past wave heights is represented by the AR parameters, where the initial values are determined from low pass filtered historical data with a forgetting factor $\lambda$. When new data is received, the AR parameters are updated using the recursive least squares (RLS) method.

Results in [15] show that the AR-RLS model with a forgetting factor is a promising way of forecasting ocean wave elevations. One main problem of the prediction model is that, if the measurements do not add new information to the system, then after a certain time, the RLS gains may grow without bound. Hence, the estimated AR parameters can experience a very large growth, known as the phenomenon of blow-up [15]. Another problem is that there is no method for determining the value of the forgetting factor $\lambda$; we have to tune $\lambda$ based on historical data and assume this will not change as new information comes in.

In order to overcome disadvantages of using the AR-RLS model with a forgetting factor, [15] introduced an LTV state space model representation for AR processes, where the state dynamics matrix is an identity matrix, so that the evolution of AR parameters follow a random walk. The AR parameters can then be estimated and predicted as unknown system states using the Kalman filter and predictor, respectively. The main difficulty of predicting ocean wave heights using a Kalman filter and predictor is that the initial state, corresponding error covariance, process and output noise statistics are all unknown.

1.2 Why Noise Covariance Identification?

Given the system dynamics, people often want to predict future system outputs based on current information, for example, when peaks and troughs of ocean waves are going to hit the WEC. This allows one to latch or declutch the power generator to maximize efficiency. In the military defense, we have to determine the arrival time and exact target of incoming cruise missile, so we could eliminate it as early as possible.

In order to predict a system’s outputs, we have to know the system’s internal behavior, the so-called “states”, which are usually unknown. If systems are disturbed by noise, which they usually are, then the situation will become more difficult. Estimating “hidden” states from a noise-driven state space system requires the a priori knowledge of both process and output
noise covariances as well as the initial conditions (initial state and its error covariance). For linear dynamic systems, incorrect noise statistics will not affect the unbiasedness of state estimates. However, the estimator will no longer be a best linear unbiased estimator (BLUE) of unknown states, because the variance of estimates is not minimized. For predicting output signals, a larger variance of state estimates will significantly affect the performance and time horizon of the predicted signals. For nonlinear systems, inaccurate noise covariances and/or initial conditions could bias or even cause a divergence of state estimates.

We firstly introduce an auto-covariance least squares (ALS) based noise covariance estimation method. The ALS method establishes a linear relationship between the unknown noise covariances and covariance of innovation sequence that is obtained by using guessed noise covariances. Covariances can be determined by solving a constrained (positive-definite) linear least squares problem. The main advantages of the ALS method are that (i) it could provide relatively good estimates for linear systems in a shorter period of time and (ii) it can estimate noise covariances for some nonlinear systems, where noises are not additive to the nonlinear dynamics, compared to the EM method.

In addition to the ALS method, we also introduce another noise covariance estimation method, called the expectation maximization (EM) method, which is based on recursively maximizing the likelihood of the noise covariances given output measurements. The EM method is a much better noise covariance estimator for some nonlinear systems, where noises are additive to the nonlinear dynamics, compared to the ALS method.

A different EM approach for identifying LTI model was introduced in [52]. [52] transferred the LTI model into "innovation form" that both state and output noises are replaced by the function of innovation sequence. So that instead of finding the state and output noise covariances, [52] uses EM method to identify the covariance of the innovation sequence.

Another way for estimating noise covariances in LTI systems is presented in [8] and is based on examining admissible values of the signature of the state noise covariance (that is, the number of positive, negative and zero eigenvalues) [8]. The state noise covariance is then estimated by solving a rank minimization problem subject to a steady state algebraic Lyapunov equation, positive definite constraints and a relation between the state and observed output covariance [8]. Since a steady state algebraic Lyapunov equation does not exist for LTV and nonlinear systems, the rank minimization method is only applicable to LTI systems.

1.3 Contributions

In the literature review section of this thesis, we firstly prove that the joint probability density function of a full column rank matrix multiplying with a normal distributed random vector is
directly proportional to the joint probability density function of that random vector. We then
prove the unbiasedness of full information estimation (FIE) and state conditions for FIE to be
a best linear unbiased estimator (BLUE). A similar proof can be found in [1]; we generalize
the proof based on an LTV system, where the state and output noise matrices are full column
rank, rather than identity matrices. We also provide several proofs to show the equivalence
between the FIE and Kalman filter/smooother in LTV systems, as well as the relationships
between the FIE and extended Kalman filter (EKF)/smoother in nonlinear systems.

This thesis introduces two noise covariance estimation algorithms for LTV and nonlinear
state space systems. The first algorithm is based on the ALS method. An ALS based algorithm
for estimating noise covariances for LTV and nonlinear systems was introduced in [35];
unfortunately, due to Assumption 1 in [35], the method in [35] may require much longer
historical data for estimating noise covariances; furthermore, the formulation of [35] is not
computationally or memory efficient for large order systems. By removing the Assumption 1
in [35] from the ALS formula, our algorithm does not involve any approximations to LTV
systems, which improves the estimation accuracy, has fewer parameters to determine and is
able to obtain the initial state and corresponding error covariance.

For nonlinear systems, we use moving horizon estimation (MHE) instead of the extended
Kalman filter (EKF) used in [35], hence the stability of nonlinear state estimation is guar-
anteed and the accuracy of state estimation can be significantly improved. We decompose
high dimensional Kronecker products in our ALS formulation into a sum of low dimensional
Kronecker or Schur products, so that the computational efficiency and memory allocation can
both be improved. We also prove the conditions for a unique solution of our ALS algorithm.

The second algorithm is based on the expectation maximization (EM) method, instead
of using the extended Kalman smoother (EKS) [5]. We use MHE/FIE to obtain filtered and
smoothed states, which allows adding state constraints, provides stable and more accurate
estimation, so the performance of noise covariance estimation can be significantly improved.
We also propose to use a semi-definite programming (SDP) solver to estimate the noise
covariance, so that the results are more accurate and guaranteed to be positive definite.

Finally, we apply both noise covariance estimation algorithms to ocean wave prediction
using two state space AR models. The instability issue of wave prediction in [15] is resolved
by using a corrected RLS formulation. Results show a significant improvement in wave
height forecasting compared with other methods.
1.4 Thesis Layout

In Chapter 2, we firstly provide a brief introduction of linear and nonlinear state estimation, followed by a deep exploration of their properties and relationships and finally, two different noise covariance identification techniques are quickly reviewed.

In Chapters 3 and 4, we will present an ALS-based and an EM-based noise covariance estimation algorithm, respectively. We will discuss the algorithm properties, evaluate the performance and efficiency, and compare with existing algorithms using small numerical examples.

In Chapter 5, we will apply our noise covariance estimation algorithms to ocean wave prediction using the state space AR models provided in [15] and compare their performance and efficiency.

Finally, we draw conclusions and propose avenues for further work in Chapter 6.
Chapter 2

Review on State Estimation and Noise Covariance Identification

If I have seen further it is by standing on the shoulders of Giants.

Isaac Newton

In order to estimate unknown states from a noise-driven state space system, the state estimator requires a priori knowledge of both process and output noise covariance matrices as well as the initial state error covariance. Unfortunately, noise statistics are usually unknown and have to be determined from output measurements. Hence, the accuracy of state estimation and noise covariance identification entirely depends on each other. In this chapter, we will briefly review different state estimation methods as well as state-of-the-art techniques for noise covariance identification.

2.1 Kalman Filter

The Kalman filter is the most notable and widely used state estimation algorithm of the past few decades. Given previous output measurements, the algorithm recursively estimates state variables in a noise-driven linear dynamical system by minimizing the mean-square error between the true and estimate state. In this section, we will present a classic Bayesian derivation of the Kalman filter, which follows the derivation given by [24].

Consider a discrete-time LTV system

\[
\begin{align*}
    x_{k+1} &= A_k x_k + G_k w_k \\
    y_k &= C_k x_k + H_k v_k
\end{align*}
\] (2.1)
The pair

Assumption 2.2. The noise sequences

where

Assumption 2.1. The noise sequences \((w_k)_{k=1}^{M} \) and \((v_k)_{k=1}^{M} \) are two uncorrelated random variables having Gaussian (or normal) distributions \(N(0, Q)\) and \(N(0, R)\), respectively, with zero mean and given positive-definite covariance matrices \(Q\) and \(R\).

Assumption 2.2. The pair \((A_k, C_k)\) in the LTV system (2.1) is uniformly detectable [2] for \(1 \leq k \leq M\).

In order to estimate current state \(x_{k+1}\) given output measurements \(y_{1:k+1}, 1 \leq k < M\) and covariance matrices \(Q\) and \(R\), the Kalman filter computes the conditional probability distribution \(p(x_{k+1}|y_{1:k+1})\) using the following equation

\[
p(x_{k+1}|y_{1:k+1}) = p(x_{k+1}|y_{k+1}, y_{1:k}) = \frac{p(x_{k+1}, y_{k+1}|y_{1:k})}{p(y_{k+1}|y_{1:k})}.
\]

Proposition 2.1. For LTV system (2.1), if Assumption 2.1 holds then the probability distribution \(p(x_{k+1}, y_{k+1}|y_{1:k})\) and \(p(y_{k+1}|y_{1:k})\) can be written as

\[
p(x_{k+1}, y_{k+1}|y_{1:k}) \sim N \left( \begin{bmatrix} A_k \hat{x}_k \\ C_{k+1} \hat{x}_{k+1|k} \end{bmatrix}, \begin{bmatrix} A_k P_k A_k^\top + G_k Q G_k^\top & P_{k+1|k} C_{k+1}^\top \\ C_{k+1} P_{k+1|k} C_{k+1} + H_{k+1} R H_{k+1}^\top \end{bmatrix} \right)
\]

and

\[
p(y_{k+1}|y_{1:k}) \sim N \left( C_{k+1} \hat{x}_{k+1|k}, C_{k+1} P_{k+1|k} C_{k+1} + H_{k+1} R H_{k+1}^\top \right),
\]

respectively, where \(\hat{x}_k := E\{x_k|y_{1:k}\}\), \(\hat{x}_{k+1|k} := E\{x_{k+1}|y_{1:k}\}\), \(P_k := P_{k|k} := C\{x_k|y_{1:k}\}\) and \(P_{k+1|k} := C\{x_{k+1}|y_{1:k}\}\).

Proof. For LTV system (2.1), given the first \(k + 1\) output measurements and Assumption 2.1, both \(x_{k+1}\) and \(y_{k+1}\) are normal distributed random variables. Thus by the definition of the joint and marginal probability distribution, we have

\[
p(x_{k+1}, y_{k+1}|y_{1:k}) \sim N \left( \begin{bmatrix} \mathbb{E}\{x_{k+1}|y_{1:k}\} \\ \mathbb{E}\{y_{k+1}|y_{1:k}\} \end{bmatrix}, \begin{bmatrix} C\{x_{k+1}|y_{1:k}\} & C\{x_{k+1}, y_{k+1}|y_{1:k}\} \\ C\{y_{k+1}, x_{k+1}|y_{1:k}\} & C\{y_{k+1}|y_{1:k}\} \end{bmatrix} \right)
\]

and

\[
p(y_{k+1}|y_{1:k}) \sim N(\mathbb{E}\{y_{k+1}|y_{1:k}\}, C\{y_{k+1}|y_{1:k}\}),
\]
2.1 Kalman Filter

in which

\[ \mathbb{E} \{ x_{k+1} | y_{1:k} \} = \hat{x}_{k+1 | k} = \mathbb{E} \{ A_k x_k + G_k w_k | y_{1:k} \} = A_k \hat{x}_k \]
\[ \mathbb{E} \{ y_{k+1} | y_{1:k} \} = \hat{y}_{k+1 | k} = \mathbb{E} \{ C_{k+1} \hat{x}_{k+1} + H_{k+1} v_{k+1} | y_{1:k} \} = C_{k+1} \hat{x}_{k+1 | k}. \]

By the definition of the covariance matrix we have

\[ \mathbb{C} \{ x_{k+1} | y_{1:k} \} = \mathbb{E} \{ (x_{k+1} - \mathbb{E} \{ x_{k+1} | y_{1:k} \}) (x_{k+1} - \mathbb{E} \{ x_{k+1} | y_{1:k} \})^\top \} = P_{k+1 | k} \]
\[ = A_k \mathbb{E} \{ (x_k - \hat{x}_k) (x_k - \hat{x}_k)^\top \} A_k^\top + G_k \mathbb{E} \{ w_k w_k^\top \} G_k^\top = A_k P_k A_k^\top + G_k Q G_k^\top \]
\[ \mathbb{C} \{ x_{k+1}, y_{k+1} | y_{1:k} \} = \mathbb{C} \{ x_{k+1}, x_{k+1} \}^\top = \mathbb{E} \{ (x_{k+1} - \mathbb{E} \{ x_{k+1} | y_{1:k} \}) (y_{k+1} - \mathbb{E} \{ y_{k+1} | y_{1:k} \}) \} \]
\[ = \mathbb{E} \{ (x_{k+1} - \mathbb{E} \{ x_{k+1} | y_{1:k} \}) (C_{k+1} x_{k+1} + H_{k+1} v_{k+1} - C_{k+1} \hat{x}_{k+1 | k}) \} \]
\[ = P_{k+1 | k} C_{k+1}^\top, \]
\[ \mathbb{C} \{ y_{k+1} | y_{1:k} \} = \mathbb{C} \{ y_{k+1} - \mathbb{E} \{ y_{k+1} | y_{1:k} \} \} (y_{k+1} - \mathbb{E} \{ y_{k+1} | y_{1:k} \}) \}
\[ = C_{k+1} \mathbb{E} \{ (x_{k+1} - \hat{x}_{k+1 | k}) (x_{k+1} - \hat{x}_{k+1 | k}) \} C_{k+1}^\top + H_{k+1} \mathbb{E} \{ v_{k+1} v_{k+1}^\top \} H_{k+1}^\top \]
\[ = C_{k+1} P_{k+1 | k} C_{k+1}^\top + H_{k+1} R H_{k+1}^\top. \]

Lemma 2.1. Given two random variables \( a \) and \( b \in \mathbb{R}^n \), if the joint and marginal probability distribution are defined as

\[ p(a,b) \sim \mathcal{N} \left( \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}, \begin{bmatrix} P_a & P_{ab} \\ P_{ab}^\top & P_b \end{bmatrix} \right) \quad \text{and} \quad p(b) \sim \mathcal{N} \left( \hat{b}, P_b \right), \]

respectively, then the conditional probability distribution \( p(a | b) \) is given by

\[ p(a | b) \sim \mathcal{N} \left( \hat{a} + P_{ab} P_b^{-1} (b - \hat{b}), P_a - P_{ab} P_b^{-1} P_{ab}^\top \right) \]

Proof. By the definition of conditional probability density

\[ p(a | b) = \frac{p(a,b)}{p(b)} \]

we have

\[ p(a | b) = \frac{1}{(2\pi)^{-n/2} |P_b|^{-1/2} e^{-(b - \hat{b})^\top P_b^{-1} (b - \hat{b})}} \left( 2\pi \right)^{-n/2} P_a^{-1/2} e^{-\frac{1}{2} (a - \hat{a}) \mathbb{C}^{-1} (a - \hat{a})} \]

\[ = \frac{1}{(2\pi)^{-n/2} |P_b|^{-1/2} e^{-(b - \hat{b})^\top P_b^{-1} (b - \hat{b})}} \left( 2\pi \right)^{-n/2} P_b^{-1/2} e^{-(b - \hat{b}) \mathbb{C} P_{ab} P_b^{-1} (b - \hat{b})}. \]

(2.2)
According to [6, Fact 2.14.9 & Fact 2.17.3]

\[
\begin{bmatrix}
P_a & P_{ab} \\
P_{ab}^\top & P_b
\end{bmatrix}^{-1} = \begin{bmatrix}
P_a - P_{ab}P_b^{-1}P_{ab}^\top & P_{ab}P_b^{-1}P_{ab}^\top \\
P_{ab}P_b^{-1}P_{ab}^\top & P_b
\end{bmatrix},
\]

where \( \bar{\hat{x}} \) is the Kalman filter algorithm for LTV system (2.1), where

\[
L_k = \begin{bmatrix}
P_{k+1|k}C_{k+1}^\top \\
C_{k+1}P_{k+1|k} + H_{k+1}RH_{k+1}^\top
\end{bmatrix}^{-1}.
\]

substituting (2.3) into (2.2) gives

\[
p(a|b) = (2\pi)^{-\frac{n}{2}} |\tilde{P}|^{-\frac{1}{2}} e^{-\frac{1}{2}(a-\tilde{a})^\top \tilde{P}^{-1} \left(\begin{array}{c} a-\tilde{a} \\ b-\hat{b} \end{array}\right)},
\]

where \( \tilde{P} := P_a - P_{ab}P_b^{-1}P_{ab}^\top \). Hence, we have

\[
p(a|b) = (2\pi)^{-\frac{n}{2}} |\tilde{P}|^{-\frac{1}{2}} e^{-\frac{1}{2}(a-\tilde{a})^\top \tilde{P}^{-1} \left(\begin{array}{c} a-\tilde{a} \\ b-\hat{b} \end{array}\right)},
\]

where \( \tilde{a} := \hat{a} + P_{ab}P_b^{-1}(b - \hat{b}) \). Therefore, \( p(a|b) \sim \mathcal{N}(\tilde{a}, \tilde{P}) \).

According to Lemma 2.1, we have \( p(x_{k+1}|y_{1:k+1}) \sim \mathcal{N}(\hat{x}_{k+1}, P_{k+1}) \), where

\[
\hat{x}_{k+1} = A_k \hat{x}_k + L_{k+1}(y_{k+1} - C_{k+1}A_k \hat{x}_k),
\]

\[
P_{k+1} = (I_n - L_{k+1}C_{k+1}) P_{k+1|k},
\]

\[
P_{k+1|k} = A_k P_{k|k} A_k^\top + G_k Q G_k^\top,
\]

\[
L_{k+1} = \left( P_{k+1|k}C_{k+1}^\top \right) \left( C_{k+1}P_{k+1|k}C_{k+1}^\top + H_{k+1}RH_{k+1}^\top \right)^{-1}.
\]

is the Kalman filter algorithm for LTV system (2.1), where \( L_k \) is the Kalman filter gain. For \( k = 1 \), we have

\[
\hat{x}_1 = \hat{x}_{1|0} + L_1 (y_1 - C_1 \hat{x}_{1|0}),
\]

\[
P_1 = (I_n - L_1 C_1) P_{1|0},
\]
2.2 Kalman Smoother

\[ L_1 = \left( P_{1|0} C_1^T \right) \left( C_1 P_{1|0} C_1^T + H_1 R H_1^T \right)^{-1}, \]

where \( \hat{x}_{1|0} \) and \( P_{1|0} \) are guesses of the initial state \( x_1 \) and corresponding error covariance, respectively. Rearranging (2.4a) gives

\[
\begin{align*}
  x_{k+1} - \hat{x}_{k+1} &= x_{k+1} - \hat{x}_{k+1|k} + L_{k+1} \left( C_{k+1} x_{k+1} + H_{k+1} v_{k+1} - C_{k+1} \hat{x}_{k+1|k} \right), \\
  &= (I_n + L_{k+1} C_{k+1}) (x_{k+1} - \hat{x}_{k+1|k}) + L_{k+1} H_{k+1} v_{k+1},
\end{align*}
\]

hence (2.4b) can be alternatively rewritten as [22, p. 208]

\[
P_{k+1} = (I_n - L_{k+1} C_{k+1}) P_{k+1|k} (I_n - L_{k+1} C_{k+1})^T + L_{k+1} H_{k+1} R H_{k+1}^T L_{k+1}^T, \quad (2.5)
\]

which ensures that if \( P_{k|k-1} \succ 0 \), then \( P_k \succ 0 \).

### 2.2 Kalman Smoother

If some output measurements \( (y_k)_{k=1}^M \) are given and Assumptions 2.1 holds, we are able to recursively obtain the sequence of state estimates

\[
\hat{x}_{k|M} := \mathbb{E}\{x_k|y_{1:M}\} \quad \text{for all } 1 \leq k \leq M,
\]

where the current \( \hat{x}_M \) is called the filtered state estimate and all past \( \hat{x}_{k|M}^{M-1} \) are called smoothed state estimates. Because more output measurements are involved in a smoothed state estimate \( \hat{x}_{k|M} \) compared to a filtered estimate \( \hat{x}_k \), \( k \neq M \), [41] had shown that errors in state estimation can be significantly reduced by smoothing.

An efficient “forward-backward” smoothing algorithm was introduced in [41], where the forward step obtains filtered state estimates \( \hat{x}_k \) \( k=1 \) and error covariances \( (P_k)_{k=1}^M \) using the Kalman filter. The backward step determines smoothed state estimates by computing conditional probability distribution \( p(x_k|y_{1:M}) \), where

\[
p(x_k|y_{1:M}) \sim \mathcal{N}(\mathbb{E}\{x_k|y_{1:M}\}, \mathbb{C}\{x_k|y_{1:M}\}) \quad \text{.} \quad (2.6)
\]

According to the law of total expectation and covariance [7, pp. 343-344], we have

\[
\begin{align*}
  \mathbb{E}\{x_k|y_{1:M}\} &= \mathbb{E}\{ \mathbb{E}\{x_k|x_{k+1}, y_{1:M}\}|y_{1:M} \} | y_{1:M} \}, \quad (2.7) \\
  \mathbb{C}\{x_k|y_{1:M}\} &= \mathbb{C}\{ \mathbb{C}\{x_k|x_{k+1}, y_{1:M}\}|y_{1:M} \} + \mathbb{C}\{ \mathbb{E}\{x_k|x_{k+1}, y_{1:M}\}|y_{1:M} \} | y_{1:M} \}. \quad (2.8)
\end{align*}
\]
Lemma 2.2. For LTV system (2.1), if Assumption 2.1 holds then the probability distribution \( p(x_k | x_{k+1}, y_k) \) can be written as
\[
p(x_k | x_{k+1}, y_{1:M}) \sim \mathcal{N} \left( \hat{x}_k + U_k(x_{k+1} - \hat{x}_{k+1|k}), P_k - U_k P_{k+1|k} U_k^\top \right)
\]
where
\[
U_k := P_k A_k^\top P_{k+1|k}^{-1}.
\]

Proof. Since \( x_{k+1} \) is given, output measurements \( y_{k+1:M} \) provide no further information about \( x_k \). By the definition of conditional probability distribution we have
\[
p(x_k | x_{k+1}, y_{1:M}) = p(x_k | x_{k+1}, y_{1:k}) = \frac{p(x_k, x_{k+1} | y_{1:k})}{p(x_{k+1} | y_{1:k})},
\]
where
\[
p(x_k, x_{k+1} | y_{1:k}) \sim \mathcal{N} \left( \begin{bmatrix} E\{x_k | y_{1:k}\} \\ E\{x_{k+1} | y_{1:k}\} \end{bmatrix}, \begin{bmatrix} C\{x_k, x_{k+1} | y_{1:k}\} \\ C\{x_{k+1}, x_k | y_{1:k}\} \end{bmatrix} \right),
\]
\[
p(x_{k+1} | y_{1:k}) \sim \mathcal{N} \left( E\{x_{k+1} | y_{1:k}\}, C\{x_{k+1} | y_{1:k}\} \right).
\]

Similar to the proof of Proposition 2.1 we have
\[
p(x_k, x_{k+1} | y_{1:k}) \sim \mathcal{N} \left( \begin{bmatrix} \hat{x}_k \\ \hat{x}_{k+1|k} \end{bmatrix}, \begin{bmatrix} P_k & P_k A_k^\top \\ A_k P_k & P_{k+1|k} \end{bmatrix} \right),
\]
\[
p(x_{k+1} | y_{1:k}) \sim \mathcal{N} \left( \hat{x}_{k+1|k}, P_{k+1|k} \right).
\]

Since \( U_k P_{k+1|k} = P_k A_k^\top \), by Lemma 2.1, we have
\[
p(x_k | x_{k+1}, y_{1:M}) = p(x_k | x_{k+1}, y_{1:k}) \sim \mathcal{N} \left( \hat{x}_k + U_k(x_{k+1} - \hat{x}_{k+1|k}), P_k - U_k P_{k+1|k} U_k^\top \right).
\]

Proposition 2.2. For LTV system (2.1), if Assumption 2.1 holds then the probability distribution \( p(x_k | y_{1:M}) \) can be written as
\[
p(x_k | y_{1:M}) \sim \mathcal{N} \left( \hat{x}_k + U_k(\hat{x}_{k+1|M} - \hat{x}_{k+1|k}), P_k + U_k (P_{k+1|M} - P_{k+1|k}) U_k^\top \right).
\]

Proof. By Lemma 2.2, (2.7) and (2.8) become
\[
\hat{x}_{k|M} = E\{x_k | y_{1:M}\} = E \{ \hat{x}_k + U_k(x_{k+1} - \hat{x}_{k+1|k}) | y_{1:M} \} = \hat{x}_k + U_k(\hat{x}_{k+1|M} - \hat{x}_{k+1|k}), \quad (2.9a)
\]
2.3 Steady State Kalman Filter

\[ P_{k|M} := \mathbb{E}\{x_k|y_{1:M}\} = \mathbb{E}\{P_k - U_k P_{k+1|M} U_k^T|y_{1:M}\} + \mathbb{E}\{\hat{x}_k + U_k(x_{k+1} - \hat{x}_{k+1}|M)|y_{1:M}\} \]  

(2.9b)

\[ = P_k - U_k P_{k+1|M} U_k^T + \mathbb{E}\{U_k(x_{k+1} - \hat{x}_{k+1}|M)(x_{k+1} - \hat{x}_{k+1}|M)^T U_k^T\} \]

\[ = P_k + U_k(P_{k+1|M} - P_{k+1|k}) U_k^T, \]

where \( U_k \) is the Kalman smoother gain.

Therefore, given filtered state estimates \((\hat{x}_k)_{k=1}^M\) and error covariances \((P_k)_{k=1}^M\), we have the Kalman smoother algorithm (2.9) for \( k = M - 1, \ldots, 1 \).

**Proposition 2.3.** For LTV system (2.1), if Assumption 2.1 holds and \( P_{1|0} \succ 0 \), then the smoothed state error covariance \( P_{k|M} \) is positive definite matrix.

**Proof.** Rearranging (2.9b), we have that

\[ P_{k|M} = P_k - P_k A_k^T \left( A_k P_k A_k^T + G_k Q G_k^T \right)^{-1} A_k P_k + U_k P_{k+1|M} U_k, \]

because \( A_k P_k A_k^T \succ 0 \) and \( G_k Q G_k^T \succeq 0 \), we have

\[ A_k P_k A_k^T + G_k Q G_k^T \succeq A_k P_k A_k^T \succ 0, \]

Since \( P_{1|0} \succ 0 \) and \( A_k \) is a nonsingular matrix, (2.4a) and (2.5) imply that \( P_k \succ 0 \), hence,

\[ \left( A_k P_k A_k^T + G_k Q G_k^T \right)^{-1} \preceq \left( A_k P_k A_k^T \right)^{-1} = A_{k}^{-1} P_{k}^{-1} A_{k}^{-1}. \]

Thus, for all nonzero vector \( x \), we have

\[ x^T P_k A_k^T \left( A_k P_k A_k^T + G_k Q G_k^T \right)^{-1} A_k P_k x \leq x^T P_k A_k^T A_{k}^{-1} P_{k}^{-1} A_{k}^{-1} A_k P_k x = x^T P_k x, \]

which implies that

\[ P_k - U_k P_{k+1|M} U_k^T = P_k - P_k A_k^T \left( A_k P_k A_k^T + G_k Q G_k^T \right)^{-1} A_k P_k \geq 0 \quad \text{and} \quad U_k P_{k+1|M} U_k > 0 \]

(2.10)

which implies \( P_{k|M} \) is a positive definite matrix.

### 2.3 Steady State Kalman Filter

The Kalman gain \( L_k \) is generally time-varying; however, for LTI systems, \( L_k \) is observed to converge to a constant value as \( k \to \infty \). Often, this “steady state” filter is the one that
is implemented. The Kalman gain and hence the state error covariances are limited in the
steady state filter and the effects of the initial conditions can be neglected.

Consider a discrete-time LTI system

\[ x_{k+1} = Ax_k + Gw_k, \]
\[ y_k = Cx_k + Hv_k, \]

(2.11)

where \( A \in \mathbb{R}^{n \times n} \) is the dynamics matrix, \( C \in \mathbb{R}^{p \times n} \) is the sensor matrix, \( G \in \mathbb{R}^{n \times r} \) and \( H \in \mathbb{R}^{p \times q} \) are full column rank constant matrices.

**Assumption 2.3.** The pair \((A, C)\) in the LTI system (2.11) is uniformly detectable [26].

The steady state error covariance \( P_\infty \) can be obtained by solving the discrete algebraic Riccati equation (DARE) [49, p. 194]

\[ P_\infty = AP_\infty A^\top + GQG^\top - AP_\infty C^\top \left( CP_\infty C^\top \right)^{-1} CP_\infty A^\top. \]

(2.12)

Hence, the steady state Kalman gain \( L_\infty \) is given by [49, p. 195]

\[ L_\infty = \left( P_\infty C^\top \right) \left( CP_\infty C^\top + HRH^\top \right)^{-1}. \]

(2.13)

### 2.4 Extended Kalman Filter

Consider the nonlinear system

\[ x_{k+1} = f(x_k, w_k) \]
\[ y_k = h(x_k, v_k) \]

(2.14)

where the nonlinear functions \( f(\cdot) \) and \( h(\cdot) \) are differentiable, \( x_k \in X_k \) is the unknown state, \( y_k \in \mathbb{R}^p \) is the output measurement. \( w_k \in \mathbb{W}_k \) and \( v_k \in \mathbb{V}_k \) are two unknown noise terms, which affect the state and output, respectively. We assume that for all \( k > 0 \), \( 0 \in \mathbb{W}_k \) and \( 0 \in \mathbb{V}_k \).

The extended Kalman filter (EKF) approximates the nonlinear system (2.14) by an LTV system by using the first order Taylor series expansion, such that

\[ f(x_k, w_k) = f(\hat{x}_k, 0) + A_k(x_k - \hat{x}_k) + G_k(w_k - 0) + \text{error}_f \]

(2.15a)

\[ h(x_k, v_k) = h(\hat{x}_k, 0) + C_k(x_k - \hat{x}_k) + H_k(v_k - 0) + \text{error}_h \]

(2.15b)
where

$A_k := \frac{\partial f(\cdot)}{\partial x_k} \bigg|_{x_k = \hat{x}_k, w_k = 0}$, \quad $G_k := \frac{\partial f(\cdot)}{\partial w_k} \bigg|_{x_k = \hat{x}_k, w_k = 0}$, \quad $C_k := \frac{\partial h(\cdot)}{\partial x_k} \bigg|_{x_k = \hat{x}_k, v_k = 0}$, \quad $H_k := \frac{\partial h(\cdot)}{\partial v_k} \bigg|_{x_k = \hat{x}_k, v_k = 0}$. \quad (2.15c)

Hence, by ignoring error terms in (2.15), $P_{k+1|k}$ and $L_{k+1}$ can be approximated by

$$
C\{x_{k+1}|y_{1:k}\} = \mathbb{E}\{(f(x_k, w_k) - f(\hat{x}_k, 0))(f(x_k, w_k) - f(\hat{x}_k, 0))^\top\}
\approx A_k P_k A_k^\top + G_k Q G_k^\top = P_{k+1|k},
$$

$$
C\{x_{k+1}, y_{k+1}|y_{1:k}\} = \mathbb{E}\{(f(x_k, w_k) - f(\hat{x}_k, 0))(y_{k+1} - h(\hat{x}_{k+1}, 0))^\top\} \approx P_{k+1|k} C_{k+1},
$$

$$
C\{y_{k+1}|y_{1:k}\} = \mathbb{E}\{(y_{k+1} - h(\hat{x}_{k+1}, 0))(y_{k+1} - h(\hat{x}_{k+1}, 0))^\top\}
\approx C_{k+1} P_{k+1|k} C_{k+1} + H_{k+1} R H_{k+1}^\top,
$$

$$
L_{k+1} = \left(P_{k+1|k} C_{k+1}ight)^{-1} \left(C_{k+1} P_{k+1|k} C_{k+1} + H_{k+1} R H_{k+1}^\top\right). \quad (2.15c)
$$

Therefore, the EKF firstly linearizes nonlinear functions to update the covariance matrix using (2.15) and then applies the Kalman filter algorithm (2.4) for nonlinear state estimation. Thus, the EKF is neither the unbiased mean-squared error estimator nor the minimum covariance unbiased estimator of nonlinear system (2.14). However, the EKF is the best linear unbiased estimator of the linearized dynamical system, which can often be a good approximation of the original system [19]. Moreover, the EKF does not guarantee stability and convergence of state estimation. In [44], a few additional conditions were introduced for improving the stability and convergence of the EKF, including observability, small initial estimation error, small noise terms and no model mismatch.

Note that, unlike the smoother for LTV systems, because $A_k$ and $P_k$ in (2.9) are approximated by the EKF via linearizing $f(\cdot)$ and $h(\cdot)$ around the current state estimate $\hat{x}_k$, state estimation may not significantly improve the nonlinear state estimation by using the “forward-backward” smoothing algorithm (2.9). A different approach is needed for improving the stability and accuracy of both filtering and smoothing in nonlinear systems.

### 2.5 Full Information Estimation

The EKF provides a solution for estimating states of the nonlinear system (2.14) by linearizing nonlinear functions with a first-order Taylor expansion around the current estimate. However, when the error in the higher order terms neglected by the linear (first-order) model are significant, the EKF can exhibit poor convergence characteristics and biased estimates [45]. Moreover, general recursive solutions, such as Kalman filtering, are unavailable if system
states, inputs and/or noises could be limited by physical constraints. Consider the nonlinear system
\begin{equation}
\begin{aligned}
x_{k+1} &= f(x_k) + G_k w_k \\
y_k &= h(x_k) + H_k v_k,
\end{aligned}
\end{equation}
where noise terms are additive to the nonlinear dynamics. One strategy for solving these two challenges is to avoid model linearization by reformulating the original estimation as an optimization problem, which allows for the natural addition of inequality constraints [39].

Given measurements \( (y_k)_{k=1}^M \) and initial conditions, the conditional density \( p(x_k|y_{1:M}) \) is difficult to obtain exactly for a nonlinear model, thus we focus our attention on the entire trajectory of states \( (x_k)_{k=1}^M \), rather than just one single state \( x_k \) by maximizing the likelihood function \( \mathcal{L}(y_{1:M}|x_{1:M}) \) [42].

**Assumption 2.4.** The discrete-time nonlinear model (2.16) is uniformly observable [30]. The nonlinear functions \( f(\cdot) \) and \( h(\cdot) \) are twice differentiable and there exists a stable state observer for (2.16) with nonempty feasible region.

**Assumption 2.5.** The marginal probability distribution \( x_1 \mapsto p(x_1) \) of the initial state \( x_1 \) has Gaussian distribution \( p(x_1) \sim N(\hat{x}_{1|0}, P_{1|0}) \), where \( \hat{x}_{1|0} \) is the a priori most likely value of \( x_1 \) and \( P_{1|0} \) is the corresponding error covariance.

**Theorem 2.1.** [50, p. 22] Let \( p(x) \) be the probability density function of continuous random variable \( x \), if \( y = f(x) \) be a bijective transformation\(^1\), then the probability density function of random variable \( y \), \( p(y) \) is given by
\[ p(y) = p \left( f^{-1}(y) \right) \det \left( J \left( f^{-1}(y) \right) \right), \]
where \( J \left( f^{-1}(y) \right) \) is the Jacobian matrix of \( f^{-1}(y) \) respect to \( y \).

**Definition 2.1.** [3, p. 31], [16, pp. 376-377] A singular joint normal distribution of random variables \( u_1 \in \mathbb{R}^{n_1 \times 1} \) and \( u_2 \in \mathbb{R}^{n_2 \times 1} \), is denoted by
\begin{equation}
\begin{bmatrix}
u_1 \\
u_2
\end{bmatrix}
\sim 
\mathcal{N}
\left(
\begin{bmatrix}
m_1 \\
m_2
\end{bmatrix},
\begin{bmatrix}
U_1 & 0 \\
0 & 0
\end{bmatrix}
\right),
\end{equation}
where \( U_1 \succ 0 \). Hence, the density function is defined by
\begin{equation}
p(u_1, u_2) := \frac{1}{\sqrt{(2\pi)^{n_1}|U_1|}} e^{-\frac{1}{2}||u_1-m_1||^2_{U_1^{-1}}} \delta(u_2-m_2).
\end{equation}
\(^1x = f^{-1}(y)\) exists and also a bijective transformation.
For the joined normal distribution (2.17), it is clear that \( u_2 \) is deterministic and equal to \( m_2 \), the probability density function of \( u_2 \) is represented by the Dirac delta function \( \delta(u_2 - m_2) \), where the area under the delta function is 1.

**Lemma 2.3.** Consider two random vectors \( w \in \mathbb{R}^{r\times 1} \) and \( x \in \mathbb{R}^{n\times 1} \), \( n > r \), where \( x \) can be obtained from \( w \) via a linear transformation \( x = m + Gw \) with a constant vector \( m \) and full column rank matrix \( G \). If \( w \sim N(0,Q) \) and \( Q \succ 0 \), then the probability distribution of \( x \) is given by

\[
p(x) \propto \frac{1}{\sqrt{(2\pi)^r|W|}} e^{-\frac{1}{2}||A(x-m)||^2_W},
\]

where \( W \) is a positive definite matrix and \( A \) is a constant matrix.

**Proof.** Because \( G \) is a full column rank matrix, random vector \( x \) has a singular joint normal distribution, such that

\[
p(x) \sim N\left(m, GQG^\top \right),
\]

where \( GQG^\top \succeq 0 \). By using the singular value or eigenvalue decomposition we have

\[
GQG^\top = U \hat{Q} U^\top = U \begin{bmatrix} \hat{Q} & 0 \\ 0 & 0 \end{bmatrix} U^\top,
\]

where \( \hat{Q} \in \mathbb{R}^r \) is a nonsingular matrix, \( U \) is a unitary matrix, such that \( U^{-1} = U^\top \). Define a new random vector \( z \), such that

\[
z := \begin{bmatrix} z_1^\top \\ z_2^\top \end{bmatrix}^\top = U^{-1}x,
\]

where \( z_1 \in \mathbb{R}^{r\times 1} \). Since \( z \sim N(0,P) \), by Definition 2.1, we have

\[
p(z) = \frac{1}{\sqrt{(2\pi)^r|\hat{Q}|}} e^{-\frac{1}{2}||z_1 - \mathcal{M}_1^{n-r} U^{-1}m||^2_{\hat{Q}^{-1}}} \delta(z_2 - \mathcal{M}_{r+1}^{n-r} U^{-1}m),
\]

because \( \delta(z_2 - \mathcal{M}_{r+1}^{n-r} U^{-1}m) \) is the probability mass function of the degenerate (deterministic) variable \( z_2 \), we have

\[
p(z) \propto \frac{1}{\sqrt{(2\pi)^r|\hat{Q}|}} e^{-\frac{1}{2}||z_1 - \mathcal{M}_1^{n-r} U^{-1}m||^2_{\hat{Q}^{-1}}}.
\]
By Theorem 2.1 (transformation of variables), we have

\[ p(x) \propto \left| U^{-1} \right| e^{-\frac{1}{2} \| A(x-m) \|_W^2} \]

where \( W = \tilde{Q} \) and \( A = M^{r,n} U^{-1} \).

**Lemma 2.4.** For nonlinear system (2.16), given output measurements \((y_k)_{k=1}^M\) if Assumption 2.1 and 2.5 hold, \( G_k \) and \( H_k \) are full column rank matrices, then \( L(y_{1:M}|x_{1:M}) \) can be written as a function of \( x_1 \), \((w_k)_k \) and \((v_k)_{k=1}^M\).

**Proof.** Since measurements \((y_k)_{k=1}^M\) are given, by the definition of likelihood function stated in the Nomenclature and Bayes’ theorem we have

\[ \max L(y_{1:M}|x_{1:M}) \propto \max p(y_{1:M}|x_{1:M}) p(x_{1:M}), \]

where

\[ p(y_{1:M}|x_{1:M}) = \prod_{k=1}^M p(y_k|x_k), \quad p(x_{1:M}) = p(x_1) \prod_{k=1}^{M-1} p(x_{k+1}|x_k), \]

since \( w_k \) and \( v_k \) in (2.16) satisfy Assumption 2.1, we have

\[ p(y_k|x_k) \sim N(h(x_k), H_k R_k^{-1}) \quad \text{and} \quad p(x_{k+1}|x_k) \sim N(f(x_k), G_k Q_k G_k^{-1}). \]

By Lemma 2.3 we have

\[ p(x_{k+1}|x_k) \propto \frac{1}{\sqrt{(2\pi)^q|R_k|}} e^{-\frac{1}{2} \| \mathcal{M}_1^{r,p} \tilde{R}_k^{-1}(x_{k+1} - f(x_k)) \|_{R_k}^2}, \]

\[ p(y_k|x_k) \propto \frac{1}{\sqrt{(2\pi)^q|R_k|}} e^{-\frac{1}{2} \| \mathcal{M}_1^{r,p} \tilde{Y}_k^{-1}(y_k - h(x_k)) \|_{R_k}^2}, \]

(2.19)

where

\[ G_k Q_k G_k^\top = \mathcal{P}_k \mathcal{Q}_k \mathcal{P}_k^\top, \]

\[ \tilde{Q}_k = \mathcal{M}_1^{r,n} \mathcal{Q}_k^{-1} G_k Q_k G_k^\top \mathcal{P}_k^{-\top}, \]

\[ \tilde{R}_k = \mathcal{M}_1^{r,p} \mathcal{Y}_k^{-1} H_k R_k H_k^\top \mathcal{Y}_k^{-\top}. \]

\( \mathcal{P}_k \) and \( \mathcal{Y}_k \) are unitary matrices, \( \tilde{Q}_k \) and \( \tilde{R}_k \) are positive definite matrices.
Rearranging \( ||M_1^{r,n} X_k^{-1}(x_{k+1} - f(x_k))||^2_{\tilde{Q}_k}^{-1} \) in (2.19) gives

\[
||M_1^{r,n} X_k^{-1}(x_{k+1} - f(x_k))||^2_{\tilde{Q}_k}^{-1} = ||M_1^{r,n} X_k^{-1}G_k w_k||^2_{\tilde{Q}_k}^{-1}
= w_k^T G_k^{\top} X_k^{-1}M_1^{r,n} X_k^{-1}G_k w_k = w_k^T G_k^{\top} X_k^{-1}(P_k^T)^\dagger X_k^{-1}G_k w_k
= w_k^T G_k^\top \left( X_k P_k^T X_k^{-1} \right)^\dagger G_k w_k
= w_k^T G_k^\top G_k (G_k^\top G_k)^{-1} Q^{-1}(G_k^\top G_k)^{-1} G_k w_k = w_k^T Q^{-1} w_k.
\]

(2.20a)

Similarly for the term \( ||M_1^{q,p} Y_k^{-1}(y_k - h(x_k))||^2_{\tilde{R}_k}^{-1} \) in (2.19), we have

\[
||M_1^{q,p} Y_k^{-1}(y_k - h(x_k))||^2_{\tilde{R}_k}^{-1} = v_k^T R^{-1} v_k.
\]

(2.20b)

Substituting (2.20) into (2.19) and replacing \( ||\tilde{Q}_k|| \) and \( ||\tilde{R}_k|| \) with \( ||Q|| \) and \( ||R|| \), respectively, gives

\[
p(x_{k+1}|x_k) \propto \frac{1}{\sqrt{(2\pi)^d||Q||}} e^{-\frac{1}{2}||M_1^{r,n} X_k^{-1}(x_{k+1} - f(x_k))||^2_{\tilde{Q}_k}^{-1}} \propto \frac{1}{\sqrt{(2\pi)^d||Q||}} e^{-\frac{1}{2}||w_k||^2_{\tilde{Q}_k}^{-1}},
\]

\[
p(y_k|x_k) \propto \frac{1}{\sqrt{(2\pi)^d||R||}} e^{-\frac{1}{2}||M_1^{q,p} Y_k^{-1}(y_k - h(x_k))||^2_{\tilde{R}_k}^{-1}} \propto \frac{1}{\sqrt{(2\pi)^d||R||}} e^{-\frac{1}{2}||v_k||^2_{\tilde{R}_k}^{-1}}.
\]

Hence, we have

\[
p(y_{1:M}|x_{1:M}) = \prod_{k=1}^M p(y_k|x_k) \propto \prod_{k=1}^M \frac{1}{\sqrt{(2\pi)^d||R||}} e^{-\frac{1}{2}v_k^T R^{-1} v_k},
\]

\[
p(x_{1:M}) = p(x_1) \prod_{k=1}^{M-1} p(x_{k+1}|x_k) \propto p(x_1) \prod_{k=1}^{M-1} \frac{1}{\sqrt{(2\pi)^d||Q||}} e^{-\frac{1}{2}w_k^T Q^{-1} w_k}.
\]

Therefore, by Assumption 2.5,

\[
\max_\mathcal{L}(y_{1:M}|x_{1:M}) = \max p(x_{1:M}|y_{1:M}) \propto \max \frac{1}{\sqrt{(2\pi)^d||P||}} e^{-\frac{1}{2}(x_1 - \tilde{x}_{1:0})^T P_{10}^{-1}(x_1 - \tilde{x}_{1:0})} \times \prod_{k=1}^{M-1} \frac{1}{\sqrt{(2\pi)^d||Q||}} e^{-\frac{1}{2}w_k^T Q^{-1} w_k} \prod_{k=1}^M \frac{1}{\sqrt{(2\pi)^d||R||}} e^{-\frac{1}{2}v_k^T R^{-1} v_k}.
\]

(2.21)

\[
\square
\]

For many applications involving likelihood functions, it is more convenient to work in terms of the natural logarithm of the likelihood function than the likelihood function.
itself. This is because, firstly, the logarithm function is a monotone increasing function that achieves its maximum value at the same points as the original function $L(\cdot|y_{1:M})$. Secondly, maximizing the likelihood function usually involves taking the derivative; the logarithm transfers product of variables as well as exponential terms into sum of variables and products, respectively, and the derivative of a log-likelihood function is often easier to compute than the derivative of a likelihood function. The expression of the log-likelihood function $L(\ell(y_{1:M}|x_{1:M}))$ is given by

$$L(\ell(y_{1:M}|x_{1:M})) = \log(p(x_{1:M}|y_{1:M})) = -\frac{1}{2} (s \log(2\pi) + \log |P_{1}| + M \log |R| + M \log |Q|) - \frac{1}{2} ||x_1 - \hat{x}_1||^2_{P_{1}^{-1}} - \frac{1}{2} \sum_{k=1}^{M-1} ||v_k||^2_{R_k^{-1}} - \frac{1}{2} \sum_{k=1}^{M} ||w_k||^2_{Q_k^{-1}},$$

(2.22)

where $s := n + r + q$. Since $P_1, Q$ and $R$ are all given, the full information estimation for a nonlinear system (2.16) is defined by

$$\begin{align*}
X_{1,M}^* &:= \arg \min_{\hat{x}_{1,M}} \frac{1}{2} ||x_1 - \hat{x}_1||^2_{P_{1}^{-1}} + \frac{1}{2} \sum_{k=1}^{M-1} ||v_k||^2_{R_k^{-1}} + \frac{1}{2} \sum_{k=1}^{M} ||w_k||^2_{Q_k^{-1}} \\
s.t. x_{k+1} &= f(x_k) + G_kv_k, \quad k = 1, \ldots, M - 1 \\
y_k &= h(x_k) + H_kv_k, \quad k = 1, \ldots, M \\
x_k \in X_k, w_k \in W_k, v_k \in V_k, \quad k = 1, \ldots, M
\end{align*}$$

(2.23)

where the decision variables and their optimal solutions are

$$
\begin{array}{c}
X_{1,M} := \begin{bmatrix} x_1^T & w_1^T & v_1^T & x_2^T & \cdots & x_{M-1}^T & w_{M-1}^T & v_{M-1}^T & x_M^T \end{bmatrix}^T, \\
\hat{X}_{1,M} := \begin{bmatrix} \hat{x}_{1|M}^T & \hat{w}_{1|M}^T & \hat{v}_{1|M}^T & \hat{x}_{2|M}^T & \cdots & \hat{x}_{M-1|M}^T & \hat{w}_{M-1|M}^T & \hat{v}_{M-1|M}^T & \hat{x}_M^T \end{bmatrix}^T,
\end{array}
$$

(\hat{x}_{k|\cdot})_{k=1}^M, (\hat{w}_{k|\cdot})_{k=1}^M, \text{ and } (\hat{v}_{k|\cdot})_{k=1}^M \text{ are the estimates of the state and system noise sequence } (x_k)_{k=1}^M, (w_k)_{k=1}^M \text{ and } (v_k)_{k=1}^M \text{ respectively, with } (\cdot)_{M} := (\cdot)|_{M} \text{, given output sequence } y_{1:M}.}

$$
$$

For LTI system (2.11) and LTV system (2.1), (2.23) becomes a convex QP, for which finding the global optimum is guaranteed regardless of the choice of initial guess. However, for nonlinear systems, the global optimum $X_{1,M}^*$ is not guaranteed to be found. In order to prevent the solver from finding a local optimum that is far away from true states, one has to provide an appropriate initial guess $X_{1,M}^{(1)}$ as a warm-start [40, p. 157] based on previous
solutions, which can be recursively obtained by, for example

\[
X_{1,k}^{(1)} := \begin{bmatrix} X_{1,k-1}^\top & 0_{1,r} & 0_{1,q} & \hat{x}_{k-1}^\top \end{bmatrix}^\top, \quad \bar{k} = 2, \ldots, M,
\]

where

\[
X_{1,k}^* := \arg\min_{X_{1,k-1}} \frac{1}{2} \|x_1 - \hat{x}_1\|_2^2 + \frac{1}{2} \sum_{k=1}^{\bar{k}-1} \|w_k\|_Q^{-1} + \frac{1}{2} \sum_{k=1}^{\bar{k}} \|v_k\|_R^{-1}
\]

\[
\text{s.t. } x_{k+1} = f(x_k) + G_kw_k, \quad k = 1, \ldots, \bar{k} - 1
\]

\[
y_k = h(x_k) + H_kv_k, \quad k = 1, \ldots, \bar{k}
\]

\[
x_k \in \mathcal{X}_k, \quad w_k \in \mathcal{W}_k, \quad v_k \in \mathcal{V}_k, \quad k = 1, \ldots, \bar{k}
\]

**Theorem 2.2.** [37, Prop. 3.3.3] if Assumption 2.4 holds and \( \mathcal{X}_1 \) is a compact set, then for all \( \hat{x}_1 \in \mathcal{X}_1 \), the full information estimator (2.23) is an asymptotically stable observer [40, Def. 2.6] for the system

\[
x_{k+1} = f(x_k),
\]

\[
y_k = h(x_k).
\]

### 2.6 Moving Horizon Estimation

For the 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 bound the problem size by using moving horizon estimation (MHE) [40]. The basic strategy of MHE is to consider explicitly a fixed amount of data (the estimation window), while approximately summarizing the old data not explicitly accounted for by the nonlinear state estimator.

Instead of maximizing the probability of the whole state trajectory \( (x_k)_{k=1}^M \) given output measurements \( (y_k)_{k=1}^M \), the MHE only maximizes \( p(x_k; 1:M \mid y_{1:M}) \), \( 1 \leq k \leq M \), such that

\[
\max p(x_k; M \mid y_{1:M}) = \max p(x_k; M, y_{1:M} \mid y_{1:M}) \propto \max p(x_k; M, y_{1:M})
\]

\[
\propto \max p(x_k; M, y_{k+1:M} \mid y_{1:k}) p(y_{1:k}) \propto \max p(x_k; M, y_{k+1:M} \mid y_{1:k}),
\]

\[
\propto \max p(x_{k+1:M}, y_{k+1:M} \mid x_k, y_{1:k}) p(x_k \mid y_{1:k}).
\]
Since given $x_k$, the outputs $y_{1:k}$ provide no further information about $x_{k+1:M}$, we have

$$\max p(x_{k:M}|y_{1:M}) \propto \max p(x_{k+1:M}, y_{k+1:M}|x_k)p(x_k|y_{1:k}),$$

$$\propto \max p(y_{k+1:M}|x_{k+1:M}, x_k)p(x_{k+1:M}|x_k)p(x_k|y_{1:k}),$$

$$\propto \max p(y_{k+1:M}|x_{k+1:M})p(x_{k+1:M}|x_k)p(x_k|y_{1:k}),$$

where $p(x_k|y_{1:k}) = \frac{1}{\sqrt{(2\pi)^m|P_k|}} e^{-\frac{1}{2}||x_k-x_{k}^*||^2_{P_k^{-1}}}$, $p(x_{k+1:M}|x_k) = \prod_{k=k_s}^{M-1} \frac{1}{\sqrt{(2\pi)^m|Q|}} e^{-\frac{1}{2}||y_k||^2_{R^{-1}}}$.

Hence, for $\bar{k} = 2, \ldots, M$, we define the MHE problem as follows [40]:

$$\Theta_{\bar{k}}^e = \max p(x_{k,\bar{k}}|y_{1:\bar{k}}) = \min_{x_{k,\bar{k}}} \frac{1}{2} \sum_{k=k_s}^{\bar{k}-1} ||w_k||^2_{Q^{-1}} + \frac{1}{2} \sum_{k=k_s}^{\bar{k}} ||v_k||^2_{R^{-1}} + \frac{1}{2} ||x_k - \hat{x}_k||^2_{P_k^{-1}} + \Theta_{\bar{k}-H_l}^e$$

subject to

$$x_{k+1} = f(x_k) + G_kw_k, \quad k = k_s, \ldots, \bar{k}-1$$

$$y_k = h(x_k) + H_kv_k, \quad k = k_s, \ldots, \bar{k}$$

$$x_k \in X_k, w_k \in W_k, v_k \in V_k, \quad k = k_s, \ldots, \bar{k}$$

(2.26a)

(2.26b)

(2.26c)

(2.26d)

where $k_s := \max\{\bar{k} - H_l, 0\} + 1$; $H_l$ is a positive integer called the horizon length and $H_l \leq M$. The MHE is equivalent to FIE, if $\bar{k} = H_l$; for $\bar{k} \leq H_l$, the term $\Theta_{\bar{k}-H_l}^e := ||v_1||^2_{R^{-1}}$, $\hat{x}_1 := \hat{x}_{1|0}$ and $P_{k_s} := P_{1|0}$.

The term $\mathcal{Z}_{k_s}(x)$ is called the arrival cost, which is a fundamental concept in MHE [40]. The arrival cost compactly summarizes the effect of the data $(y_k)_{k=1}^{\bar{k}-1}$ on the state $x_{k_s}$, thereby allowing one to fix the dimension of the optimization $\mathcal{P}_M$ [37, p. 16]. For linear systems, the arrival cost can be exactly determined using the Kalman filter. However, when the system is nonlinear or constrained, an algebraic expression for the arrival cost rarely exists [40].

If $\bar{k} > H_l$, $\Theta_{\bar{k}-H_l}^e$ in the arrival cost $\mathcal{Z}_{k_s}(x)$ will be a constant part and $\hat{x}_{k_s}$ can be obtained from previous estimates. Therefore, we only need to focus on determining the error covariance $P_{k_s}$. An algorithm for estimating the arrival cost $\mathcal{Z}(\cdot)$ based on monitoring and limiting the growth of $\Theta_{k}^e$ is given in [40], which does not require any linearization or approximation, but still guarantees the observer to be locally asymptotically stable [40, Def. 2.6] for the system (2.25) if Assumption 2.4 holds.

The main disadvantage of the original estimation algorithm given in [40] is that one has to sequentially solve the optimization problems (2.26) and (2.27) at every time step. However, in most of cases, solving the optimization problems (2.26) and (2.27) could be
time-consuming, especially when the number of decision variables and horizon length are large. Thus, the original algorithm has been rearranged in this thesis to be compatible with a parallel computing system, which could save up to 50% of computing time.

The flow chart of the algorithm is shown in Figure 2.1, where $P_{k_s}^{-1} := \gamma W^{-1}$, $\gamma \in [0, 1]$. The weight $W$ in the flow chart could be any given positive-definite matrix [40] and the optimization problem is defined as [40]

$$
\Theta_k^0 = \min_{X_{k_s}^0, k} \frac{1}{2} \sum_{k=k_s}^{\bar{k}} \| v_k \|_R^{-1}^2 + \frac{1}{2} \sum_{k=k_s}^{\bar{k}-1} \| w_k \|_Q^{-1}^2
$$

s.t.

$$
x_{k+1} = f(x_k) + G_k w_k, \quad k = k_s, \ldots, \bar{k} - 1
$$

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

$$
x_k \in X_k, \; w_k \in W_k, \; v_k \in V_k, \quad k = k_s, \ldots, \bar{k}
$$

with the decision variables and its optimal solutions as

$$
X_{k_s, \bar{k}}^0 := \begin{bmatrix} x_{k_s}^T & w_{k_s}^T & v_{k_s}^T & \cdots & x_{k-1}^T & w_{k-1}^T & v_{k-1}^T & x_{\bar{k}}^T \end{bmatrix}^T,
$$

$$
X_{k_s, \bar{k}}^{0*} := \begin{bmatrix} x_{k_s}^{0*} & w_{k_s}^{0*} & v_{k_s}^{0*} & \cdots & x_{k-1}^{0*} & w_{k-1}^{0*} & v_{k-1}^{0*} & x_{\bar{k}}^{0*} \end{bmatrix}^T.
$$

**Theorem 2.3.** [40, Coro. 3.8] if Assumption 2.4 holds, $X_k$, $W_k$ and $V_k$ are all compact sets, then there exists a long enough horizon $H_f$, such that the estimation error $e_{\bar{k}} := \| x_{\bar{k}} - \hat{x}_{\bar{k}} \|$ for the estimation algorithm in Figure 2.1 is bounded for all $\bar{k} \geq H_f$.

### 2.7 Relations Between the Kalman Filter/Smoother and Full Information Estimation

In [19], the Kalman filter as well as the EKF are considered from a different point of view, where the algorithms can be derived from a single iteration of Newton’s method on a certain quadratic form with a judiciously chosen initial guess [19]. This idea is very important and allows us to establish a relationship between Kalman-based algorithms and the FIE, hence we could compare their performance for both linear and nonlinear systems.
Fig. 2.1 Flow chart of moving horizon estimation
2.7 Relations Between the Kalman Filter/Smoother and Full Information Estimation

**Theorem 2.4.** For LTV system (2.1), given output measurements \((y_k)_{k=1}^{\hat{k}}\), \(1 < \hat{k} \leq M\), if Assumption 2.5 holds, then the FIE problem

\[
\mathcal{X}^*_{1,\hat{k}} := \arg \min_{x_{1,\hat{k}}} \frac{1}{2} \|x_1 - \hat{x}_{1|0}\|_{P_{1|0}}^2 + \frac{1}{2} \sum_{k=1}^{\hat{k}-1} \|w_k\|_{Q^{-1}}^2 + \frac{1}{2} \sum_{k=1}^{\hat{k}} \|v_k\|_{R^{-1}}^2
\]

subject to

\[
x_{k+1} = A_k x_k + G_k w_k, \quad k = 1, \ldots, \hat{k}-1
\]

\[
y_k = C_k x_k + H_k v_k, \quad k = 1, \ldots, \hat{k}
\]

is a linear unbiased estimator [25, p. 11] of the state sequence \((x_k)_{k=1}^{\hat{k}}\) regardless of the choice of covariances \(P_{1|0}, Q\) and \(R\); if \(P_{1|0}, Q\) and \(R\) are accurate, then the FIE is a best linear unbiased estimator (BLUE) [25, p. 555] of the state sequence \((x_k)_{k=1}^{\hat{k}}\).

**Proof.** If both \(G_k\) and \(H_k\) are invertible matrices, then a proof can be found in [1]. By involving Lagrange multipliers \((\alpha_k)_{k=1}^{\hat{k}-1}, (\beta_k)_{k=1}^{\hat{k}}\) and \(\gamma\) for equality constraints and the initial condition, respectively, the FIE problem (2.28) can be written as the following unconstrained optimization problem:

\[
Z^*_k := \arg \min_{Z_k} S_k = \arg \min_{Z_k} \frac{1}{2} \left\{ \|\epsilon_1\|_{P_{1|0}}^2 + 2\gamma_1 (\hat{x}_{1|0} - x - \epsilon_1) \right\}
\]

\[
+ \sum_{k=1}^{\hat{k}-1} \|w_k\|_{Q^{-1}}^2 + 2\alpha_k^T (x_{k+1} - A_k x_k - G_k w_k) + \sum_{k=1}^{\hat{k}} \|v_k\|_{R^{-1}}^2 + 2\beta_k^T (y_k - C_k x_k - H_k v_k) \right\}
\]

where \(Z_1 := [\epsilon_{1|0}^T, v_{1}^T, \gamma_1^T, \beta_1^T, x_1^T]\),

\[
Z^*_k := \begin{bmatrix} Z_1 \ v_1^T \ \alpha_1^T \ \beta_1^T \ \hat{x}_2^T \ \cdots \ \hat{x}_{k-1}^T \ w_k^T \ \alpha_{k-1}^T \ \beta_{k-1}^T \ \hat{x}_k^T \end{bmatrix}^T,
\]

\[
Z_k^T = [\hat{\epsilon}_{1|k}^T \ \hat{\gamma}_{1|k}^T \ \hat{\beta}_{1|k}^T \ \hat{x}_2^T \ \cdots \ \hat{x}_{k-1}^T \ \hat{w}_k^T \ \hat{\alpha}_{k-1|k}^T \ \hat{\beta}_{k-1|k}^T \ \hat{x}_k^T]^T.
\]

We define \((S_k)_{k=1}^{\hat{k}}\) as

\[
S_1 := \frac{1}{2} \|\epsilon_1\|_{P_{1|0}}^2 + \frac{1}{2} \|v_{1}\|_{R^{-1}}^2 + \beta_1^T (y_1 - C_1 x_1 - H_1 v_1),
\]

\[
S_2 := S_1 + \frac{1}{2} \|v_2\|_{R^{-1}}^2 + \beta_2^T (y_2 - C_2 x_2 - H_2 v_2) + \frac{1}{2} \|w_1\|_{Q^{-1}}^2 + \alpha_1^T (x_2 - A_1 x_1 - G_1 w_1)
\]

\[
S_k = S_{k-1} + \frac{1}{2} \|v_k\|_{R^{-1}}^2 + \beta_k^T (y_k - C_k x_k - H_k v_k) + \frac{1}{2} \|w_{k-1}\|_{Q^{-1}}^2 + \alpha_{k-1}^T (x_k - A_{k-1} x_{k-1} - G_{k-1} w_{k-1}).
\]
The necessary condition of $Z^*_k$ being an optimum of (2.29) is that
\[ \frac{\partial S_k(Z_k)}{\partial Z_k} = J_{S_k}(Z_k) = H_{S_k} Z_k - b_{S_k} = 0, \]  
(2.30)
where the Hessian matrix $H_{S_k}$ is defined by
\[ H_{S_k} := \frac{\partial^2 S_k(Z_k)}{\partial Z_k^2} = \left[ \begin{array}{c} H_{S_{k-1}} \ 0 \\ 0 \ 0_{N_0} \end{array} \right] + \left[ \begin{array}{c} 0_{N_{k-1} - n} \ 0 \\ 0 \ H_{S_{k-1}}^+ \end{array} \right], \]
where $N_0 := r + q + 2n + p$, $N_{k-1}$ is the number of rows/columns of previous Hessian matrix $H_{S_{k-1}}$; matrices $H_{S_1}$ and $H_{S_{k-1}}^+$ are given by
\[ H_{S_1} := \begin{bmatrix} P_{1|0}^{-1} & 0 & -I_n & 0 & 0 \\ 0 & R^{-1} & 0 & -H_1^T & 0 \\ -I_n & 0 & 0 & 0 & -I_n \\ 0 & -H_1 & 0 & 0 & -C_1 \\ 0 & 0 & I_n & -C_1^T & 0 \end{bmatrix} \]
and
\[ H_{S_{k-1}}^+ := \begin{bmatrix} 0 & 0 & 0 & -A_{k-1}^T & 0 & 0 \\ 0 & Q^{-1} & 0 & -G_{k-1}^T & 0 & 0 \\ 0 & 0 & R^{-1} & 0 & -H_k^T & 0 \\ -A_{k-1} & -G_{k-1} & 0 & 0 & 0 & I_n \\ 0 & 0 & -H_k & 0 & 0 & -C_k \\ 0 & 0 & 0 & I_n & -C_k^T & 0 \end{bmatrix}, \]
respectively; the constant vector $b_{S_k}$ is defined by
\[ b_{S_k} := \begin{bmatrix} 0_{1,n+q} \ -\hat{x}_{1|0}^T \ -y_1^T \ 0_{1,n} \ 0_{1,r+q+n} \ y_2^T \ 0_{1,r+q+2n} \ -y_3^T \ \cdots \ -y_k^T \ 0_{1,n} \end{bmatrix}^T = H_{S_k} Z_k = H_{S_k} Z_k^x + B_{S_k} Z_k^d, \]
where
\[ B_{S_k} := H_{S_k} \times \text{diag} \left( \begin{bmatrix} \text{diag} \left( I_n, I_q \right) \\ 0_{2n+p,n+q} \end{bmatrix}, I_{k-1} \otimes \begin{bmatrix} \text{diag} \left( I_r, I_q \right) \\ 0_{2n+p,q+r} \end{bmatrix} \right) \]
and
\[ Z_k^d := \begin{bmatrix} e_1^T & v_1^T & w_1^T & v_2^T & w_2^T & v_3^T & \cdots & w_{k-1}^T & v_k^T \end{bmatrix}^T, \]
Taking the expectation on both sides yields

$$E\{Z^*_k\} = \mathcal{H}^{-1}_{S_k}E\{b_{S_k}\} = \mathcal{H}^{-1}_{S_k}E\{\mathcal{H}_{S_k}Z^*_k + B_{S_k}Z^*_k\} = Z^*_k.$$ 

Now, if we build the matrix $\mathcal{H}_{S_k}$ based on inaccurate $P_{1|0}$, $Q$ and $R$, and called it $\tilde{\mathcal{H}}_{S_k}$, $\tilde{P}_{1|0}$, $\tilde{Q}$ and $\tilde{R}$, respectively, then there exists a matrix $D_H$ such that $D_H := \mathcal{H}_{S_k} - \tilde{\mathcal{H}}_{S_k}$, with only few non-zero entries on the diagonal, such that

$$D_H = \text{diag}\left(\begin{bmatrix} D_P & D_R & 0_{2n+p} & D_Q & D_R & 0_{2n+p} & \cdots & D_Q & D_R & 0_{2n+p} \end{bmatrix}\right),$$

where $D_P := P^{-1}_{1|0} - \tilde{P}^{-1}_{1|0}$, $D_Q := Q^{-1} - \tilde{Q}^{-1}$ and $D_R := R^{-1} - \tilde{R}^{-1}$. Hence we have $D_HZ^*_k = 0$ and

$$E\{\tilde{Z}^*_k\} = \tilde{\mathcal{H}}^{-1}_{S_k}E\{b_{S_k}\} = \tilde{\mathcal{H}}^{-1}_{S_k}(\tilde{\mathcal{H}}_{S_k} + D_H)Z^*_k = Z^*_k + \tilde{\mathcal{H}}^{-1}_{S_k}D_HZ^*_k = Z^*_k.$$

Since $(x_k)_{k=1}^{\tilde{k}}$ are parts of vector $Z^*_k$, the FIE is an unbiased linear estimator of the state sequence $(x_k)_{k=1}^{\tilde{k}}$, regardless the choice of covariances $P_{1|0}$, $Q$ and $R$. The covariance of $Z^*_k$ is given by

$$\mathbb{C}\{Z^*_k\} = \mathcal{H}^{-1}_{S_k}B_{S_k}\Sigma B_{S_k}^\top \mathcal{H}^{-\top}_{S_k},$$

where $\Sigma = \text{diag}(P_{1|0}, R, I_{k-1} \otimes \text{diag}(Q, R))$. There exists a matrix $D_{H^{-1}} := \mathcal{H}_{S_k}^{-1} - \tilde{\mathcal{H}}_{S_k}^{-1}$, hence we have

$$\mathbb{C}\{\tilde{Z}^*_k\} = \left(\mathcal{H}^{-1}_{S_k} + D_{H^{-1}}\right)B_{S_k}\Sigma B_{S_k}^\top \left(\mathcal{H}^{-1}_{S_k} + D_{H^{-1}}\right)^\top,$n

$$= \mathbb{C}\{Z^*_k\} + 2\mathcal{H}^{-1}_{S_k}B_{S_k}\Sigma B_{S_k}^\top D_{H^{-1}} + D_{H^{-1}}B_{S_k}\Sigma B_{S_k}^\top D_{H^{-1}}.$$

Since $D_{H^{-1}}B_{S_k}\Sigma B_{S_k}^\top D_{H^{-1}} \succeq 0$ and

$$\mathcal{H}^{-1}_{S_k}B_{S_k}\Sigma B_{S_k}^\top = \text{diag}\left(\begin{bmatrix} E_k \\ 0_{2n+p, 3n+p+q} \end{bmatrix} \bigoplus_{k=2}^\tilde{k} \begin{bmatrix} E_k \\ 0_{2n+p, N_0} \end{bmatrix}\right),$$

2.7 Relations Between the Kalman Filter/Smoother and Full Information Estimation
where
\[
\mathcal{E} := \begin{bmatrix} I_n & 0 & -P_{1|0} & 0 & 0 \\ 0 & I_p & 0 & -H_1 R & 0 \end{bmatrix}, \quad \mathcal{E}_k := \begin{bmatrix} I_n & 0 & -G_{k-1} Q & 0 & 0 \\ 0 & I_p & 0 & -H_k R & 0 \end{bmatrix},
\]

covariance of \((x_k)_{k=1}^\bar{k}\) is minimized if \(D_{H^{-1}} = 0\). Thus, the FIE is a best linear unbiased estimator (BLUE) of the state sequence \((x_k)_{k=1}^\bar{k}\), if \(P_{1|0}, Q\) and \(R\) are all accurate.

Note that if matrices \(G_k\) and \(H_k\) are nonsingular for all \(1 \leq k \leq M\), then the FIE problem (2.28) can be equivalently written as a weighted linear least squares (WLLS) problem
\[
X^*_k := \arg\min_{X_k} \|AX_k - b\|_W^{-1}^2
\]
(2.32)
with the analytical solution given by
\[
X^*_k = \left(A^TW^{-1}A\right)^{-1}A^TW^{-1}b,
\]
where \(X_k := \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_{\bar{k}-1}^T \\ x_{\bar{k}}^T \end{bmatrix}, X^*_k := \begin{bmatrix} \hat{x}_{1|\bar{k}}^T \\ \hat{x}_{2|\bar{k}}^T \\ \vdots \\ \hat{x}_{\bar{k}-1|\bar{k}}^T \\ \hat{x}_{\bar{k}|\bar{k}}^T \end{bmatrix}^T,\)
\[
W := \text{diag} \left(P_{1|0}, I_{\bar{k}-1} \otimes \text{diag} (Q, R), R\right),
\]
\[
b := -B \times \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_{\bar{k}-1}^T \\ x_{\bar{k}}^T \end{bmatrix} \begin{bmatrix} 0_{1,n} & y_1^T \\ 0_{1,n} & y_2^T \\ \vdots \\ 0_{\bar{k}-1,n} & y_{\bar{k}-1}^T \\ 0_{\bar{k},n} & y_{\bar{k}}^T \end{bmatrix}^T
\]
(2.33a)
\[
A := B \times \begin{bmatrix} -I_n \\ -\Theta_{k=1}^{\bar{k}-1} \left[ -A_k \right] \\ \Theta_{k=1}^{\bar{k}-1} \left[ -C_k \right] \\ \left[ 0_{n,n(\bar{k}-1)} \right] \\ \left[ 0_{n,n(\bar{k})} \right] \end{bmatrix} + \begin{bmatrix} 0_n \\ \Theta_{k=1}^{\bar{k}-1} \left[ I_n \right] \\ \Theta_{k=1}^{\bar{k}-1} \left[ 0_{p,n} \right] \end{bmatrix}
\]
(2.33b)
\[
A := B \times \begin{bmatrix} -I_n \\ -\Theta_{k=1}^{\bar{k}-1} \left[ -A_k \right] \\ \Theta_{k=1}^{\bar{k}-1} \left[ -C_k \right] \\ \left[ 0_{n,n(\bar{k}-1)} \right] \\ \left[ 0_{n,n(\bar{k})} \right] \end{bmatrix} + \begin{bmatrix} 0_n \\ \Theta_{k=1}^{\bar{k}-1} \left[ I_n \right] \\ \Theta_{k=1}^{\bar{k}-1} \left[ 0_{p,n} \right] \end{matrix}
\]
(2.33c)
\[
and
\]
\[
B := \text{diag} \left(I_n, \bigoplus_{k=1}^{\bar{k}-1} \text{diag} \left(G_k^{-1}, H_k^{-1}\right), H_k^{-1}\right)
\]
(2.33d)
For any \(G_k\) and \(H_k\), it is a common mistake to write
\[
w_k = G_k^\top (x_{k+1} - A_k x_k), \quad k = 1, \ldots, \bar{k} - 1,
\]
\[
v_k = H_k^\top (y_k - C_k x_k), \quad k = 1, \ldots, \bar{k},
\]
(2.34a)
and then replace $G_k^{-1}$ and $H_k^{-1}$ in (2.33d) by $G_k^+$ and $H_k^+$, respectively. Unfortunately, solving the WLLS problem will not give any meaningful (and may not be unique) estimates unless matrices $G_k$ and $H_k$ are nonsingular. This is because, in order to satisfy equality constraints in (2.28), $x_k$ and $x_{k+1}$ are required to be uniquely determined by (2.34a) given $w_k$, $v_k$ and $y_k$.

**Proposition 2.4.** For the LTV system (2.1), given output measurements $(y_k)_{k=1}^\tilde{M}$, if Assumptions 2.1 and 2.5 hold, then the FIE problem (2.28) and Kalman filter are two equivalent methods for estimating the current state $\hat{x}_k$.

**Proof.** The proof is a generalization of the one in [19], which full column rank matrices $G_k$ and $H_k$ are taking into account. Since $\mathcal{H}_{S_k}$ is nonsingular for all $\tilde{k}$ (the proof is given in Appendix A). The proof of Theorem 2.4 shows that the current state $\hat{x}_k$ can be estimated using (2.31). By involving a constant vector $Z^*_k$ into (2.31), gives

$$Z^*_k = \mathcal{H}_{S_k}^{-1} b_s = Z^*_{k|\tilde{k}-1} - \mathcal{H}_{S_k}^{-1} (\mathcal{H}_{S_k} Z^*_{k|\tilde{k}-1} - b_s) = Z^*_{k|\tilde{k}-1} - \mathcal{H}_{S_k}^{-1} \mathcal{J}_{S_k},$$

which gives a single iteration of Newton’s method and the optimum $Z^*_k$ can be determined regardless of the choice of $Z^*_{k|\tilde{k}-1}$. Thus, we define $Z^*_{k|\tilde{k}-1}$ as

$$Z^*_{k|\tilde{k}-1} := \begin{bmatrix} 0_{1,2n+q+p} & \hat{x}_1^\top & 0_{1,r+q+n+p} & \hat{x}_2^\top & \cdots & 0_{1,r+q+n+p} & \hat{x}_{\tilde{k}|k-1}^\top \end{bmatrix}^\top,$$

hence, substituting $Z^*_{k|\tilde{k}-1}$ into $\mathcal{J}_{S_k}$ gives

$$\mathcal{J}_{S_k} = \begin{bmatrix} 0_{1,n+q} & (\hat{x}_1 - C_1\hat{x}_1)^\top & 0_{1,n+r+q} & (\hat{x}_2 - A_1\hat{x}_1)^\top & (y_2 - C_2\hat{x}_2)^\top & \cdots & 0_{1,2n+r+q} & (y_{\tilde{k}} - C_k\hat{x}_{\tilde{k}|k-1})^\top & 0_{1,n} \end{bmatrix}^\top.$$  

Since the Hessian matrix $\mathcal{H}_{S_k}$ can be written as

$$\mathcal{H}_{S_k} := \begin{bmatrix} A & C^\top \\ C & B \end{bmatrix} = \begin{bmatrix} \mathcal{H}_{S_k-1} & 0 & 0 & -\mathcal{M}_k^{nN_k-1} & 0 & 0 \\ 0 & Q^{-1} & 0 & -G_{k-1}^\top & 0 & 0 \\ 0 & 0 & R^{-1} & 0 & -H_k^\top & 0 \\ -A_{k-1}^\top \mathcal{M}_k^{nN_k-1} & -G_{k-1} & 0 & 0 & 0 & I_n \\ 0 & 0 & -H_k & 0 & 0 & -C_k \\ 0 & 0 & 0 & I_n & -C_k^\top & 0 \end{bmatrix}.$$  

(2.38)
Since block $A$ is nonsingular, by using [6, Fact 2.17.3] the corresponding lower right part of $(\mathcal{H}_k)^{-1}$ is given by

$$
\mathcal{M}_{N_A+1}^{N_B,N_k} \mathcal{H}^{-1}_{S_{k}} \mathcal{M}_{N_A+1}^{N_B,N_k} \top = \left( B - CA^{-1}C^\top \right)^{-1} = \begin{bmatrix} D & F \top \\ F & E \end{bmatrix}^{-1}
$$

$$
= \begin{bmatrix}
- A_{k-1} \mathcal{M}_{N_k-1-n+1}^{n_N_k} \mathcal{H}^{-1}_{S_{k-1}} \mathcal{M}_{N_k-1-n+1}^{n_N_k} \top A_{k-1}^\top - G_{k-1}QG_{k-1}^\top & 0 & - I_n \\
0 & - H_k R_k H_k^\top - C_k \\
- I_n & - C_k^\top & 0
\end{bmatrix}^{-1},
$$

where $N_A$ and $N_B$ are the number of rows/columns of block matrices $A$ and $B$, respectively. We then define

$$
P_{k-1} := \mathcal{M}_{N_k-1-n+1}^{n_N_k} \mathcal{H}^{-1}_{S_{k-1}} \mathcal{M}_{N_k-1-n+1}^{n_N_k} \top.
$$

Because $P_{k-1} \succ 0$ and $G_{k-1}QG_{k-1}^\top \succeq 0$, we have

$$
D = - A_{k-1} \mathcal{M}_{N_k-1-n+1}^{n_N_k} \mathcal{H}^{-1}_{S_{k-1}} \mathcal{M}_{N_k-1-n+1}^{n_N_k} \top A_{k-1}^\top - G_{k-1}QG_{k-1}^\top = - P_{k|k-1} \prec 0.
$$

Hence, the corresponding lower right part of $\mathcal{M}_{N_A+1}^{N_N,k} \mathcal{H}^{-1}_{S_{k}} \mathcal{M}_{N_A+1}^{N_N,k} \top$ gives

$$
\mathcal{M}_{N_A+N_D+1}^{N_N,k} \mathcal{H}^{-1}_{S_{k}} \mathcal{M}_{N_A+N_D+1}^{N_N,k} \top = \left( E - FD^{-1}F^\top \right)^{-1} = \begin{bmatrix}
- H_k R_k H_k^\top & - C_k \\
- C_k^\top & P_{k|k-1}^{-1}
\end{bmatrix}
= \begin{bmatrix}
- J \\
- P_{k|k-1}^{-1} C_k J P_{k|k-1} - P_{k|k-1}^{-1} C_k J C_k P_{k|k-1}
\end{bmatrix},
$$

(2.39)

where $N_D$ and $N_E$ are the number of rows/columns of block matrices $D$ and $E$, respectively, and $J := \left( C_k P_{k|k-1}^{-1} C_k^\top + H_k R_k H_k^\top \right)^{-1}$. Finally, we have

$$
\mathcal{M}_{N_k-n+1}^{n_N,k} \mathcal{H}^{-1}_{S_{k}} \mathcal{M}_{N_k-n+1}^{n_N,k} \top = P_{k|k-1} - P_{k|k-1}^{-1} C_k \left( C_k P_{k|k-1}^{-1} C_k^\top + H_k R_k H_k^\top \right)^{-1} C_k P_{k|k-1}
= P_k = \left( I - L_k C_k \right) P_{k|k-1},
$$
hence by substituting (2.36), (2.37) and (2.39) into (2.35) gives

\[ \mathcal{Z}_{N_k}^{n,N_k} c_k = \hat{x}_{k|k-1} + P_{k|k-1} C_k^T \left( C_k^T P_{k|k-1} C_k^T + H_k R H_k^T \right)^{-1} \left( y_k - C_k \hat{x}_{k|k-1} \right) \]

which is the same as the Kalman filter algorithm (2.4).

**Proposition 2.5.** For the LTV system (2.1), given output measurements \((y_k)_{k=1}^M\) and all estimated states \((\hat{x}_k)_{k=1}^M\), if Assumptions 2.1 and 2.5 hold, then the FIE and Kalman smoother are equivalent methods for estimating the smoothed state \(\hat{x}_{k|M}\), \(1 < k \leq M\).

**Proof.** The proof is inspired by [41]. The MHE for the LTV system (2.1) with \(H_i = 1\) is given by

\[
\begin{align*}
\mathcal{X}_{M-1,M}^* &= \arg \min_{\mathcal{X}_{M-1,M}} \frac{1}{2} \| x_{M-1} - \hat{x}_{M-1} \|^2_{P_{M-1}^{-1}} + \frac{1}{2} \| w_{M-1} \|^2_{Q^{-1}} + \frac{1}{2} \| v_M \|^2_{R^{-1}} \\
\text{s.t.} & \quad x_M = A_{M-1} x_{M-1} + G_{M-1} w_{M-1}, \quad (2.40b) \\
& \quad y_M = C_M x_M + H_{MV} \nu_M. \quad (2.40c)
\end{align*}
\]

By involving Lagrange multipliers \(\alpha_{M-1}\) and \(\beta_M\) for equality constraints, the MHE problem (2.40) can be written as the following unconstrained optimization problem:

\[
\begin{align*}
Z_{M-1,M}^* &:= \arg \min_{Z_{M-1,M}} \frac{1}{2} \| x_{M-1} - \hat{x}_{M-1} \|^2_{P_{M-1}^{-1}} + \frac{1}{2} \| w_{M-1} \|^2_{Q^{-1}} + \alpha_{M-1}^\top (x_M - A_{M-1} x_{M-1} - G_{M-1} w_{M-1}) \\
& \quad + \left[ \frac{1}{2} \| w_{M-1} \|^2_{Q^{-1}} + \alpha_{M-1}^\top (x_M - A_{M-1} x_{M-1} - G_{M-1} w_{M-1}) \right],
\end{align*}
\]

where

\[
\begin{align*}
Z_{M-1,M}^* := [x_{M-1}^\top \ w_{M-1}^\top \ v_M^\top \ \alpha_{M-1}^\top \ \beta_M^\top \ \hat{x}_M^\top]^\top,
\end{align*}
\]

\[
\begin{align*}
Z_{M-1,M}^* := [\hat{x}_{M-1|M}^\top \ \hat{w}_{M-1|M}^\top \ \hat{v}_M^\top \ \alpha_{M-1|M}^\top \ \beta_M^\top \ \hat{x}_M^\top]^\top.
\end{align*}
\]

Since all estimated states \((\hat{x}_k)_{k=1}^M\) are given, \(\nu_M\) is deterministic and \(\beta_M\) becomes a free variable, hence we have

\[
\begin{align*}
Z_{M-1|M}^* := \arg \min_{Z_{M-1|M}} S_{M-1|M} = \arg \min_{Z_{M-1|M}} \frac{1}{2} \| x_{M-1} - \hat{x}_{M-1} \|^2_{P_{M-1}^{-1}} \\
& \quad + \left[ \frac{1}{2} \| w_{M-1} \|^2_{Q^{-1}} + \alpha_{M-1}^\top (x_M - A_{M-1} x_{M-1} - G_{M-1} w_{M-1}) \right],
\end{align*}
\]
where $Z_{M-1|M} := \begin{bmatrix} x_{M-1}^\top & w_{M-1}^\top & \alpha_{M-1}^\top \end{bmatrix}^\top$ and $Z_{M-1|M}^* := \begin{bmatrix} \hat{x}_{M-1}^\top & \hat{w}_{M-1}^\top & \hat{\alpha}_{M-1}^\top \end{bmatrix}^\top$.

The necessary condition of $Z_{M-1|M}^*$ being an optimum of (2.40) is

$$\frac{\partial S_{M-1|M}(Z_{M-1|M})}{\partial Z_{M-1|M}} = \mathcal{H}_{S_{M-1|M}}(Z_{M-1|M}) = \mathcal{H}_{S_{M-1|M}}(Z_{M-1|M}) - b_{S_{M-1|M}} = 0,$$

where the Hessian matrix $\mathcal{H}_{S_{M-1|M}}$ and constant vector $b_{S_{M-1|M}}$ are defined by

$$\mathcal{H}_{S_{M-1|M}} := \begin{bmatrix} P_{M-1} & 0 & -A_{M-1}^\top \\ 0 & Q^{-1} & -G_{M-1}^\top \\ -A_{M-1} & -G_{M-1} & 0 \end{bmatrix} \quad \mbox{and} \quad b_{S_{M-1|M}} := \begin{bmatrix} P_{M-1} \hat{x}_{M-1} \\ 0 \\ -\hat{x}_M \end{bmatrix},$$

respectively. Since $P_{M-1}^{-1}$ is nonsingular, by using [6, Fact 2.17.3], we have

$$\hat{x}_{M-1|M} = \mathcal{H}_{1}^{n+2n+r} Z_{M-1|M}^* = \mathcal{H}^{n+2n+r S_{M-1|M}} b_{S_{M-1|M}}$$

$$= \left( P_{M-1} - P_{M-1} A_{M-1}^\top P_{M|M-1}^{-1} A_{M-1} - P_{M-1} \right) P_{M-1}^{-1} \hat{x}_{M-1} + \hat{x}_{M-1} A_{M-1}^\top P_{M|M-1}^{-1} \hat{x}_M$$

$$= \hat{x}_{M-1} + P_{M-1} A_{M-1}^\top P_{M|M-1}^{-1} \left( \hat{x}_M - A_{M-1} \hat{x}_{M-1} \right).$$

Now, if increase the horizon length $H_l$ to 2, the MHE for the LTV system (2.1) becomes

$$Z_{M-2|M}^* := \arg \min_{Z_{M-2|M}} \frac{1}{2} x_{M-2}^2 - \hat{x}_{M-2}^2 \bigg| P_{M-2}^{-1} + \frac{1}{2} \sum_{k=M-1}^M \left( \|v_k\|^2 R^{-1} + \beta_k^\top (y_k - C_k x_k - H_k v_k) \right)$$

$$+ \frac{1}{2} \sum_{k=M-2}^{M-1} \left( \|w_k\|^2 Q^{-1} + \alpha_k^\top (x_{k+1}^* - A_k x_k - G_k w_k) \right).$$

Since $\hat{x}_{M-1|M}$, $\hat{x}_k^M$ and $(P_{k})^M_{k=M-1}$ are given, $(v_k)^M_{k=M-1}$ and $w_{M-1}$ are deterministic, $(\beta_k)^M_{k=M-1}$ and $\alpha_{M-1}$ are free variable, hence we have

$$Z_{M-2|2}^* := \arg \min_{Z_{M-2|2}} S_{M-2|2} = \arg \min_{Z_{M-2|2}} \frac{1}{2} x_{M-2}^2 - \hat{x}_{M-2}^2 \bigg| P_{M-2}^{-1}$$

$$+ \left[ \frac{1}{2} \|w_{M-2}\|^2 Q^{-1} + \alpha_{M-2}^\top \left( \hat{x}_{M-1|M} - A_{M-2} x_{M-2} - G_{M-2} w_{M-2} \right) \right],$$

and

$$\hat{x}_{M-2|M} = \hat{x}_{M-2} + P_{M-2} A_{M-2}^\top P_{M-1|M-2}^{-1} \left( \hat{x}_{M-1|M} - A_{M-2} \hat{x}_{M-2} \right).$$
We define
\[ \hat{x}_{k|M} = \hat{x}_k + P_k A_k^t P^{-1}_{k+1|k}(\hat{x}_{k+1|M} - A_k \hat{x}_k), \]
which is the same as the Kalman smoother algorithm (2.9).

**Proposition 2.6.** For the nonlinear system (2.16), where \( G_k \) and \( H_k \) are full column rank matrices, given output measurements \( (y_k)_{k=1}^k \), \( 1 < k \leq M \), if Assumptions 2.1 and 2.5 hold, then state estimation using the FIE (2.23) will have a smaller estimation error \( e_k \) compared to the EKF.

**Proof.** The proof is based on [19]. By involving Lagrange multipliers \( (\alpha_k)_{k=1}^{k-1}, (\beta_k)_{k=1}^k \) and \( \gamma_1 \) for equality constraints and the initial condition, respectively, the FIE problem (2.23) can be written as the following unconstrained optimization problem:

\[
Z_k^* := \arg \min_{Z_k} \bar{S}_k = \arg \min_{Z_k} \frac{1}{2} \left\{ \| e_1 \|^2_{P_0^{-1}} + 2\gamma_1 (\hat{x}_{1|0} - x_1 - e_1) \right. \\
+ \sum_{k=1}^{k-1} \left[ \| w_k \|^2_{Q^{-1}} + 2\alpha_k^T (x_{k+1} - f(x_k) - G_k w_k) \right] + \sum_{k=1}^k \left[ \| v_k \|^2_{Q^{-1}} + 2\beta_k^T (y_k - h(x_k) - H_k v_k) \right] \right\},
\]

We define \( (\bar{S}_k)_{k=1}^k \) as

\[
\bar{S}_1 := \frac{1}{2} \| e_1 \|^2_{P_0^{-1}} + \gamma_1 (\hat{x}_{1|0} - x_1 - e_1) + \frac{1}{2} \| v_1 \|^2_{Q^{-1}} + \beta_1^T [y_1 - h(x_1) - H_1 v_1], \\
\bar{S}_2 := \bar{S}_1 + \frac{1}{2} \| v_2 \|^2_{Q^{-1}} + \beta_1^T [y_2 - h(x_2) - H_2 v_2] + \frac{1}{2} \| w_1 \|^2_{Q^{-1}} + \alpha_1^T [x_2 - f(x_1) - G_1 w_1], \\
\vdots \\
\bar{S}_k = \bar{S}_{k-1} + \frac{1}{2} \| v_k \|^2_{Q^{-1}} + \beta_k^T [y_k - h(x_k) - H_k v_k] + \frac{1}{2} \| w_{k-1} \|^2_{Q^{-1}} + \alpha_k^T [x_k - f(x_{k-1}) - G_{k-1} w_{k-1}].
\]

Because (2.41) is an unconstrained nonlinear optimization problem, in order to ensure the convergence of this nonlinear optimization, one could use a backtracking line search method combined with Newton’s method, so the optimum\(^2\) of (2.41) can be found by recursively calculating

\[
Z_k^{(i+1)} = Z_k^{(i)} - a_i \mathcal{H}_k^{-1} \left( Z_k^{(i)} \right) f_k \left( Z_k^{(i)} \right),
\]

\(^2\)In order to prevent converging to a stationary point rather than a minimizer, negative curvature information from the Hessian \( \mathcal{H}_k \) may be required [32, p. 40]
given an initial guess \( z_k^{(1)} \), until \( z_k^{(j+1)} \) has converged to its optimal value \( z_k^* \), where the scalar \( a_i \) is called step length, which is determined using a backtracking line search method, with initial value \( a_1 = 1 \) [32, p. 37]. The expressions of \( \mathcal{J}_k(z_k) \) and \( \mathcal{H}_k(z_k) \) are given by

\[
\mathcal{J}_k(z_k) = \begin{bmatrix}
\frac{\partial s_k}{\partial e_k} \\
\vdots \\
\frac{\partial s_k}{\partial e_k}
\end{bmatrix},
\mathcal{H}_k(z_k) = \begin{bmatrix}
\frac{\partial s_{k-1}}{\partial e_{k-1}} \\
\vdots \\
\frac{\partial s_{k-1}}{\partial e_{k-1}}
\end{bmatrix}
\]

and

\[
\mathcal{H}_k(z_k) \approx \begin{bmatrix}
\mathcal{J}_k(z_k) & 0 \\
0 & Q^{-1} \\
0 & 0 & R^{-1}
\end{bmatrix}
\]

where all the second derivative of functions \( f(x_k) \) and \( h(x_k) \) are ignored.

Since the system dynamics is nonlinear, unlike Proposition 2.4, the global optimum \( z_k^* \) is not guaranteed to be found and the optimization usually takes several iterations to find a local optimum. Picking an initial guess of the \( z_k^* \) based on the previous estimated states \( (x_k)_{k=1}, \)
such that

$$Z^{(1)}_k := \begin{bmatrix} 0_{1,2n+q+p} \hat{x}_1^T & 0_{1,r+q+n+p} \hat{x}_2^T & \cdots & 0_{1,r+q+n+p} \hat{x}_k^T \end{bmatrix}^T.$$

where $\hat{x}_k^T := \hat{x}_{k|k-1}^T$. By substituting $Z^{(1)}_k$ into $J_{\tilde{S}}(Z_k)$, gives

$$J_{\tilde{S}}\left(Z^{(1)}_k\right) = \begin{bmatrix} 0_{1,n+q} \left(\hat{x}_1^T - \hat{x}_1\right) \quad y_1^T - h^T(\hat{x}_1) & 0_{1,n+r+q} \hat{x}_2^T - f^T(\hat{x}_1) \\
\quad & y_2^T - h^T(\hat{x}_2) \quad \cdots \quad 0_{1,2n+r+q} y_k^T - h^T(\hat{x}_{k|k-1}) \quad 0_{1,n} \end{bmatrix}^T.$$

Substituting (2.15c) into $J_{\tilde{S}}(Z^{(1)}_k)$ will yield (2.39). We then repeat the same procedures as in Proposition 2.4 and get

$$P_k = P_{k|k-1} - P_{k|k-1}C_k^T C_k^{-1} \left(P_{k|k-1}C_k^T + H_kR_kH_k^T \right)^{-1} C_k P_{k|k-1},$$

$$\hat{x}_k^{(2)} = \mathcal{M}_{N_{k-n+1}^k} \left(Z_k^{(2)} = \hat{x}_{k|k-1} + L_k \left(y_k - h(\hat{x}_{k|k-1}) \right) \right).$$

Therefore, rather than iterating until we find a local optimum, the EKF is equivalent to solving the FIE problem (2.23) by only one backtracking line search step with a carefully chosen initial guess $Z^{(1)}_k$ and ignoring all second derivatives. Hence, by optimality, the state estimate using FIE will have a smaller estimation error $e_k$ than the EKF. \qed

### 2.8 Auto-covariance Least Squares for LTI Systems

Kalman-based state estimators, as well as FIE, assume a priori knowledge of the covariance matrices of the process and observation noise. However, in most practical situations, noise statistics are often unknown and need to be estimated from measurement data.

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

An algorithm for noise covariance estimation of LTI systems, which is a constrained auto-covariance least-squares (ALS) method inspired by the innovation correlation method of [28] was presented in [33]. The method estimates noise covariance matrices using least-
Review on State Estimation and Noise Covariance Identification

squares semi-definite programming (SDP), rather than solving an unconstrained optimization problem, which greatly reduces the variance of the estimation compared to the innovation correlation method. A simplified version of the ALS method, called the “one-column” ALS method was introduced in [36]. The computational complexity of constructing the ALS problem for high dimensional systems can be significantly reduced with this method. We are going to briefly review the “one-column” ALS method introduced in [36].

Assumption 2.6. The noise sequences \((w_k)_{k=1}^M\) and \((v_k)_{k=1}^M\) are two uncorrelated random variables having Gaussian (or normal) distributions \(N(0, Q)\) and \(N(0, R)\), respectively, with zero mean and unknown positive-definite covariance matrices \(Q\) and \(R\).

Theorem 2.5. [26] For the LTI system (2.11), If Assumption 2.3 holds, then there exists a constant steady-state filter gain \(L_\infty\) such that all trajectories of the closed-loop LTI system

\[ x_{k+1} := (A - AL_\infty C)x_k \]

are asymptotically stable.

Since the true noise covariance matrices \(Q\) and \(R\) are unknown, it is not possible to design a best linear unbiased estimator (BLUE) state estimator with an optimal steady-state filter gain \(L_\infty\). Instead, we calculate the state error covariance \(P_\infty^s\), hence the sub-optimal filter gain \(L_\infty^s\) using (2.12) and (2.13), respectively, with guessed noise covariance matrices \(Q_g\) and \(R_g\), as well as any appropriate given initial state guess \(\hat{x}_{1|0}\) and given output measurement sequence \((y_k)_{k=1}^M\) to obtain the estimated state sequence

\[ \hat{x}_k := \hat{x}_{k|k-1} + L_\infty^s (y_k - C\hat{x}_{k|k-1}), \quad k = 1, \cdots, M. \]

The state error terms are defined as

\[ \varepsilon_k := x_k - \hat{x}_{k|k-1}, \quad k = 1, \cdots, M. \]

Applying the steady-state Kalman filter yields

\[ \varepsilon_{k+1} = (A - AL_\infty^s C)\varepsilon_k + \begin{bmatrix} G - AL_\infty^s H \\ \bar{G} \end{bmatrix} \begin{bmatrix} w_k \\ v_k \end{bmatrix}. \]
Therefore, if Assumption 2.2 holds, then the steady-state Kalman filter will ensure that $\mathbb{E}\{e_k\} = 0$ for any $k$, hence we have

$$P^s_\infty = C\{e_k\} = \mathbb{E}\{e_{k+1}e_{k+1}^\top\} = \tilde{A}P^s_\infty \tilde{A}^\top + \tilde{G} \begin{bmatrix} Q_g & 0 \\ 0 & R_g \end{bmatrix} \tilde{G}^\top. \quad (2.42)$$

We define the state space model of innovations as

$$z_k := C\epsilon_k + Hv_k.$$

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 [27], [23]. 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:

$$\mathcal{C}_j(z_k) := \mathbb{E}\{(z_{k+j} - \mathbb{E}\{z_{k+j}\})(z_k - \mathbb{E}\{z_k\})^\top\} = \mathbb{E}\{z_{k+j}z_k^\top\}$$

for $j = 0, 1, \cdots, N - 1$, where $N$ is the maximum number of time lags; hence, the auto-covariance matrix (ACM) of $z_1$ can now be defined as a function of $P^s_\infty$ and $R_g$

$$\mathcal{R}_1 := \mathbb{E}\begin{bmatrix} z_1z_1^\top \\ z_2z_1^\top \\ \vdots \\ z_Nz_1^\top \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ C\tilde{A}^{N-1} \end{bmatrix} P^s_\infty C^\top + \begin{bmatrix} H \\ -CA\tilde{A}sH \\ \vdots \\ -C\tilde{A}^{N-2}sH \end{bmatrix} R_g. \quad (2.43)$$

By vectorizing $P^s_\infty$ and $\mathcal{R}_1$ in (2.42) and (2.43), respectively, we have

$$(P^s_\infty)_s = (\tilde{A} \otimes \tilde{A}) (P^s_\infty)_s + (G \otimes G) (Q_g)_s + (AL^s_\infty H \otimes AL^s_\infty H) (R_g)_s, \quad (2.44a)$$

$$(\mathcal{R}_1)_s = (C \otimes \Upsilon) (P^s_\infty)_s + (I_q \otimes \Xi) (R_g)_s. \quad (2.44b)$$

Substituting (2.44a) into (2.44b) yields

$$(\mathcal{R}_1)_s = (C \otimes \Upsilon) \left( I_q^2 - \tilde{A} \otimes \tilde{A} \right)^{-1} (G \otimes G) (Q_g)_s + \left[ (C \otimes \Upsilon) \left( I_q^2 - \tilde{A} \otimes \tilde{A} \right)^{-1} (AL^s_\infty H \otimes AL^s_\infty H) + (I_q \otimes \Xi) \right] (R_g)_s.$$
Next, we define the sample estimate of $R_1$ as

$$\bar{R}_1 := \frac{1}{M_b - N + 1} \times \begin{bmatrix} \bar{z}_1 & \bar{z}_2 & \ldots & \bar{z}_{M_b - N + 1} \\ \bar{z}_2 & \bar{z}_3 & \ldots & \bar{z}_{M_b - N + 2} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{z}_N & \bar{z}_{N + 1} & \ldots & \bar{z}_{M_b} \end{bmatrix} \times \begin{bmatrix} \bar{z}_1^T \\ \bar{z}_2^T \\ \vdots \\ \bar{z}_{M_b - N + 1}^T \end{bmatrix},$$

where

$$\bar{z}_k := y_k - C\hat{x}_{k|k-1}$$

represents the actual innovation terms given the output measurement $y_k$ and estimated state $\hat{x}_{k|k-1}$. $M_b$ is the data length for estimating $\bar{R}_1$ with $N \ll M_b \leq M$ and $C \{ \bar{R}_1 \} \rightarrow 0$ as $M_b$ goes to infinity [33].

The true covariance $Q$ and $R$ can be estimated by solving the following constrained least squares optimization problem:

$$((\hat{Q})_{ss}, (\hat{R})_{ss}) := \arg \min_{\vartheta} \left\| \begin{bmatrix} \mathcal{A}_{Qg} & \mathcal{A}_{Rg} \end{bmatrix} \begin{bmatrix} (Q)_{ss} \\ (R)_{ss} \end{bmatrix} - (\bar{R}_1)_s \right\|^2_{2}$$

subject to $(Q, R) \succ 0$. (2.45)

where

$$\mathcal{A}_{Qg} := \left( C \otimes \Upsilon \right) \left( I_{n_2} - \tilde{A} \otimes \tilde{A} \right)^{-1} \left( G \otimes G \right) \mathcal{D}_r,$$

$$\mathcal{A}_{Rg} := \left( C \otimes \Upsilon \right) \left( I_{n_2} - \tilde{A} \otimes \tilde{A} \right)^{-1} \left( AL_s H \otimes AL_s H \right) + \left( I_q \otimes \Xi \right) \mathcal{D}_q$$

and matrices $\mathcal{D}_r$ and $\mathcal{D}_q$ are defined in the Nomenclature. The solution of (2.45) exists for all $(\bar{R}_1)_s$; the solution is unique if and only if $\left[ \mathcal{A}_{Qg} \quad \mathcal{A}_{Rg} \right]$ has full column rank [36]. The estimates $\hat{Q}$ and $\hat{R}$ of (2.45) are unbiased for all sample sizes and converge asymptotically to the true covariances $Q$ and $R$ as $M_b$ goes to infinity [33]. Instead of using the identity matrix as the weight in the least-squares problem (2.45), a method was also proposed in [36] to calculate the optimal weighting for further minimizing the variance of the estimation error.

### 2.9 Expectation Maximization for Linear Systems

In addition to the ALS method, the noise covariances can also be identified using the expectation maximization (EM) method, which is based on the maximum likelihood estimation
(MLE), and was first introduced in [11]. The MLE aims to maximize the likelihood of the noise covariance given the output sequence. A Kalman filter-based recursive algorithm for estimating noise covariances for linear systems using the EM method is presented in [48], where \( G_k \) and \( H_k \) are both identity matrices.

**Theorem 2.6.** [2] For the LTV system (2.1) given output measurements \((y_k)_{k=1}^M\), if Assumption 2.2 holds and 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

\[
x_{k+1} := (A_k - A_kL_kC_k)x_k
\]

are exponentially stable over a finite horizon \(k = 1, \ldots, M\).

For the LTV system (2.1), given output measurements \((y_k)_{k=1}^M\), if we select \(\hat{x}_{1|0}, P_{1|0}, Q_1 = Q_g\) and \(R_1 = R_g\) as the initial guess of \(\hat{x}_{1|0}, P_{1|0}, Q\) and \(R\), respectively, the set of true system parameters \(\Theta := \{\hat{x}_{1|0}, P_{1|0}, Q, R\}\) can be recursively estimated using the expectation maximization (EM) method [48].

Let the estimate of \(\hat{x}_{1|0}\) and covariance matrices \(P_{1|0}, Q\) and \(R\) at the \(i\)th iteration be

\[
\Theta_i := (\hat{x}_{1|0,i}, P_{1|0,i}, Q_i, R_i), \quad i = 2, \ldots, N,
\]

where \(N \gg 2\) is the maximum number of iterations. The basic idea of the EM method is to recursively maximize the expectation of the log-likelihood function \(E(\Theta|y_{1:M})\), until the log-likelihood function converges to its maximum value [11].

The expression of the log-likelihood function \(E(\Theta|y_{1:M})\) is given by

\[
\mathcal{L}_\ell(\Theta|y_{1:M}) = \log(p(y_{1:M}|\Theta)) = \log \left( \frac{p(x_{1:M}, y_{1:M}|\Theta)}{p(x_{1:M}|y_{1:M}, \Theta)} \right) = \log(p(x_{1:M}, y_{1:M}|\Theta)) - \log(p(x_{1:M}|y_{1:M}, \Theta)).
\] (2.46)

Taking the conditional expectation on both sides of (2.46) given \(y_{1:M}\) and \(\Theta_{i-1}\), we get the expectation of the log-likelihood function \(E(\Theta|y_{1:M})\)

\[
E\{\log(p(y_{1:M}|\Theta))|y_{1:M}, \Theta_{i-1}\} = \Omega(\Theta|\Theta_{i-1}) - \mathcal{J}(\Theta|\Theta_{i-1}),
\]

where \(\Omega\) and \(\mathcal{J}\) are given by

\[
\Omega(\Theta|\Theta_{i-1}) := E\{\log(p(x_{1:M}|y_{1:M}|\Theta))|y_{1:M}, \Theta_{i-1}\}, \quad (2.47a)
\]

\[
\mathcal{J}(\Theta|\Theta_{i-1}) := E\{\log(p(x_{1:M}|y_{1:M}, \Theta))|y_{1:M}, \Theta_{i-1}\}. \quad (2.47b)
\]
Because $y_{1:M}$ is a given measurement sequence, it follows that
\[
\mathbb{E}\{\log(p(y_{1:M}|\Theta))|y_{1:M}, \Theta_{i-1}\} = \mathcal{L}_\ell(\Theta|y_{1:M}) = \Omega(\Theta|\Theta_{i-1}) - \partial(\Theta|\Theta_{i-1}).
\]

**Theorem 2.7.** [11, 53] For the EM method, the value of $\mathcal{L}_\ell(\Theta|y_{1:M})$ will monotonically increase at each iteration and converge to the maximum if
\[
\Omega(\Theta_{i}|\Theta_{i-1}) \geq \Omega(\Theta_{i-1}|\Theta_{i-1}), \quad \forall i
\]
with equality if and only if $\Theta_{i} = \Theta_{i-1}$.

*Proof.* This theorem is a key contribution\(^3\) of [11]. However, the proof of convergence of the EM sequences in [11] was flawed; a rigorous proof of the convergence of $\mathcal{L}_\ell(\cdot|y_{1:M})$ as well as the sequence $(\Theta_{i})_{i=1}^{N}$ is given in [53].

Theorem 2.7 simplifies calculation, so that we only need to focus on fulfilling (2.48) by maximizing $\Omega(\Theta|\Theta_{i-1})$, instead of $\mathcal{L}_\ell(\cdot|y_{M})$. Before giving the expression of $\Omega$, we start with the expression of $\log(p(x_{1:M}, y_{1:M}|\Theta))$ in (2.47a).

**Proposition 2.7.** if Assumptions 2.1 and 2.5 hold, then $\log(p(x_{1:M}, y_{1:M}|\Theta))$ can be written as a function of $x_{1}$, $(w_k)_{k=1}^{M}$, $(v_k)_{k=1}^{M}$ and $\Theta$, such that
\[
\log(p(x_{1:M}, y_{1:M}|\Theta)) = \mathcal{L}_\ell(\Theta|x_{1:M}, y_{1:M}) = -\frac{1}{2} (s \log(2\pi) + \log |P_{1}| + M \log |R| + M \log |Q|) - \frac{1}{2} (x_{1} - \hat{x}_{1|0})^T P_{1|0}^{-1} (x_{1} - \hat{x}_{1|0}) - \frac{1}{2} \sum_{k=1}^{M} v_{k}^T R^{-1} v_{k} - \frac{1}{2} \sum_{k=1}^{M-1} w_{k}^T Q^{-1} w_{k}.
\]

*Proof.* By using the relation between the joined and conditional probability stated in the Nomenclature, we have $\log(p(x_{1:M}, y_{1:M}|\Theta)) \propto \log(p(x_{1:M}|y_{1:M}, \Theta))$, then Lemma 2.4 and (2.22) gives (2.49).

**Proposition 2.8.** For the LTV system (2.1) with $(G_k)_{k=1}^{M}$ and $(H_k)_{k=1}^{M}$ nonsingular, given output measurements $(y_k)_{k=1}^{M}$, the function $\Omega$ defined in (2.47a) can be written as a function of $\Theta$, $A_k$, $C_k$, $\hat{x}_{k|\Theta}$, $P_{k|\Theta}$ and $P_{k+1|\Theta}$, where
\[
P_{k+1|M} := \mathbb{E}\left\{ (x_{k} - \hat{x}_{k|M}) (x_{k-1} - \hat{x}_{k-1|M})^T \right\}.
\]

\(^3\)See Appendix B for the proof of monotonically increasing of $\mathcal{L}_\ell(\Theta|y_{1:M})$. 

Proof. Taking the conditional expectation over (2.49) gives

\[
\begin{align*}
\mathcal{Q}(\theta | \theta_{i-1}) &= \mathbb{E}\{\log(p(x_{1:M}, y_{1:M} | \theta')) | y_{1:M}, \theta_{i-1}\} = \\
& \quad - \frac{1}{2} \left[ s \log(2\pi) + \log|P_{1|0}| + M \log|Q| + M \log|R| \right] - \frac{1}{2} \text{tr} \left\{ P_{1|0}^{-1} \mathbb{E} \left[ \varepsilon_1 \varepsilon_1^\top | y_{1:M}, \theta_{i-1} \right] \right\} \\
& \quad + Q^{-1} \sum_{k=1}^{M-1} \mathbb{E} \left[ w_k w_k^\top | y_{1:M}, \theta_{i-1} \right] - R^{-1} \sum_{k=1}^{M} \mathbb{E} \left[ v_k v_k^\top | y_{1:M}, \theta_{i-1} \right] \right\},
\end{align*}
\]

(2.51)

where \(w_k = x_{k+1} - A_k x_k\) and \(v_k = y_k - C_k x_k\). Recalling the definition of the smoothed state error covariance \(P_{k|M}\) and lag-one smoothed state error covariance \(P_{k,k-1|M}\) in (2.9b) and (2.50), respectively, we have

\[
P_{k|M} = \mathbb{E} \left\{ x_k x_k^\top | y_{1:M}, \theta_{i-1} \right\} - \hat{x}_{k|M} \hat{x}_{k|M}^\top,
\]

(2.52)

\[
P_{k,k-1|M} = \mathbb{E} \left\{ x_k x_{k-1}^\top | y_{1:M}, \theta_{i-1} \right\} - \hat{x}_{k|M} \hat{x}_{k-1|M}^\top.
\]

Substituting (2.52) into (2.51) gives

\[
\begin{align*}
\mathcal{Q}(\theta | \theta_{i-1}) &= \mathbb{E}\{\log(p(x_{1:M}, y_{1:M} | \theta')) | y_{1:M}, \theta_{i-1}\} = \\
& \quad - \frac{1}{2} \left[ s \log(2\pi) + \log|P_{1|0}| + (M - 1) \log|Q| + M \log|R| \right] \\
& \quad - \frac{1}{2} \text{tr} \left\{ P_{1|0}^{-1} \left[ P_{1|M} + (\hat{x}_{1|0} - \hat{x}_{1|M}) (\hat{x}_{1|0} - \hat{x}_{1|M})^\top \right] \right\} \\
& \quad - \frac{1}{2} \text{tr} \left\{ Q^{-1} \sum_{k=1}^{M-1} \left[ P_{k|M} + \hat{x}_{k|M} \hat{x}_{k|M}^\top - \left( P_{k,k-1|M} + \hat{x}_{k|M} \hat{x}_{k-1|M}^\top \right) A_{k-1}^\top \right] \\
& \quad - A_{k-1} \left( \hat{x}_{k-1|M} \hat{x}_{k-1|M}^\top + P_{k,k-1|M} \right) + A_{k-1} \left( P_{k-1|M} + \hat{x}_{k-1|M} \hat{x}_{k-1|M}^\top \right) A_{k-1}^\top \right\} \\
& \quad - \frac{1}{2} \text{tr} \left\{ R^{-1} \sum_{k=1}^{M} \left[ (y_k - C_k \hat{x}_{k|M}) (y_k - C_k \hat{x}_{k|M})^\top + C_k P_{k|M} C_k^\top \right] \right\},
\end{align*}
\]

where the “lag-one” smoothed state error covariance \(P_{k,k-1|M}\) can be recursively calculated by (2.53) (more details are given in Appendix C):

\[
P_{k-1,k-2|M} = P_{k-1} U_{k-2}^\top + U_{k-1} \left( P_{k,k-1|M} - A_{k-1} P_{k-1} \right) U_{k-2}^\top, \quad k = M, \ldots, 2
\]

(2.53a)

\[
P_{M,M-1|M} = (I_n - L_M C_M) A_{M-1} P_{M-1}.
\]

(2.53b)
Because \( \hat{x}_{1|0,i} \) and \( P_{1|0,i} \) cannot be identified simultaneously, we let \( \hat{x}_{1|0,i} = \hat{x}_{1|M} \) and \( P_{1|0,i} = P_{1|M} \) [48]. Thus, if the sequences \( (\hat{x}_{k|M})_{k=1}^{M} \), \( (P_{k+1,k|M})_{k=1}^{M-1} \), and \( (P_{k|M})_{k=1}^{M} \) are all computed using (2.9) and (2.53), respectively, then the parameter set \( O_i \) can be estimated by iteratively solving

\[
O_i = \arg \max_{\hat{x}_{1|0}, P_{1|0}, Q, R} Q(O_i | O_{i-1})
\]

s.t. \( (P_{1|0}, Q, R) \prec 0 \) (2.54)

until both \( \|Q_{i+1} - Q_i\|_F \) and \( \|R_{i+1} - R_i\|_F \) are smaller than a user-defined positive threshold \( \zeta \), \( \zeta \ll 1 \).

### 2.10 Conclusions

In the chapter, we started with the classic derivations of the Kalman filter and smoother for LTV systems, followed by its two famous variants, the steady state Kalman filter for LTI systems and the extended Kalman filter for nonlinear systems. For unconstrained linear systems, if all given information is accurate, then the Kalman filter and smoother are the best linear unbiased estimators of the unknown system states. However, for a nonlinear system, both estimation accuracy and stability are not guaranteed, which is due to the model mismatch caused by first order model linearization. In order to improve the performance and stability of nonlinear state estimation and be compatible with system constraints, we briefly reviewed the optimization-based full information estimation and its fixed-length version, namely moving horizon estimation.

After a brief introduction of different state estimation methods, we used the definition of singular joint probability density to prove that the joint probability density function of a full column rank matrix \( G \) multiplying with a normal distributed random vector \( w \) is directly proportional to the joint probability density function of that random vector \( w \). We then compared the FIE with Kalman based algorithms for both linear and nonlinear systems. We proved that, for a linear system, the FIE is a BLUE of the unknown system states, if initial and noise statistics are accurate. The FIE and Kalman filter/smoother are completely equivalent for a linear system and FIE will provide better estimates for nonlinear systems compared to the EKF.

Finally, we summarized two existing noise covariance estimation algorithms, the auto-covariances estimation method for LTI systems [33] and the expectation maximization method for both LTV and LTI systems [48]. The ALS method estimates noise covariances by solving one optimization, whereas the EM method is an iterative estimation method.
For LTI systems, both ALS and EM methods are easy to implement, and are able to provide accurate noise covariance estimations. However, the EM method is an iterative algorithm compared to “one-off” ALS estimation methods, hence the EM method may be inefficient for covariance estimation of some LTI systems.
Chapter 3

Noise Covariance Identification for Linear Time-Varying Systems using the Auto-covariance Least Squares Method

Kalman-based state estimators assume a priori knowledge of the covariance matrices of the process and observation noise as well as the initial conditions (initial state and corresponding error covariance). However, in most practical situations, noise statistics and initial conditions are often unknown and need to be estimated from measurement data. Incorrect noise covariances in the design of a Kalman filter could result in large estimation errors or even a divergence of state estimates. In this chapter, we present an auto-covariance least squares based algorithm for noise covariance and initial condition estimation of linear time-varying systems.

In Section 2.8 we have briefly introduced the ALS-based noise covariance estimation methods for LTI models, given by [33, 36]. The standard ALS method was extended to linear time-varying (LTV) and nonlinear systems in [35]. However, due to the structure and approximations, if the historical data is not sufficiently long, the existing algorithm may not

---

I am the poet of the Body and I am the poet of the Soul, The pleasures of heaven are with me and the pains of hell are with me. The first I graft and increase upon myself, the latter I translate into a new tongue.

*Song of Myself*

Walter Whitman
be able to provide accurate estimates. We therefore provide a modified algorithm for noise covariance estimation by removing all approximations in the existing formulation, so the structure of our algorithm becomes simpler, has fewer parameters to determine and is able to provide more accurate results.

Our algorithm also provides the estimate of initial state error covariance $P_{1|0}$, which is required in most state estimation algorithms. For high-order systems, we provide a less memory demanding formulation by splitting large Kronecker products with sums of smaller Kronecker products, so that our ALS algorithm can be applied to real-world applications.

Both the ALS and EM algorithms introduced in Section 2.9 could provide good noise covariance estimates for LTV systems. The ALS method has more parameters to tune and it is more complicated to construct the estimation problem compared to the EM method. However, noise covariance estimation using the EM algorithm could be time consuming and may fail to converge to the globally optimal value after many iterations if starting from a bad initial guess [34, p. 125].

### 3.1 Auto-covariance Least Squares for LTV Systems

Recall the discrete-time LTV model (2.1)

$$x_{k+1} = A_k x_k + G_k w_k$$
$$y_k = C_k x_k + H_k v_k$$

, where Assumptions 2.2 and 2.6 are hold.

**Theorem 3.1.** [2] If Assumption 2.2 holds and 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

$$x_{k+1} := (A_k - A_k L_k C_k) x_k$$

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

Since the true noise covariance matrices $Q$ and $R$ are unknown, Theorem 2.4 shows that 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 $(L^s_k)_{k=1}^M$ and any appropriate given initial state guess $\hat{x}_{1|0}$ to obtain the estimated state sequence

$$\hat{x}_k := \hat{x}_{k|k} := \hat{x}_{k|k-1} + L^s_k (y_k - \hat{y}_{k|k-1})$$

(3.1a)
where \( \hat{y}_{k|k-1} \) are the 1-step ahead predicted output, given by

\[
\hat{y}_{k|k-1} := C_k \hat{x}_{k|k-1}.
\]

(3.1b)

Hence

\[
\hat{x}_{k+1|k} = A_k (\hat{x}_{k|k-1} + L_k^s (y_k - \hat{y}_{k|k-1}))
\]

\[= A_k \hat{x}_{k|k-1} + A_k L_k^s (C_k x_k + H_k v_k - C_k \hat{x}_{k|k-1})
\]

and

\[
\varepsilon_{k+1} = (A_k - A_k L_k^s C_k) \varepsilon_k + \begin{bmatrix} G_k & -A_k L_k^s H_k \end{bmatrix} \begin{bmatrix} w_k \\ \bar{v}_k \end{bmatrix}.
\]

(3.2a)

Therefore, if Assumption 2.2 holds, Theorem 3.1 will ensure that \( \mathbb{E}[\varepsilon_k] \to 0 \) as \( k \to \infty \).

We define the state space model of innovations as

\[
z_k := C_k \varepsilon_k + H_k v_k.
\]

(3.2b)

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 [23], [27]. 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:

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

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 3.1.** Given output measurements \( (y_k)_{k=1}^\kappa \), with large enough \( \kappa, \kappa \leq M \), there exists a smoothed initial state \( \hat{x}_{1|\kappa} \), such that if let \( \hat{x}_{1|0} = \hat{x}_{1|\kappa} \), then the expectation of the initial state error term \( \varepsilon_1 \) will be zero.

For any \( k \), the state error term \( \varepsilon_{k+1} \) is a function of \( \varepsilon_1 \) and \( (\bar{w}_k)_{k=1}^M \), hence, Assumption 3.1 and (3.2b) ensure that

\[
\forall k, j : \mu_{k+j} = 0 \quad \Rightarrow \quad C_j(z_k) = \mathbb{E}[z_{k+j}z_k^\top].
\]
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

$$C_j \left( (z_{k+1})_{k=1}^{N_z} \right) = \mathbb{E} \left[ z_{2+j} \cdots z_{M_e-N+2+j} \right] \in \mathbb{R}^{N_z \times N_z},$$

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

$$\mathcal{R} := \begin{bmatrix} C_0^\top \left( (z_{k+1})_{k=1}^{N_z} \right) & C_1^\top \left( (z_{k+1})_{k=1}^{N_z} \right) & \cdots & C_{N_e-1}^\top \left( (z_{k+1})_{k=1}^{N_z} \right) \end{bmatrix}. $$

We also define matrix $\mathcal{R}_i$ as

$$\mathcal{R}_i := \mathbb{E} \left[ z_{2+i} \cdots z_{M_e-N+i} \right], \quad i = 0, \ldots, M_e - N,$$

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 $P_{1|0}$; see (3.6) 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 way as in [36]. Instead, we define the sample estimate of $\mathcal{R}$ as

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

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

$$\bar{z}_k := y_k - \hat{y}_{k|k-1}. \quad (3.3)$$

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

$$(P^*_1, Q^*, R^*) := \arg \min_{P_{1|0}, Q, R} \left\| \mathcal{R}(P_{1|0}, Q, R) - \tilde{\mathcal{R}} \right\|_F^2. \quad (3.4)$$

Compared to LTI models, the estimated error covariance $P_{k|k-1} := \mathbb{E}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^\top]$ 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 [33].

### 3.2 Solving the Optimization Problem

In order to apply the Kalman filter, a guess of the initial state error covariance $P_1 := P_g > 0$ and guessed noise covariances $Q_g$ and $R_g$ have to be provided, hence the sub-optimal filter gains $L_k$ can be recursively obtained from the Kalman filter equations (2.4). Let us start from $e_1$ and consider the evolution of (3.2a) and (3.2b). The innovation sequence $(z_k)_{k=1}^{M_e}$ can be shown to be given by (more details are given in Appendix D)

$$ z = \tilde{\mathbf{v}} (\tilde{e}_1 + \tilde{g}_i w) + \tilde{\mathbf{f}} \bar{v}, \quad (3.5) $$

where

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

$$ \tilde{\mathbf{v}} := \begin{bmatrix} v_2^T & v_3^T & \cdots & v_{M_e}^T & v_{M_e+1}^T \end{bmatrix}^T, \quad \tilde{\mathbf{A}} := \bar{I}_{M_e} - \begin{bmatrix} 0 & 0 \\ \mathcal{M}_{k=2}^{M_e} \tilde{A}_k & 0 \end{bmatrix}, $$

$$ \tilde{\mathbf{c}} := \mathcal{M}_{k=2}^{M_e+1} C_k, \quad \mathcal{V} := \tilde{\mathbf{c}} \tilde{\mathbf{A}}^{-1}, \quad \tilde{\mathbf{g}} := \mathcal{M}_{k=1}^{M_e} G_k, \quad \tilde{\mathbf{f}} := \mathcal{M}_{k=1}^{M_e} H_k, \quad \tilde{\mathbf{e}} := \begin{bmatrix} \tilde{A}_1 \\ 0 \end{bmatrix}. $$

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

$$ \mathcal{R}(P_1, Q, R) = \Gamma (I_{N_c} \otimes P_{1|0}) \bar{\mathbf{f}}^T + \Omega (I_{N_c} \otimes Q) \Omega^T + \Phi (I_{N_d} \otimes R) \Phi^T + \Psi (I_{N_c} \otimes R) \Psi^T, \quad (3.6) $$

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

$$ \Gamma := \tilde{\mathbf{c}} \tilde{\mathbf{f}}, \quad \bar{\mathbf{f}} := \tilde{\mathbf{c}} \tilde{\mathbf{d}} \tilde{\mathbf{f}}, \quad \Omega := \tilde{\mathbf{c}} \tilde{\mathbf{d}} \tilde{\mathbf{f}}, \quad \bar{\mathbf{U}} := \tilde{\mathbf{c}} \tilde{\mathbf{d}} \tilde{\mathbf{f}}, \quad \bar{\mathbf{F}} := \bar{I}_{N_c} \otimes \bar{\mathbf{F}}_s, $$

$$ \tilde{\mathbf{v}} := \mathcal{M}_{k=2}^{M_e-N+2} H_k \quad \bar{\mathbf{U}} := \tilde{\mathbf{c}} \tilde{\mathbf{U}}, \quad \bar{\mathbf{F}} := \tilde{\mathbf{c}} \tilde{\mathbf{d}} \tilde{\mathbf{U}}, \quad \mathbf{V} := \begin{bmatrix} (1 \times N_c) \otimes I_p \end{bmatrix} \mathcal{P} \bar{\mathbf{U}}, $$

$$ \tilde{\mathbf{p}} := \mathcal{M}_{p(i+1)+1}^{(N-1),pM_e} \quad \tilde{\mathbf{M}} := \mathcal{M}_1^{p, pN} \quad \tilde{\mathbf{B}} := \mathcal{M}_{k=1}^{M_e} G_k \quad \tilde{\mathbf{d}} := -\tilde{\mathbf{v}} \mathcal{M}_{k=1}^{M_e} A_k L_s H_k, $$

$$ \tilde{\mathbf{g}} := \mathcal{M}_{l=0}^{(l+1),qM_e} \quad \tilde{\mathbf{u}} := \mathcal{M}_{k=0}^{M_e-N} \bar{\mathbf{U}}_l \quad \tilde{\mathbf{U}} := \mathcal{M}_{l=1}^{(l+1),qM_e} \tilde{\mathbf{u}}. $$

$$ \tilde{\mathbf{P}} := \begin{bmatrix} \bar{\mathbf{P}}_0 & \bar{\mathbf{P}}_1 & \cdots & \bar{\mathbf{P}}_{M_e-N} \end{bmatrix} = \begin{bmatrix} 0_{(N-1)p,p} & 1_{1,M_e-N} \otimes \mathcal{M}_1^{p(N-1),p(M_e+1)} & I_{(N-1)p} \end{bmatrix}. $$
Noise Covariance Identification for Linear Time-Varying Systems using the Auto-covariance Least Squares Method

\[
\tilde{S}_i := \mathcal{M}_{\beta_{i+1}}^{\beta_{i+1}}, \quad \tilde{S} := \left[ \tilde{S}_0 \ \tilde{S}_1 \ \ldots \ \tilde{S}_{M_{e}-N} \right] = \left[ I_{1,M_{e}-N} \otimes \mathcal{M}_1^{\beta_{1}} \beta_{1+1} \ \ I_{Np} \right], \\
\tilde{S}^d := \bigoplus_{i=0}^{M_{e}-N} \mathcal{M}(\tilde{S}_i) = \begin{bmatrix} I_{M_{e}-N} & \mathcal{M}_1^{\beta_{1}} \beta_{1+1} & 0 \\ 0 & \mathcal{M}_1^{\beta_{1}} \beta_{1+1} \end{bmatrix}, \quad \tilde{\Theta}_i := \left( \begin{array}{c} \mathcal{M}_{q,q}^{\beta_{1}} \beta_{1+1} \end{array} \right), \quad \tilde{\Theta} := \bigoplus_{i=0}^{M_{e}-N} \tilde{\Theta}_i.
\]

In order to fit problem (3.4) 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

\[
(\mathcal{R})_s = (\tilde{\Gamma} \otimes \Gamma)(\mathcal{J}_{N_e,n}(P_{1|0}))(Q) + [(\tilde{\Phi} \otimes \Phi)(\mathcal{J}_{N_e,q}^r) + (\tilde{\Psi} \otimes \Psi)(\mathcal{J}_{N_e,q})(R)_s.
\]

Considering the dimension and structure of matrices \( \tilde{\Gamma}, \Gamma, \tilde{\Omega}, \Omega, \tilde{\Phi}, \Phi, \tilde{\Psi} \) and \( \Psi \), 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 \( (\mathcal{R}_i)_s \) can be expressed as [35]

\[
(\mathcal{R}_i)_s = (\tilde{\Gamma}_i \otimes \Gamma_i)(\mathcal{J}_{1,n}(P_{1|0}))(Q) + [(\tilde{\Phi}_i \otimes \Phi_i)(\mathcal{J}_{1,q}^r) + (\tilde{\Psi}_i \otimes \Psi_i)(\mathcal{J}_{1,q})(R)_s,
\]

where

\[
\tilde{\Gamma}_i := \tilde{S}_i \tilde{\gamma}_d, \quad \tilde{\gamma}_d := \tilde{S}_i \tilde{\gamma}_d, \quad \tilde{\Omega}_i := \tilde{\gamma}_d \tilde{\gamma}_d, \quad \tilde{\Phi}_i := \tilde{S}_i \tilde{\gamma}_d, \quad \tilde{\Psi}_i := \tilde{S}_i \tilde{\gamma}_d,
\]

Let \( \tilde{b} := (\mathcal{R})_s \). The original optimization problem (3.4) can now be rearranged into an unconstrained least-squares problem with decision variables \( (P_{1|0})_s, (Q)_s, \) and \( (R)_s \):

\[
\min_{\tilde{\vartheta}} \left\| \begin{bmatrix} \mathcal{A}_0 \\ \vdots \\ \mathcal{A}_{M_{e}-N} \end{bmatrix} \begin{bmatrix} (P_{1|0})_s \\ (Q)_s \\ (R)_s \end{bmatrix} - \begin{bmatrix} \tilde{b}_0 \\ \vdots \\ \tilde{b}_{M_{e}-N} \end{bmatrix} \right\|_2^2 (3.8a)
\]

where, for \( i = 0, \ldots, M_{e}-N, \tilde{p}_i := (\mathcal{R}_i)_s \) and

\[
\mathcal{A} := \left[ (\tilde{\Gamma}_i \otimes \Gamma_i)(\mathcal{J}_{1,n}) \right] \mathcal{D}_n \quad (\tilde{\Omega}_i \otimes \Omega_i)(\mathcal{J}_{i+1,r}) \mathcal{D}_r \quad (\tilde{\Phi}_i \otimes \Phi_i)(\mathcal{J}_{i+1,q} + I_q \otimes \Psi_i) \mathcal{D}_q \right). (3.8b)
3.3 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 3.1, the matrix $A$ in (3.8a) 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 discussions about the effect of varying $M_e$ will be given in Section 3.5. Theorem 2.4 shows that if we obtain $\hat{x}_{1|\kappa}$ using the FIE (2.28), then Assumption 3.1 will be fulfilled regardless of the choice of covariances $Q_g$, $R_g$ and $P_g$.

Recall the matrix $A$ in the auto-covariance least squares (ALS) problem (3.8a):

$$A = \begin{bmatrix} \tilde{A}_1 & \tilde{A}_2 & (\tilde{A}_3 + \tilde{A}_5) \\ \overline{A}_3 \end{bmatrix} \in \mathbb{R}^{p_2NN_z \times (n^2+r^2+q^2)}$$

where

$$\tilde{A}_1 := (\overline{\Gamma} \otimes \Gamma)J_{N_z,n}, \quad \tilde{A}_2 := (\overline{\Omega} \otimes \Omega)J_{N_d,r}, \quad \tilde{A}_3 := (\overline{\Psi} \otimes \Psi)J_{N_z,q}, \quad \tilde{A}_4 := (\overline{\Phi} \otimes \Phi)J_{N_d,q}.$$

Assumption 3.2. The number of rows of matrix $A$ is greater than the number of columns (i.e. $NN_z \geq n^2$), $\mathcal{R}(\tilde{A}_1) \cap \mathcal{R}(\tilde{A}_2) \cap \mathcal{R}(\tilde{A}_3) = \{0\}$ and $\mathcal{R}(\tilde{A}_4) \cap \mathcal{R}(\tilde{A}_5) = \{0\}$.

Assumption 3.3. For the LTV system (2.1), $P_g$, $Q_g$ and $R_g$ are positive definite, $A_k$ is nonsingular, $C_k$ is full row rank matrix, $G_k$ and $H_k$ are full column rank matrices, for all $k$.

Theorem 3.2. For the LTV system (2.1), if Assumptions 3.2 and 3.3 hold, then matrix $A$ is full rank and (3.4) has a unique solution.

Proof. Assumption 3.3 ensures that $G_kQG_k^\top \succeq 0$ in (2.4c), so that, by the definition of positive definite matrices, if $P_{k-1} > 0$, then $P_{k|k-1} > 0$. (2.5) ensures that if $P_{k|k-1} > 0$, then $P_k > 0$. Hence, by Assumption 3.3 and [6, Corollary 8.3.6]

$$\tilde{A}_k := A_k (I - L_k^2 C_k) = A_k P_k P_{k|k-1}^{-1},$$

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

Assumptions 3.3 and [6, Fact 2.10.3] ensure that the suboptimal Kalman filter gain \( L^x_k \in \mathbb{R}^{n \times p} \) in (2.4d) is full rank \( \forall k \). According to the definition of the permutation matrix, \( J \) has full column rank.

From the definition of \( M^c_i \), the following matrices are all full rank:

\[
\tilde{\mathcal{P}}_i \in \mathbb{R}^{p(N-1) \times pM}, \quad \tilde{\mathcal{S}}_i \in \mathbb{R}^{pN \times pM}, \quad \tilde{\mathcal{O}}_i \in \mathbb{R}^{pM \times p}, \quad \tilde{\mathcal{U}}_i \in \mathbb{R}^{pM \times p(i+1)}, \quad \forall i; \\
\tilde{\mathcal{G}}_i \in \mathbb{R}^{M \times r(i+1)}, \quad \forall i; \\
\tilde{\mathcal{P}} \in \mathbb{R}^{p(N-1) \times pM_N}, \quad \tilde{\mathcal{S}} \in \mathbb{R}^{pN \times pM_N}, \quad \tilde{\mathcal{G}} \in \mathbb{R}^{pM_n \times pN}, \quad \tilde{\mathcal{U}} \in \mathbb{R}^{pM_n \times p(i+1)}.
\]

If \( \text{rank}(\hat{A}_k) = n \) ensures \( \text{rank}(\tilde{\mathcal{E}}) = n \). Assumption 3.3 and [6, Fact 2.11.13] will ensure the following matrices are full rank: \( \tilde{\mathcal{C}} \in \mathbb{R}^{pM_n \times nM_c}, \tilde{\mathcal{B}} \in \mathbb{R}^{pN_i \times qN_c}, \tilde{\mathcal{A}} \in \mathbb{R}^{nM_n \times nM_c} \)

\[
\left( \bigoplus_{k=2}^{M_c+1} G_k \right) \in \mathbb{R}^{nM_n \times rM_c}, \quad \left( \bigoplus_{k=2}^{M_c+1} A_k L^c_{kH} \right) \in \mathbb{R}^{nM_n \times qM_c}.
\]

Hence, by using [6, Fact 2.10.3] and [6, Fact 7.4.23], we have the following full rank matrices:

\[
\tilde{\mathcal{V}} \in \mathbb{R}^{pM_n \times nM_c}, \quad \tilde{\mathcal{B}} \in \mathbb{R}^{pM_n \times rM_c}, \quad \tilde{\mathcal{G}} \in \mathbb{R}^{pM_n \times rN_d}, \\
\tilde{\mathcal{F}} \in \mathbb{R}^{pM_n \times nN_c}, \quad \tilde{\mathcal{D}} \in \mathbb{R}^{rM_n \times qM_c}, \quad \tilde{\Psi} \in \mathbb{R}^{pM_n \times qN_c}.
\]

Therefore, by using [6, Fact 2.10.3] again, the following matrices are also full rank:

\[
\Gamma \in \mathbb{R}^{pN_i \times nM_c}, \quad \tilde{\Gamma} \in \mathbb{R}^{nN_i \times nM_c}, \quad \Omega \in \mathbb{R}^{pN_i \times rN_d}, \quad \tilde{\Omega} \in \mathbb{R}^{nN_i \times nN_c}, \\
\tilde{\Omega} \in \mathbb{R}^{rN_i \times rN_d}, \quad \Phi \in \mathbb{R}^{nN_i \times qN_c}, \quad \tilde{\Phi} \in \mathbb{R}^{pN_i \times qN_d}, \quad \Psi \in \mathbb{R}^{pN_i \times qN_c}.
\]

Finally, according to [6, Fact 2.10.3] and [6, Fact 7.4.23] we have

\[
\begin{align*}
\mathcal{A}_1 & \in \mathbb{R}^{p^2N_i \times n^2}, & \mathcal{A}_2 & \in \mathbb{R}^{p^2N_i \times r^2}, & \mathcal{A}_4 & \in \mathbb{R}^{p^2N_i \times q^2}, & \mathcal{A}_5 & \in \mathbb{R}^{p^2N_i \times q^2} \\
\text{rank}(\mathcal{A}_1) & = n^2, & \text{rank}(\mathcal{A}_2) & = r^2, & \text{rank}(\mathcal{A}_4) & = q^2, & \text{rank}(\mathcal{A}_5) & = q^2.
\end{align*}
\]

Assumption 3.2 and [6, Fact 2.11.11] will ensure that

\[
\mathcal{A}_3 := (\mathcal{A}_4 + \mathcal{A}_5) \in \mathbb{R}^{p^2N_i \times q^2}, & \text{ rank}(\mathcal{A}_3) = q^2.
\]

Assumption 3.2 and [6, Fact 2.11.9] will ensure the rank of \( \mathcal{A} \) equals to \( n^2 + r^2 + q^2 \), which is a full rank matrix.

Note that, if Assumptions 3.2 and/or 3.3 are not satisfied, one has to manually check the rank of matrix \( \mathcal{A} \) and make sure matrix \( \mathcal{A} \) is full column rank.

When dealing with a small sample of measurements, inappropriate choices of \( M_c \) or significant model error, the ALS estimate of the covariances may not be positive-definite.
and such estimates are physically meaningless [33]. This problem can be solved by adding positive-definite constraints to the linear least squares problem (3.8a) to get

$$\min_{\vartheta} \left\| \begin{bmatrix} \mathcal{A}_0 \\ \vdots \\ \mathcal{A}_{M_t-N} \end{bmatrix} \begin{bmatrix} (P_{1|0})_{ss} \\ (Q)_{ss} \\ (R)_{ss} \end{bmatrix} - \begin{bmatrix} \vec{b}_0 \\ \vdots \\ \vec{b}_{M_t-N} \end{bmatrix} \right\|_2^2$$

s.t. $$(P_{1|0}, Q, R) \succ 0.$$ (3.9)

Since the noise covariance might be a square matrix rather than a scalar, we examine the estimation accuracy by calculating the “error norm”

$$e_Q := \| Q - Q^* \|_F,$$
$$e_R := \| R - R^* \|_F,$$ (3.10a)

and “error percentage”

$$p_Q := \frac{e_Q}{\| Q \|_F} \times 100 \text{ %}, \quad p_R := \frac{e_R}{\| R \|_F} \times 100 \text{ %},$$ (3.10b)

where $Q^*$ and $R^*$ are the estimated state and output noise covariance matrices, respectively.

In the numerical examples of [33, 35, 36], the off-diagonal entries in the estimated noise covariance matrices were ignored due to the true covariances being either scalars or diagonal matrices, which is only acceptable if the state and output noises $w_k$ and $v_k$ are known to be i.i.d. random variables. However, since $Q$ and $R$ may not be diagonal matrices, if we only consider their diagonal entries, the estimation error would be inaccurate. Moreover, in practice, $G_k$ and $H_k$ are rarely known, hence $G_k Q G_k^\top$ and $H_k R H_k^\top$ are usually estimated, thus off-diagonal entries in $\hat{Q}$ and $\hat{R}$ should not be discarded.

A different approach for estimating the noise covariances for time-varying and nonlinear systems is provided in [35], which assumes that there exists a $k_0$ with $1 < k_0 < M$ such that

$$\mathbb{E}\{\varepsilon_{k_0}\} = 0 \text{ and } \lim_{N_k \to \infty} \left( \prod_{k=k_0}^{k_0+N_k} \bar{A}_k \right) \varepsilon_{k_0} = 0.$$ (3.11)

Thus, the advantage of the algorithm in [35] 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 (3.11) will hold as long as the LTV system (2.1) is uniformly detectable;
Noise Covariance Identification for Linear Time-Varying Systems using the Auto-covariance Least Squares Method

however, if the available output measurements \((y_k)_{k=1}^M\) is limited, the statements (3.11) may not hold within the historical data size \(M\). 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

\[
\begin{pmatrix}
  (k_0+N_k)^{(b)} \\
  \prod_{k=k_0} \tilde{A}_k
\end{pmatrix}
\hat{P}_{k_0}
\begin{pmatrix}
  (k_0+N_k)^{(b)} \\
  \prod_{k=k_0} \tilde{A}_k
\end{pmatrix}^T \approx 0
\] (3.12)

mainly depend on the value of \(N_k\), dimension and sparsity of matrices \(\tilde{A}_k\). Our method sets \(k_0 = 1\) by letting \(\hat{x}_{1|0} = \hat{x}_{1|M}\) and treats the term \((P_{1|0})_{ss}\) as a decision variable in the ALS estimation problem (3.9), hence our formulation does not involve any approximations, has fewer 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 \(P_{1|0}\) of a uniformly detectable LTV system.

Given an appropriate choice of \(N_k\) that satisfies (3.12) and \(N_k \leq M_e - N\), if let \(k_0 = 2\), then the existing ALS algorithm provided in [35] is given by

\[
\min_{(Q)_{ss}, (R)_{ss}} \left\| \begin{bmatrix}
\mathcal{A}_{N_k} \\
\vdots \\
\mathcal{A}_{M_e-N}
\end{bmatrix}
\begin{bmatrix}
(Q)_{ss} \\
(R)_{ss}
\end{bmatrix} - \begin{bmatrix}
\tilde{b}_{N_k} \\
\vdots \\
\tilde{b}_{M_e-N}
\end{bmatrix}\right\|_2^2 \quad \text{s.t.} \quad (Q, R) \succ 0,
\] (3.13)

where for \(i = N_k, \ldots, M_e - N\),

\[
\mathcal{A}_i := \left[ (\bar{\Omega}_i \otimes \Omega_i)J_{i+1,r} \mathcal{P}_r \right. \\
\left. \left( (\bar{\Phi}_i \otimes \Phi_i)J_{i+1,q} + I_q \otimes \Psi_i \right) \mathcal{P}_q \right].
\]

3.4 Memory Allocation for the ALS Estimation

Although we have split \((R)_i\) into several smaller portions \((R_i)_s\) in (3.7); however, due to the Kronecker products in (3.7), as well as in (3.8b), calculating each \(\mathcal{A}_i\) in (3.8b) will exhaust large amounts of computer memory as \(i\) goes to \(M_e - N\). Thus, in order to apply our ALS algorithm for higher order applications, such as ocean wave forecasting, one has to carefully modify (3.7) so that the memory can be used efficiently.
3.4 Memory Allocation for the ALS Estimation

3.4.1 Full Matrix or Sparse Matrix?

By default, MATLAB represents a number using floating-point in double-precision. Thus for any full matrix $B \in \mathbb{R}^{r \times c}$, each entry will cost 8 bytes memory space, so the matrix $B$ requires $r \times c \times 8$ bytes of memory. For 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

$$M_r = (1 + c) \times 8 + n_0 \times 8 + n_0 \times 8,$$

where the first index term $IA$ represents 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 3.1.** In MATLAB, 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 $B \in \mathbb{R}^{r \times c}$, 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

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

$$p_n = 50\% - \frac{(1 + c)}{(r \times c) \times 2},$$

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

Due to the structures of matrices $\Gamma_i$, $\Omega_i$, $\Phi_i$ and $\Psi_i$ in (3.8b), 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.
where \( J_{i+1,n}, J_{i+1,r} \) and \( J_{i+1,q} \) have been decomposed as

\[
\begin{align*}
J_{i+1,n} &= \sum_{j=1}^{i+1} \bar{\zeta}_j^T \otimes \bar{\zeta}_j^r, \\
J_{i+1,r} &= \sum_{j=1}^{i+1} \bar{\zeta}_j^r \otimes \bar{\zeta}_j^r, \\
J_{i+1,q} &= \sum_{j=1}^{i+1} \bar{\zeta}_j^r \otimes \bar{\zeta}_j^r,
\end{align*}
\]

Because \( P_{1|0}, Q \) and \( R \) are all symmetric matrices, from a memory efficiency point of view, we should involve \( D_n, D_r \) and \( D_q \) in the decomposition as well. However, we are unable to decompose any of \( J_{i+1,n} D_n, J_{i+1,r} D_r \) or \( J_{i+1,q} D_q \) into a sum of \( i + 1 \) Kronecker product terms, thus involving \( D_n, D_r \) and \( D_q \) in the decomposition using [6, Prop. 7.1.6] will be computational inefficient.

### 3.4.3 Modified Method 2

Instead of using [6, Prop. 7.1.6], a sum Schur/Hadamard products could be used to improve the memory efficiency of our ALS algorithm, such as

\[
\begin{align*}
(\mathcal{R}_i)_s &= \\
& \left\{ \sum_{j=1}^{i+1} \left[ (\bar{\Gamma}_i \bar{\xi}_j^F) \otimes 1_{pN,1} \right] + \left[ (\bar{\Phi}_i \bar{\xi}_j^F) \otimes 1_{pN,1} \right] \right\} (P_{1|0})_{ss} \\
&+ \left\{ \sum_{j=1}^{i+1} \left[ (\bar{\Omega}_i \bar{\xi}_j^O) \otimes 1_{pN,1} \right] + \left[ (\bar{\Phi}_i \bar{\xi}_j^O) \otimes 1_{pN,1} \right] \right\} (Q)_{ss} \\
&+ \left\{ \sum_{j=1}^{i+1} \left[ (\bar{\Phi}_i \bar{\xi}_j^* ) \otimes 1_{pN,1} \right] + \left[ (\bar{\Phi}_i \bar{\xi}_j^* ) \otimes 1_{pN,1} \right] \right\} (R)_{ss},
\end{align*}
\]
3.5 Numerical Examples

where

\[ \xi^\Gamma_j, \bar{\xi}^\Gamma_j, \tilde{\xi}^\Gamma_j, \xi^\Omega_j, \bar{\xi}^\Omega_j, \tilde{\xi}^\Omega_j, \xi^\Phi_j, \bar{\xi}^\Phi_j, \tilde{\xi}^\Phi_j, \xi^H, \bar{\xi}^H, \tilde{\xi}^H, \xi^\Psi \in \mathbb{R}^{(i+1) \times \frac{n(n+1)}{2}}, \]

are matrices that only contain zeros and ones. Note that, if one assumes \( P_{10} \) and \( R \) are diagonal matrices, then all \( \tilde{\xi} \) matrices are equal to zero.

### 3.4.4 Numerical Tests

According to Appendix D, we have for \( i = 0, \ldots, M_e - N \)

\[ \Omega_i \in \mathbb{R}^{pN \times r \times (i+1)}, \quad \Phi_i \in \mathbb{R}^{pN \times p \times (i+1)}, \quad \Gamma_i \in \mathbb{R}^{pN \times n}, \quad \Psi_i \in \mathbb{R}^{pN \times q}, \]

Since only the size of \( \Omega_i, \bar{\Omega}_i, \Phi_i \) and \( \tilde{\Phi}_i \) will grow with \( i \) and, in practice \( p \ll r \), we only focus on monitoring how the size of \( \Omega_i \) and \( \bar{\Omega}_i \) vary with \( i \) and \( r \).

Figure 3.1 and 3.2 are the plots of memory requirements and computational time for calculating the part

\[ \Omega_Q := (\bar{\Omega}_i \otimes \Omega_i)\mathcal{I}_{i+1}^r, \] (3.16)

respectively, using (3.16), the memory efficient forms (3.14) and (3.15) with \( N = 50 \). All results are based on an Intel Xeon E5-2699v3 octadeca-core CPU at 3.0 GHz and 128 GB DDR4 memory. Figure 3.1 illustrates that, by using the memory efficient form (3.14) and (3.15), the maximum memory requirement reduces from 84.94 Gigabytes to just 372.5 and 118.1 Megabytes, respectively. Figure 3.2 shows that for a larger \( Q \), using memory efficient methods 1 and 2 will be faster than using (3.16).

### 3.5 Numerical Examples

We present two LTV examples to investigate the performance of our new ALS method. The first LTV example satisfies (3.11), which is used to compare the estimation accuracy between our ALS algorithm and the existing algorithm provided in [35]; the second LTV example is used to show that our algorithm will work even if (3.11) is not satisfied for the given output measurements; All results in this section are based on an Intel Xeon E5-2699v3 octadeca-core CPU at 3.0 GHz, MATLAB R2015a and MOSEK 7.0 SDP solver. Figure 3.3 shows the time taken versus the number of CPU cores when computing the ALS algorithm with \( M_e = 1000 \), \( N = 150 \) and \( n = 3 \), based on an Intel Xeon E5-2699 CPU.
Fig. 3.1 Memory requirement for calculating $\Omega_Q$

Fig. 3.2 Computational time for calculating $\Omega_Q$
3.5 Numerical Examples

Fig. 3.3 Time taken vs. the number of CPU cores for setting up the ALS algorithm

3.5.1 Linear Time-Varying System 1

Consider tracking an irregular sinusoidal wave with given time-varying frequencies $\left(c_k\right)_{k=1}^{1000}$.

\[
\begin{bmatrix}
  a_{k+1} \\
  b_{k+1}
\end{bmatrix}_{x_{k+1}} =
\begin{bmatrix}
  \cos(c_k T_s) & \sin(c_k T_s) \\
  -\sin(c_k T_s) & \cos(c_k T_s)
\end{bmatrix}_{A_k}
\begin{bmatrix}
  a_k \\
  b_k
\end{bmatrix}_{x_k} +
\begin{bmatrix}
  w^a_k \\
  w^b_k
\end{bmatrix}_{w_k}
\]

\[
y_k := \begin{bmatrix} 1 & 0 \end{bmatrix}_C x_k + v_k
\]

where $T_s = 0.1$ sec is the sampling time and $w_k$ and $v_k$ satisfy Assumption 2.6. We randomly generate time-varying frequencies $\left(c_k\right)_{k=1}^{1000}$, such that $c_k \sim N(0.55, 1 \times 10^{-2})$ and randomly pick an initial state from a uniform distribution, such that

\[
x_1 :=
\begin{bmatrix}
  a_1 \sim U(2,4) \\
  b_1 \sim U(-4,-2)
\end{bmatrix}
\]
Fig. 3.4 Eigenvalues of matrix $\prod_{k=2}^{2+N_k} A_k$ as $N_k$ increases.

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

$$Q = \begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix} \times 10^{-4}, \quad R = 1 \times 10^{-4}. $$

By guessing an initial state error covariance $P_{1|0} = P_g = 20 \times I_2$ and the guessed initial state $\hat{x}_{1|0} \sim \mathcal{N}(0, 5)$, the sub-optimal filter gains $(L_{k})_{k=1}^{1000}$ and the state error covariance $(P_{k})_{k=1}^{1000}$ can be obtained from (2.4). Constructing the sequence of sub-optimal filter gains $(L_{k})_{k=1}^{1000}$ requires appropriate initial guesses of the noise covariance matrices, $Q_g$ and $R_g$. In this example we simply let $Q_g = I_2$ and $R_g = 1$.

Figure 3.4 shows that, in this case, two eigenvalues of square matrix $\prod_{k=2}^{2+N_k} A_k$ quickly converge to zero as $N_k$ increases, thus (3.11) is fulfilled. Therefore, it is possible to compare the estimation accuracy between our ALS algorithm and the existing algorithm in [35].

The maximum number of time lags $N$ can be determined by examining Figure 3.5, which is the plot of the autocorrelation function of the innovation sequence $(\tilde{z}_k)_{k=1}^{1000}$ against the time-lagged variable up to 500. Two blue horizontal lines in Figure 3.5 represent the 95% confidence intervals for the distribution $\mathcal{N}(0, \frac{1}{499})$, which is equal to $\pm 1.96 \sqrt{\frac{1}{499}}$ [27]. Figure 3.5
3.5 Numerical Examples

Fig. 3.5 Autocorrelation function of the innovation sequence shows that for all $N > 10$ the correlation between $\bar{z}_k$ and $\bar{z}_{k+N}$ are within 95% confidence intervals, which can be ignored.

We re-generate output measurements using different initial state and noise sequences and repeat the simulation 200 times with $N = 20$, $M_e = 1000$. Figures 3.6 and 3.7 are scatter plots of 200 estimates of the noise covariances $Q^*$ and $R^*$, as well as the average of all 200 estimates.

Table 3.1 presents the mean error percentage of 200 estimations of $Q$ and $R$ such that

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

where

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

and the variances of $e_Q$ and $e_R$ (defined in (3.10)) using different $M_e$ as well as the average time taken for constructing ($\bar{T}_c$) and solving ($\bar{T}_s$) the optimization problem (3.9). Table 3.1 clearly shows that as $M_e$ goes to $M$, error percentage $\bar{p}_Q$ and the variance of $e_Q$ are getting smaller, but the $\bar{T}_c$ are greatly increased.
Fig. 3.6 Estimation of noise covariances $Q^*$

Fig. 3.7 Estimation of noise covariances $Q^*$ and $R^*$
3.5 Numerical Examples

Table 3.1 Noise covariance estimation results using different $M_e$

<table>
<thead>
<tr>
<th>$M_e$</th>
<th>$\bar{p}_Q$ (%)</th>
<th>Variance of $e_Q$</th>
<th>$\bar{p}_R$ (%)</th>
<th>Variance of $e_R$</th>
<th>$\tilde{T}_c$ (sec)</th>
<th>$\tilde{T}_s$ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>14.47</td>
<td>$2.265 \times 10^{-7}$</td>
<td>1.67</td>
<td>$1.395 \times 10^{-9}$</td>
<td>0.191</td>
<td>0.2681</td>
</tr>
<tr>
<td>400</td>
<td>10.05</td>
<td>$1.320 \times 10^{-7}$</td>
<td>3.61</td>
<td>$1.128 \times 10^{-9}$</td>
<td>1.103</td>
<td>0.2954</td>
</tr>
<tr>
<td>600</td>
<td>5.31</td>
<td>$1.121 \times 10^{-7}$</td>
<td>4.77</td>
<td>$8.588 \times 10^{-10}$</td>
<td>2.11</td>
<td>0.3284</td>
</tr>
<tr>
<td>800</td>
<td>4.18</td>
<td>$9.872 \times 10^{-8}$</td>
<td>5.62</td>
<td>$6.387 \times 10^{-10}$</td>
<td>3.58</td>
<td>0.3024</td>
</tr>
<tr>
<td>1000</td>
<td>3.50</td>
<td>$8.504 \times 10^{-8}$</td>
<td>4.83</td>
<td>$4.180 \times 10^{-10}$</td>
<td>5.64</td>
<td>0.3172</td>
</tr>
</tbody>
</table>

Table 3.2 Performance comparison between two ALS algorithms

<table>
<thead>
<tr>
<th>$N_k$</th>
<th>$\bar{p}_Q$ (%)</th>
<th>Variance of $e_Q$</th>
<th>$\bar{p}_R$ (%)</th>
<th>Variance of $e_R$</th>
<th>$\tilde{T}_c$ (sec)</th>
<th>$\tilde{T}_s$ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Our ALS Algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3.5</td>
<td>$8.504 \times 10^{-8}$</td>
<td>4.83</td>
<td>$4.180 \times 10^{-10}$</td>
<td>5.64</td>
<td>0.3172</td>
</tr>
<tr>
<td></td>
<td>Existing ALS Algorithm [35]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>89.57</td>
<td>$2.000 \times 10^{-7}$</td>
<td>9.70</td>
<td>$3.492 \times 10^{-10}$</td>
<td>5.59</td>
<td>0.2336</td>
</tr>
<tr>
<td>10</td>
<td>12.70</td>
<td>$7.206 \times 10^{-8}$</td>
<td>3.06</td>
<td>$1.682 \times 10^{-10}$</td>
<td>5.49</td>
<td>0.2362</td>
</tr>
<tr>
<td>15</td>
<td>6.65</td>
<td>$6.914 \times 10^{-8}$</td>
<td>6.65</td>
<td>$1.657 \times 10^{-10}$</td>
<td>5.54</td>
<td>0.2472</td>
</tr>
<tr>
<td>20</td>
<td>5.87</td>
<td>$6.992 \times 10^{-8}$</td>
<td>3.23</td>
<td>$1.652 \times 10^{-10}$</td>
<td>5.53</td>
<td>0.2503</td>
</tr>
<tr>
<td>25</td>
<td>5.63</td>
<td>$7.057 \times 10^{-8}$</td>
<td>3.01</td>
<td>$1.600 \times 10^{-10}$</td>
<td>5.50</td>
<td>0.2550</td>
</tr>
<tr>
<td>30</td>
<td>5.58</td>
<td>$7.096 \times 10^{-8}$</td>
<td>2.97</td>
<td>$1.600 \times 10^{-10}$</td>
<td>5.66</td>
<td>0.2679</td>
</tr>
</tbody>
</table>

Table 3.2 compares the mean error percentage $\bar{p}_Q$, $\bar{p}_R$, the variances of $e_Q$ and $e_R$ and the average time taken $\tilde{T}_c$ and $\tilde{T}_s$ between our ALS algorithm with $M_e = 1000$ and $N = 20$ and the existing algorithm provided in [35] with different choices of $N_k$. For the sake of fairness, the values of $k_0$ in [35] are set to 2. Table 3.2 shows that as $N_k$ goes to 30, the mean of the estimated covariances is getting closer to the true covariances and the variance of the estimated covariances is getting smaller. The reason why the results using the ALS algorithm in [35] are still worse than our ALS algorithm, when $N_k = 30$, is because the effective $M_e$ used in the existing ALS algorithm is now 970 rather than 1000.

Given the exact same information, Table 3.3 compares noise covariance estimates and the time taken $T$ for our ALS algorithm, the existing ALS algorithm given in [35] and the EM algorithm given in [48]. Table 3.3 also provides the mean error percentage of state estimates $\bar{p}_x$ such that

$$\bar{p}_x = \frac{1}{M} \sum_{k=1}^{M} \frac{\| x_k - \hat{x}_k \|_2}{\| x_k \|_2}$$

using estimated noise covariances.
Table 3.3 Comparison amongst different noise covariance estimation algorithms

<table>
<thead>
<tr>
<th></th>
<th>Our ALS Algorithm</th>
<th>Existing ALS Algorithm [35] ($N_k = 30$)</th>
<th>EM Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P^*_{1</td>
<td>0}$</td>
<td>$\begin{bmatrix} 0.025 &amp; 0.144 \ 0.144 &amp; 0.825 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0 &amp; 0 \ 0 &amp; 1 \end{bmatrix} \times 20$</td>
</tr>
<tr>
<td>$\hat{x}_{1</td>
<td>0}$</td>
<td>$\begin{bmatrix} -0.191 \ 1.278 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 7.45 \ 5.60 \end{bmatrix}$</td>
</tr>
<tr>
<td>$Q^* (\times 10^{-4})$</td>
<td>$\begin{bmatrix} 3.554 &amp; 0.948 \ 0.948 &amp; 4.932 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 4.270 &amp; 3.212 \ 3.212 &amp; 4.138 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 3.328 &amp; 1.832 \ 1.832 &amp; 4.895 \end{bmatrix}$</td>
</tr>
<tr>
<td>$R^* (\times 10^{-4})$</td>
<td>$0.717$</td>
<td>$0.623$</td>
<td>$0.904$</td>
</tr>
<tr>
<td>$\tilde{p}_x (%)$</td>
<td>$7.37$</td>
<td>$7.34$</td>
<td>$7.83$</td>
</tr>
<tr>
<td>$T$ (sec)</td>
<td>$7.61$</td>
<td>$22.15$</td>
<td>$198.69$</td>
</tr>
</tbody>
</table>

We set $\delta = 1 \times 10^{-8}$ and the EM algorithm took 282 iterations and almost 200 seconds to find all estimates; Table 3.3 shows that our ALS algorithm provides more accurate noise covariance and state estimations than the EM algorithm. Both our and existing ALS algorithms provide similar noise covariance estimates; however, since the existing ALS algorithm cannot provide any estimates of initial state and error covariance, our ALS algorithm has a smaller $\tilde{p}_x (\%)$.

### 3.5.2 Linear Time-Varying System 2

Consider a linear time-varying system, for which the evolution of each state follows the same random walk and the output matrix $(C_k)_{k=1}^{1000}$ varies with time:

$$
x_{k+1} := x_k + w_k,

y_k := C_k x_k + v_k,
$$

where $x_k \in \mathbb{R}^{3 \times 1}$, $w_k$ and $v_k$ satisfy Assumption 2.6 and the time-varying output matrix $C_k$ consists of a historical ocean waveform\(^1\) $\eta_k^{1003}$, for which $C_k := \begin{bmatrix} \eta_{k+3} & \eta_{k+2} & \eta_k \end{bmatrix}$.

We randomly pick an initial state $x_1 \sim \mathcal{U}(-2, 2)$, then generate output measurements $(y_k)_{k=1}^{1000}$ based on noise covariances $Q = I_3 \times 10^{-3}$ and $R = 1 \times 10^{-3}$. Figure 3.8 shows the

\(^1\)Recorded at Galway at 5:20 on the 10th of February 2005 with sampling frequency at 2.56Hz
output measurements that is generated using model (3.17). By guessing an initial state error covariance $P_{1|0} = 20 \times I_3$ and the guessed initial state $\hat{x}_{1|0} \sim N(0, I_3)$, the sub-optimal filter gains $(L_k)_{k=1}^{1000}$ and the state error covariance $(P_k)_{k=1}^{1000}$ can be obtained from (2.4) with the guessed noise covariance matrices $Q_g = I_3$ and $R_g = 1$.

By constructing the observability Gramian matrix $[2] M_{1,1000}$ of the LTV system (3.17)

$$M_{t_0,t_0+\delta} := \sum_{k=t_0}^{t_0+\delta-1} \left( A_{t_0+\delta,k+1} C_k C_k^\top A_{t_0+\delta,k+1}^\top \right),$$

where

$$A_{t_0+\delta,t_0} := \prod_{k=t_0}^{t_0+\delta-1(b)} A_k = I_3,$$

and checking the eigenvalues of matrix $M_{1,1000}$, it can be shown that all three eigenvalues are greater than zero, which indicates that the observability Gramian matrix $M_{1,1000}$ is positive-definite. According to [47], the LTV system (3.17) is therefore uniformly observable. Because observability is a sufficient condition for detectability [2], the LTV system (3.17) is also uniformly detectable.
Noise Covariance Identification for Linear Time-Varying Systems using the Auto-covariance Least Squares Method

Fig. 3.9 Eigenvalues of matrix $\prod_{k=2}^{2+N_k} \bar{A}_k$ as $N_k$ increases

Figure 3.9 shows that, in this case, the eigenvalues of $\prod_{k=2}^{2+N_k} \bar{A}_k$ finally converge to almost zero only if $N_k$ is very large, therefore, the existing algorithm in [35] will not provide accurate estimates.

We re-generate output measurements using different initial state and noise sequences and repeat the simulation 200 times with $N = 20$ and $M_e = 1000$. Figures 3.10 to 3.12 are scatter plots of 200 estimates of the noise covariances $Q^*$ and $R^*$, as well as the average of all 200 estimates.

Table 3.4 presents the mean error percentage $\bar{p}_Q$, $\bar{p}_R$ and the variances of $e_Q$ and $e_R$ using different $M_e$ as well as $\bar{T}_c$ and $\bar{T}_s$ of the optimization problem (3.9). Table 3.4 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 $\bar{T}_c$ dramatically increases.

Table 3.5 compares the mean error percentage $\bar{p}_Q$, $\bar{p}_R$, the variances of $e_Q$ and $e_R$ using different $M_e$ and the average time taken for constructing and solving the optimization problem (3.9) between our ALS algorithm with $M_e = 1000$ and $N = 20$ and the existing algorithm provided in [35] with different choices of $N_k$ and the value of $k_0$ in [35] is set to 2. Because $M_e = M - N_k$, the estimation accuracy is a trade off between the $N_k$ and $M_e$ and Table 3.5 shows the best estimates appear at $N_k = 300$. 
3.5 Numerical Examples

Fig. 3.10 Estimation of noise covariances $Q^*$

Fig. 3.11 Estimation of noise covariances $Q^*$ and $R^*$
Noise Covariance Identification for Linear Time-Varying Systems using the Auto-covariance Least Squares Method

![Graph showing the estimation of noise covariances $Q^*$](image)

Fig. 3.12 Estimation of noise covariances $Q^*$

Table 3.4 Noise covariance estimation results using different $M_e$

<table>
<thead>
<tr>
<th>$M_e$</th>
<th>$\tilde{p}_Q$ (%)</th>
<th>Variance of $e_Q$</th>
<th>$\tilde{p}_R$ (%)</th>
<th>Variance of $e_R$</th>
<th>$T_e$ (sec)</th>
<th>$T_s$ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>165.1</td>
<td>$2.833 \times 10^{-5}$</td>
<td>7.40</td>
<td>$1.842 \times 10^{-9}$</td>
<td>0.22</td>
<td>0.3619</td>
</tr>
<tr>
<td>400</td>
<td>89.76</td>
<td>$5.933 \times 10^{-6}$</td>
<td>3.17</td>
<td>$1.256 \times 10^{-9}$</td>
<td>0.78</td>
<td>0.4060</td>
</tr>
<tr>
<td>600</td>
<td>65.09</td>
<td>$3.424 \times 10^{-6}$</td>
<td>4.86</td>
<td>$9.273 \times 10^{-10}$</td>
<td>1.72</td>
<td>0.4154</td>
</tr>
<tr>
<td>800</td>
<td>51.59</td>
<td>$1.880 \times 10^{-6}$</td>
<td>6.32</td>
<td>$8.865 \times 10^{-10}$</td>
<td>3.38</td>
<td>0.4075</td>
</tr>
<tr>
<td>1000</td>
<td>36.49</td>
<td>$1.172 \times 10^{-6}$</td>
<td>7.06</td>
<td>$7.586 \times 10^{-10}$</td>
<td>6.07</td>
<td>0.3752</td>
</tr>
</tbody>
</table>
3.6 Conclusions

We have developed a noise covariance estimation algorithm for time-varying systems based on a constrained (positive-definite) auto-covariance least-squares method. We used two

<table>
<thead>
<tr>
<th>$N_k$</th>
<th>$\bar{p}_Q$ (%)</th>
<th>Variance of $e_Q$</th>
<th>$\bar{p}_R$ (%)</th>
<th>Variance of $e_R$</th>
<th>$T_c$ (sec)</th>
<th>$T_s$ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>36.49</td>
<td>$1.172 \times 10^{-6}$</td>
<td>7.06</td>
<td>$7.586 \times 10^{-10}$</td>
<td>6.07</td>
<td>0.3752</td>
</tr>
<tr>
<td>50</td>
<td>82.36</td>
<td>$2.848 \times 10^{-6}$</td>
<td>10.21</td>
<td>$8.071 \times 10^{-10}$</td>
<td>6.12</td>
<td>0.3643</td>
</tr>
<tr>
<td>100</td>
<td>63.80</td>
<td>$1.810 \times 10^{-6}$</td>
<td>9.73</td>
<td>$8.940 \times 10^{-10}$</td>
<td>5.68</td>
<td>0.2762</td>
</tr>
<tr>
<td>300</td>
<td>58.68</td>
<td>$1.643 \times 10^{-6}$</td>
<td>12.89</td>
<td>$1.303 \times 10^{-9}$</td>
<td>5.48</td>
<td>0.1634</td>
</tr>
<tr>
<td>500</td>
<td>65.13</td>
<td>$1.983 \times 10^{-6}$</td>
<td>22.84</td>
<td>$1.759 \times 10^{-9}$</td>
<td>4.64</td>
<td>0.1654</td>
</tr>
<tr>
<td>700</td>
<td>80.54</td>
<td>$2.721 \times 10^{-6}$</td>
<td>12.9</td>
<td>$1.386 \times 10^{-9}$</td>
<td>4.03</td>
<td>0.2810</td>
</tr>
<tr>
<td>900</td>
<td>147.56</td>
<td>$1.309 \times 10^{-5}$</td>
<td>21.4</td>
<td>$3.066 \times 10^{-10}$</td>
<td>2.68</td>
<td>0.4072</td>
</tr>
</tbody>
</table>

Given the exact same information, similar to Table 3.3 in Section 3.5.1, Table 3.6 compares noise covariance estimates, $\bar{p}_s$, and time taken $T$ amongst our ALS algorithm, existing ALS algorithm and EM algorithm. For the EM algorithm, we set $\delta = 1 \times 10^{-9}$ and the EM algorithm took 1500 iterations and almost 1400 seconds to find all estimates. Unlike Table 3.3 in Section 3.5.1, Table 3.6 shows that the EM algorithm provides the most accurate noise covariance and state estimates. It is interesting to point out, although existing ALS algorithm provides more accurate covariance estimates than the initial guesses, using the existing ALS algorithm will cause much higher state estimation errors than using the initial guesses. Figure 3.13 is the estimation plot of the 1st state, because the existing ALS algorithm cannot provide the estimate of initial state error covariance, the Kalman filter failed to track system states until after 15 seconds, where

$$P_{1|0}^{*}(ALS) := \begin{bmatrix} 0.211 & -0.326 & 0.145 \\ -0.326 & 0.617 & -0.286 \\ 0.146 & -0.286 & 0.157 \end{bmatrix} \times 10^{-4}$$

and

$$P_{1|0}^{*}(EM) := \begin{bmatrix} 0.143 & -0.165 & 0.022 \\ -0.165 & 0.318 & -0.143 \\ 0.022 & -0.143 & 0.121 \end{bmatrix} \times 10^{-4}$$
Noise Covariance Identification for Linear Time-Varying Systems using the Auto-covariance
Least Squares Method

Fig. 3.13 State estimation using the existing ALS algorithm

Table 3.6 Comparison of different noise covariance estimation algorithms

<table>
<thead>
<tr>
<th></th>
<th>Our ALS Algorithm</th>
<th>Existing ALS Algorithm [35] (Nk = 300)</th>
<th>EM Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>P*_{1</td>
<td>0}</td>
<td>P*_{1</td>
<td>0}(ALS)</td>
</tr>
<tr>
<td>̂x*_{1</td>
<td>0}</td>
<td>–0.866</td>
<td>–0.887</td>
</tr>
<tr>
<td></td>
<td>0.236</td>
<td>0.214</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td>1.020</td>
<td>0.071</td>
<td>1.566</td>
</tr>
<tr>
<td>Q* (×10⁻³)</td>
<td>1 0.1 –0.2</td>
<td>0.4 0.4 –0.3</td>
<td>0.8 –0.1 –0.2</td>
</tr>
<tr>
<td></td>
<td>0.1 0.5 0.5</td>
<td>0.4 1.4 0.1</td>
<td>–0.1 1.6 0.1</td>
</tr>
<tr>
<td></td>
<td>–0.2 0.5 0.7</td>
<td>–0.3 0.1 0.7</td>
<td>–0.2 0.1 0.8</td>
</tr>
<tr>
<td>R* (×10⁻⁴)</td>
<td>0.843</td>
<td>0.703</td>
<td>0.930</td>
</tr>
<tr>
<td>̃pₓ (%)</td>
<td>8.67</td>
<td>82.93</td>
<td>7.84</td>
</tr>
<tr>
<td>T (sec)</td>
<td>7.91</td>
<td>7.69</td>
<td>1388.9</td>
</tr>
</tbody>
</table>
LTV examples to investigate the performance of the algorithm, both numerical examples returned accurate estimates and the estimation accuracy relies on the choice of $M_e$. Since our algorithm does not involve any approximations and has fewer parameters involved, our algorithm provided better covariance estimates, compared to [35]. Our method added one more decision variable in the optimization problem, but in return is able to estimate the initial state error covariance, which will further reduce the error of in the state estimates. The overall computational time of constructing and solving the optimization problem can be significantly reduced by using parallel implementations and efficient SDP solvers.
Chapter 4

Noise Covariance Identification for Nonlinear Systems using Auto-covariance Least Squares and Expectation Maximization Method

For estimating noise covariances in nonlinear systems, the time-varying ALS and EM algorithms was extended in [5, 36] by linearizing the nonlinear function around current state estimates via a first order Taylor series, where the state estimates are obtained using the EKF. The main problem of linearization is model mismatch, where the difference between the real and linearized model depends on the order of Taylor series and the accuracy of the state estimate. Rather than the EKF, linearization was avoided by using the particle, unscented Kalman filter/sm oother in [4, 5, 9], combined with the EM method.

Instead of changing the order of linearization, we replace the EKF with the MHE in our ALS and EM based estimation algorithm, so the accuracy and stability of the state and

We also ascend dazzling and tremendous as the sun, We found our own O my soul in the calm and cool of the daybreak. My voice goes after what my eyes cannot reach, With the twirl of my tongue I encompass worlds and volumes of worlds.

Song of Myself
Walter Whitman
covariance estimation can be improved. Furthermore, the MHE allows one to add model constraints, so the convergence of state estimation can be further enhanced. For the EM method, we propose to use a semi-definite programming (SDP) solver to estimate the noise covariance, rather than the analytical solution given by [5], so that the covariance estimates are more accurate and guaranteed to be positive definite.

Both the ALS and EM methods are able to provide accurate noise covariance estimates for LTV and some nonlinear systems. Compared to the EM method, the advantages of the ALS method are that (i) it could handle a nonlinear system with noises are not additive to the nonlinear dynamics and (ii) it estimates noise covariances by solving one optimization problem, which can be potentially faster than the iterative-based EM method. However, for some nonlinear systems with additive noise terms, the ALS method may fail to provide decent noise covariance estimates if the correlation of the innovation sequence cannot be approximately written as a linear function of noise covariances. Furthermore, the ALS method has more parameters to tune and it is more complicated to construct the estimation problem, compared to the EM method.

4.1 Auto-covariance Least Squares Method with the Moving Horizon Estimation

Recalling the discrete-time nonlinear model (2.14)

\[
\begin{align*}
  x_{k+1} &:= f(x_k, w_k) \\
  y_k &:= h(x_k, v_k)
\end{align*}
\]

where Assumptions 2.4 and 2.6 hold. If we linearize the nonlinear functions \( f(\cdot) \) and \( h(\cdot) \) around the current estimate \( \hat{x}_k \) using (2.15), then \( A_k, G_k, C_k \) and \( H_k \) are linearized matrices, hence we have

\[
\begin{align*}
  \epsilon_{k+1} &\approx \tilde{A}_k \epsilon_k + \tilde{G}_k \tilde{w}_k \\
  z_k &\approx C_k \epsilon_k + H_k v_k.
\end{align*}
\] (4.1)

**Assumption 4.1.** For the nonlinear system (2.14), linearized matrices \( G_k \) and \( H_k \) obtained from (2.15) that are full column rank matrices, for all \( k \).

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 \( L_k^x \) and innovation \( \tilde{z}_k \) can all be determined using (2.15), (2.4) and (3.3), respectively. If Assumptions 3.1 and 4.1 are fulfilled, then the noise covariances for nonlinear systems can be estimated by establishing and solving the ALS problem (3.9) using all the
4.2 Expectation Maximization Method with the Full Information Estimation

information derived in Section 3.2. In the following sections, we discuss the properties of using the EKF and FIE (MHE) to estimate the true system state sequence \((x_k)_{k=1}^M\).

4.1.1 Estimating States Using an Extended Kalman Filter

Assumption 3.1 requires the estimation errors \((\varepsilon_k)_{k=1}^M\) to be bounded and the expectation of \(\varepsilon_k\) equal to zero for all \(k\). Unfortunately, for nonlinear state estimation using the EKF, Assumption 3.1 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, due to divergence of state estimates. Thus, additional conditions are needed to improve the stability and convergence of the EKF [44], including observability, small initial estimation error, small noise terms and no model mismatch.

4.1.2 Estimating States Using Moving Horizon Estimation

In order to improve state estimation for nonlinear system (2.16), where both state and output noises \(w_k\) and \(v_k\) are additive to the nonlinear dynamics, one may have to use the FIE, rather than EKF, to estimate the unknown system states. Because FIE propagates the state using the original nonlinear model; physical constraints can be involved in the FIE and Proposition 2.6 shows that the EKF equations can be derived by minimizing the same objective function as FIE by a single optimization step. Hence, by optimality, FIE will provide smaller error norms \(\|e_k\|_{k=1}^M\) than the EKF.

The initial state \(\hat{x}_{1|0}\) can be determined by firstly selecting a \(\kappa \ll M\), then recursively solving the FIE (2.24), for \(k = 1, \ldots, \kappa\) and setting \(\hat{x}_{1|0} = \hat{x}_{1|\kappa}\). After \(\hat{x}_{1|0}\) is obtained using the FIE, the full state trajectory \((x_k)_{k=1}^M\) can be estimated by solving the MHE problem (2.26) using the flow chart in Figure 2.1, hence if both \(\|w_k\|\) and \(\|v_k\|\) are bounded, the stability of nonlinear state estimation can be guaranteed by Theorem 2.3. Algorithm 4.1 gives the procedures of estimating noise covariance using ALS combined with MHE.

4.2 Expectation Maximization Method with the Full Information Estimation

Proposition 4.1. For the nonlinear state space model (2.16), where \(G_k\) and \(H_k\) are full column rank matrices, if there exists a stable state observer with nonempty feasible region,
Algorithm 4.1 Noise Covariance Estimation using the ALS combined with MHE 
\((\hat{P}_1, \hat{Q}, \hat{R}) = \text{ALS+MHE} \left(f(\cdot), h(\cdot), \kappa, \hat{x}_{1|0}, P_g, Q_g, R_g, (y_k)^{M}_{k=1}, H_l \right)\)

1: Given \(\kappa, \kappa \leq M, (y_k)^{k}_{k}, \hat{x}_{1|0}, P_g, Q_g, R_g, \) determine the smoothed initial state \(\hat{x}_{1|\kappa}\) by recursively calculating (2.24) for \(k = 1, \cdots, \kappa\).
2: Set \(\hat{x}_{1|0} = \hat{x}_{1|\kappa}\), then recursively determining the filtered states \((\hat{x}_k)^{M}\) using the MHE (2.26) and the algorithm provided in Figure 2.1 with horizon \(H_l\).
3: Calculate \((A_k, G_k, H_k, C_k)^{M}_{k=1}\) using (2.15) and \((\delta_k)^{M}\).
4: Calculate innovation sequence \((z_k)^{M}_{k=1}\) using \((\delta_k)^{M}\) obtained from MHE.
5: Estimate \(P_1, Q\) and \(R\) using the ALS algorithm (3.9).

then the \(\Omega\) defined in (2.47a) can be approximated by a function of \(\Theta, A_k^{M}, G_k, H_k, C_k^{M}, \hat{x}_k^{M}, P_k^{M}\) and \(P_{k-1}^{M}\), where

\[
A_k^{M} := \frac{\partial f(\cdot)}{\partial x_k} \bigg|_{x_k = \hat{x}_k^{M}, w_k = \hat{w}_k^{M}}^{}, \quad C_k^{M} := \frac{\partial h(\cdot)}{\partial x_k} \bigg|_{x_k = \hat{x}_k^{M}, y_k = \hat{y}_k^{M}}^{} \tag{4.2}
\]

are assumed to be full rank matrices.

Proof. The expression for \(\Omega\) for an LTV model was provided in [48]. We extend this method to the nonlinear model (2.16) via linearization.

Taking the conditional expectation over (2.49) gives

\[
\Omega(\Theta|\Theta_{i-1}) = \mathbb{E}\{\log(p(x_{1:M}, y_{1:M} | \Theta)) | y_{1:M}, \Theta_{i-1}\} = \\
-\frac{1}{2} \left[s\log(2\pi) + \log |P_{i-1}| + M\log |Q| + M\log |R| \right] - \frac{1}{2} \text{tr}\left(P_{i-1}^{-1} \mathbb{E}\left[\epsilon_1 \epsilon_1^\top | y_{1:M}, \Theta_{i-1}\right] \right) \\
+ \sum_{k=1}^{M-1} \mathbb{E}\left[w_k w_k^\top | y_{1:M}, \Theta_{i-1}\right] - R^{-1} \sum_{k=1}^{M} \mathbb{E}\left[v_k v_k^\top | y_{1:M}, \Theta_{i-1}\right], \tag{4.3}
\]

where

\[
\mathbb{E}\left[w_k w_k^\top | y_{1:M}, \Theta_{i-1}\right] = \mathbb{C}\left[w_k | y_{1:M}, \Theta_{i-1}\right] + \mathbb{E}\left[w_k | y_{1:M}, \Theta_{i-1}\right] \mathbb{E}\left[w_k | y_{1:M}, \Theta_{i-1}\right]^\top, \\
= \mathbb{E}\left[(w_k - \hat{w}_k^{M}) (w_k - \hat{w}_k^{M})^\top \right] + \hat{w}_k^{M} \hat{w}_k^{M\top}, \tag{4.4a}
\]

\[
\mathbb{E}\left[v_k v_k^\top | y_{1:M}, \Theta_{i-1}\right] = \mathbb{C}\left[v_k | y_{1:M}, \Theta_{i-1}\right] + \mathbb{E}\left[v_k | y_{1:M}, \Theta_{i-1}\right] \mathbb{E}\left[v_k | y_{1:M}, \Theta_{i-1}\right]^\top, \\
= \mathbb{E}\left[(v_k - \hat{v}_k^{M}) (v_k - \hat{v}_k^{M})^\top \right] + \hat{v}_k^{M} \hat{v}_k^{M\top}. \tag{4.4b}
\]
4.2 Expectation Maximization Method with the Full Information Estimation

Hence, \((w_k - \hat{w}_{k|M})\) and \((v_k - \hat{v}_{k|M})\) can be approximated by computing the first order Taylor series on (2.16), given \(Y_M\) and \(O_{i-1}\) [49, p. 408], such that

\[
x_{k+1} \approx f(\hat{x}_{k|M}, \hat{w}_{k|M}) + A_{k|M} (x_k - \hat{x}_{k|M}) + G_k (w_k - \hat{w}_{k|M}),
\]

\[\begin{align*}
(w_k - \hat{w}_{k|M}) & \approx G_k \left[ (x_{k+1} - \hat{x}_{k+1|M}) - A_{k|M} (x_k - \hat{x}_{k|M}) \right], \\
y_k & \approx h(\hat{x}_{k|M}, \hat{v}_{k|M}) + C_{k|M} (x_k - \hat{x}_{k|M}) + H_k (v_k - \hat{v}_{k|M}), \\
(v_k - \hat{v}_{k|M}) & \approx -H_k^T C_{k|M} (x_k - \hat{x}_{k|M}).
\end{align*}\]

Hence, we have

\[
\mathbb{C}[v_k|Y_M, O_{i-1}] = \mathbb{C}^R_k = H_k^T C_{k|M} P_{k|M} C_{k|M}^T H_k^T, \quad (4.5a)
\]

\[
\mathbb{C}[w_k|Y_M, O_{i-1}] = \mathbb{C}^O_k = G_k \left[ (P_{k+1|M} - A_{k|M} P_{k+1,1,M} - P_{k+1,1,M} A_{k|M} + A_{k|M} P_{k|M} A_{k|M}) G_k^T \right],
\]

\[= G_k \left[ (I_n - A_{k|M} U_k) P_{k+1|M} (I_n - A_{k|M} U_k)^T + A_{k|M} (P_k - U_k P_{k+1|k} U_k^T) A_{k|M}^T \right] G_k^T. \quad (4.5b)
\]

Substituting (4.4) and (4.5) into (4.3) gives

\[
\mathbb{Q}(O|O_{i-1}) = \mathbb{E}\{\log(p(X_M, Y_M|O))|Y_M, O_{i-1}\} \approx -\frac{1}{2} \text{tr} \left( Q^{-1} \mathbb{C}_Q \right) - \frac{1}{2} \text{tr} \left( R^{-1} \mathbb{C}_R \right)
\]

\[ \quad - \frac{1}{2} \text{tr} \left\{ P^{-1}_{1|0} \left[ P_{1|M} + (\hat{x}_1 - \hat{x}_{1|M}) (\hat{x}_1 - \hat{x}_{1|M})^T \right] \right\} \quad \text{(4.6)}\]

where \(\mathbb{C}_Q := \sum_{k=1}^{M-1} \mathbb{C}^R_k \hat{w}_{k|M} \hat{w}_{k|M}^T + \mathbb{C}^O_k\) and \(\mathbb{C}_R := \sum_{k=1}^{M-1} \mathbb{C}^R_k \hat{v}_{k|M} \hat{v}_{k|M}^T + \mathbb{C}^R_k\). \(\Box\)

Because \(\hat{x}_{1;i}\) and \(P_{1;i}\) cannot be identified simultaneously, we let \(\hat{x}_{1;i} = \hat{x}_{1|M}\) and \(P_{1;i} = P_{1|M}\) [48]. Thus, if the sequences \((\hat{x}_{k|M})_{k=1}^M, (P_{k|M})_{k=1}^M\) and \((P_{k,k-1|M})_{k=2}^M\) are all computed, then the parameter set \(O_i\) can be estimated by solving (4.6). The optimization problem (4.6) can be solved either by using a SDP solver, such as SDPT-3 [51], or by taking the partial derivative of (4.6) with respect to each entries of \(Q^{-1}\) and \(R^{-1}\), such that

\[
\frac{\partial}{\partial (Q^{-1})_{rc}} Q(O|O_{i-1}) = - (M - 1) \frac{\partial}{\partial (Q^{-1})_{rc}} \log |Q^{-1}| + \frac{\partial}{\partial (Q^{-1})_{rc}} \text{tr} (Q^{-1} \mathbb{C}_Q),
\]

\[
= - \frac{M - 1}{|Q^{-1}|} \frac{\partial |Q^{-1}|}{\partial (Q^{-1})_{rc}} + (\mathbb{C}_Q)_{c,r} = -(M - 1) Q_{c,r} + (\mathbb{C}_Q)_{c,r}, \quad (4.7a)
\]

\[
\frac{\partial}{\partial (R^{-1})_{rc}} Q(O|O_{i-1}) = - \frac{M}{|R^{-1}|} \frac{\partial |R^{-1}|}{\partial (R^{-1})_{rc}} + (\mathbb{C}_R)_{c,r} = -M R_{c,r} + (\mathbb{C}_R)_{c,r}. \quad (4.7b)
\]
By setting \((4.7a)\) and \((4.7b)\) to zero, \(Q_i\) and \(R_i\) can be determined by \([5]\)

\[
Q_i = \frac{1}{M-1} c_Q \quad \text{and} \quad R_i = \frac{1}{M} c_R, \tag{4.8}
\]

where matrices \(Q_i\) and \(R_i\) should be both positive definite, because Proposition 2.3, \((2.5)\) and \((4.5)\) ensure that if \(P_{|0}>0\), then \(c_Q\) and \(c_R\) are positive definite matrices.

From the previous section, it follows that \((2.49)\) is actually the objective function of an FIE problem \([42]\), when the output sequence \(y_1:M\) and parameter set \(O_i−1\) are given. Thus, it is natural to estimate the noise and state sequences \((\hat{w}_k)_{k=1}^{M} , (\hat{v}_k)_{k=1}^{M}\) and \((\hat{x}_k)_{k=1}^{M}\), respectively, by recursively solving the MHE problem \((2.26)\), and then estimating the smoothed state \((\hat{x}_k|M)_{k=1}^{M}\) using the FIE \((2.24)\), for \(\bar{k} = 2, \cdots, M\).

In Proposition 2.6, it was proved that the EKF is equivalent to solving the FIE problem \((2.23)\) by only one optimization step with a carefully chosen initial guess; hence, by optimality the state estimate using the FIE will have a smaller error \(e_{\bar{k}}\) than the EKF. Next, we are going to prove that Proposition 2.6 still holds for the smoothed state estimates \((\hat{x}_k|M)_{k=1}^{M}\).

**Proposition 4.2.** For the nonlinear system \((2.16)\), given output measurements \((y_k)_{k=1}^{M}\), all estimated states \((\hat{x}_k)_{k=1}^{M}\) and noise covariances, if Assumption 2.5 holds, then the state estimate using the FIE \((2.23)\) will have smaller error \(e_{k|M} := \|x_k - \hat{x}_k|\) compared to the extended Kalman smoother (EKS), such that \([5]\)

\[
\hat{x}_k|M = \hat{x}_k + P_k A_k^T P_{k+1|M}^{-1} (\hat{x}_{k+1|M} - f(\hat{x}_k)) , \tag{4.9}
\]

where \(A_k\) is defined by \((2.15)\).

**Proof.** The MHE for the nonlinear system \((2.16)\) with \(H_l = 1\) is given by

\[
X_{M-1,M}^* = \arg \min_{X_{M-1,M}} \frac{1}{2} \|x_{M-1} - \hat{x}_{M-1}\|_{P_{M-1}}^2 + \frac{1}{2} \|w_{M-1}\|_{Q^{-1}}^2 + \frac{1}{2} \|v_M\|_{R^{-1}}^2 , \tag{4.10}
\]

s.t. \(x_M = f(x_{M-1}) + G_{M-1} w_{M-1}\)

\(y_M = h(x_M) + H M v_M\).

By involving Lagrange multipliers \(\alpha_{M-1}\) and \(\beta_M\) for equality constraints, the MHE problem \((4.10)\) can be written as the following unconstrained optimization problem:

\[
Z_{M-1,M}^* := \arg \min_{Z_{M-1,M}} \frac{1}{2} \|x_{M-1} - \hat{x}_{M-1}\|_{P_{M-1}}^2 + \left[ \frac{1}{2} \|v_M\|_{R^{-1}}^2 + \frac{1}{2} \|w_{M-1}\|_{Q^{-1}}^2 + \alpha_{M-1}^T (x_M - f(x_{M-1}) - G_{M-1} w_{M-1}) \right] + \left[ \frac{1}{2} \|v_M\|_{R^{-1}}^2 + \beta_M^T (y_M - h(x_M) - H M v_M) \right].
\]
Since all estimated states \( \hat{x}_k^{M_{k=1}} \) is given, \( v_M \) is deterministic and \( \beta_M \) becomes a free variable, hence we have

\[
Z_{M-1|M}^* := \arg \min_{Z_{M-1|M}} \hat{s}_{M-1|M} = \arg \min_{Z_{M-1|M}} \frac{1}{2} \| x_{M-1} - \hat{s}_{M-1} \|_P^{-1}^2 + \left[ \frac{1}{2} \| w_{M-1} \|_Q^{-1} + \alpha_M^{-1} ( \hat{s}_M - f(x_{M-1}) - G_{M-1}w_{M-1} ) \right].
\] (4.11)

Because (4.11) is an unconstrained nonlinear optimization problem, in order to ensure the convergence of this nonlinear optimization, one could use a backtracking line search method combined with Newton’s method, so the optimum\(^1\) of (4.11) can be found by recursively calculating

\[
Z_{M-1|M}^{(i+1)} = Z_{M-1|M}^{(i)} - a_i \mathcal{H}_{SM-1|M}^{-1}(Z_{M-1|M}^{(i)}) \mathcal{J}_{SM-1|M}(Z_{M-1|M}^{(i)}),
\]

given an initial guess \( Z_{M-1|M}^{(1)} \), until \( Z_{M-1|M}^{(i+1)} \) has converged to its optimal value \( Z_{M-1|M}^* \), where the scalar \( a_i \) is called step length, which is determined using a backtracking line search method, with initial value \( a_1 = 1 \) [32, p. 37]. The expressions for \( \mathcal{J}_{SM-1|M}(Z_{M-1|M}) \) and \( \mathcal{H}_{SM-1|M}(Z_{M-1|M}) \) are given by

\[
\mathcal{J}_{SM-1|M}(Z_{M-1|M}) := \begin{bmatrix}
\frac{\partial s_{M-1|M}}{\partial x_{M-1}} \\
\frac{\partial s_{M-1|M}}{\partial w_{M-1}} \\
\frac{\partial s_{M-1|M}}{\partial \alpha_{M-1}}
\end{bmatrix} = \begin{bmatrix}
P_{M-1}^{-1}x_{M-1} - P_{M-1}^{-1}\hat{s}_{M-1} - \frac{\partial f}{\partial x_{M-1}}^\top \alpha_{M-1} \\
Q^{-1}w_{M-1} - G_{M-1}^\top \alpha_{M-1} \\
\hat{s}_M - f(x_{M-1}) - G_{M-1}w_{M-1}
\end{bmatrix},
\]

\[
\mathcal{H}_{SM-1|M}(Z_{M-1|M}) \approx \begin{bmatrix}
P_{M-1}^{-1} & 0 & -\frac{\partial f}{\partial x_{M-1}}^\top \\
0 & Q^{-1} & -G_{M-1}^\top \\
-\frac{\partial f}{\partial x_{M-1}} & -G_{M-1} & 0
\end{bmatrix},
\]

where all the second derivative of functions \( f(x_{M-1}) \) is ignored.

Since the system dynamics are nonlinear, unlike Proposition 2.5, the global optimum \( Z_{k}^* \) is not guaranteed to be found and optimization usually takes several iterations to find a local optimum. Let us pick an initial guess of the optimum, such that

\[
Z_{M-1|M}^{(1)} := \begin{bmatrix}
\hat{x}_{M-1}^{(1)}^\top \\
0
\end{bmatrix}^\top,
\]

\(^1\)In order to prevent converging to a stationary point rather than a minimizer, negative curvature information from the Hessian \( \mathcal{H}_{SM-1|M} \) may be required [32, p. 40]
where $\hat{x}_{M-1|M}^{(1)} = \hat{x}_{M-1}$. Hence,

$$Z_{M-1|M}^{(2)} = \begin{bmatrix} \hat{x}_{M-1}^\top \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} P_{M-1}^{-1} & 0 & -A_{M-1}^\top \\ 0 & Q^{-1} & -G_{M-1}^\top \\ -A_{M-1} & -G_{M-1} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \hat{x}_M - f(\hat{x}_{M-1}) \end{bmatrix},$$

where

$$\hat{x}_{M-1|M}^{(2)} = \hat{x}_{M-1|M}^{(1)} + P_{M-1}A_{M-1}^\top P_{M|M-1}^{-1} \left( \hat{x}_M - f(\hat{x}_{M-1|M}) \right).$$

It is clear to see that for a smoothed estimate $\hat{x}_{M-1|M}$, the EKS is equivalent to solving (4.10) by only one optimization step with a carefully chosen initial guess $Z_{k|M}^{(1)}$.

If increase the horizon length $H_t$ to 2, the MHE for the nonlinear system (2.16) becomes

$$Z_{M-2,M}^* := \arg\min_{Z_{M-2,M}} \frac{1}{2} \left\| x_{M-2} - \hat{x}_{M-2} \right\|_{P_{M-2}}^2 + \frac{1}{2} \sum_{k=M-1}^{M} \left\| \alpha_k \right\|_{Q^{-1}}^2 + \left\| \beta_k \right\|_{R^{-1}}^2 \left( \hat{x}_k - h(x_k) - H_k r_k \right)$$

$$+ \frac{1}{2} \sum_{k=M-2}^{M-1} \left\| \alpha_k \right\|_{Q^{-1}}^2 + \alpha_k^\top (x_{k+1} - f(x_k) - G_k w_{k+1}).$$

Since $\hat{x}_{M-1|M}$, $\hat{x}_k^M_{k=1}$ and $(P_k)^M_{k=1}$ are given, $(v_k)^M_{k=1}$ and $w_{M-1}$ are deterministic, $(\beta_k)^M_{k=M-1}$ and $\alpha_{M-1}$ are free variable, hence we have

$$Z_{M-2,M}^* := \arg\min_{Z_{M-2,M}} S_{M-2,M} = \arg\min_{Z_{M-2,M}} \frac{1}{2} \left\| x_{M-2} - \hat{x}_{M-2} \right\|_{P_{M-2}}^2$$

$$+ \left[ \frac{1}{2} \left\| w_{M-2} \right\|_{Q^{-1}}^2 + \alpha_{M-2}^\top \left( \hat{x}_{M-1|M} - f(x_{M-2}) - G_{M-2} w_{M-2} \right) \right],$$

and

$$\hat{x}_{M-2|M}^{(2)} = \hat{x}_{M-2|M}^{(1)} + P_{M-2}A_{M-2}^\top P_{M-1|M-2}^{-1} \left( \hat{x}_{M-1|M} - f(\hat{x}_{M-2|M}) \right).$$

Therefore, by induction, estimating $(\hat{x}_k)_{k=1}^M$ using the EKS is equivalent to solving (2.23) by only one backtracking line search step, given $(\hat{x}_k)_{k=1}^M$ and a carefully chosen initial guess

$$Z_{M}^{(1)} := \begin{bmatrix} 0_{1,2n+q+p} & \hat{x}_1^\top \\ 0_{1,r+q+n+p} & \hat{x}_2^\top \\ \vdots \\ 0_{1,r+q+n+p} & \hat{x}_M^\top \end{bmatrix}^\top.$$

Hence, by optimality, the state estimate using the FIE will have a smaller error $e_{k|M}$ than the EKS.
4.2 Expectation Maximization Method with the Full Information Estimation

After the state sequences \(\hat{x}_k^{M}\) and \(\hat{x}_k|M\) are obtained from (2.26) and (2.23), respectively, the time-varying matrices \(A_k\) and \(C_k\) are determined using (2.15), hence \(P_k|M\) and \(P_{k-1}|M\) in (4.6) can both be determined by (2.4) and (2.53).

We formulate the noise covariance estimation strategy in Algorithm 4.2, which will iterate until the condition at step 10 is satisfied. The main problem with Algorithm 4.2 is that step 3 could be time-consuming, especially when the number of decision variables and the horizon length are large. Instead of using an MHE, an EKF and EKS (4.9) could be used to provide the initial guess of (2.23) with

\[
X^{(1)}_{1,M} := \begin{bmatrix} \hat{x}_{1|M}^\top & 0 & \hat{x}_{2|M}^\top & \cdots & \hat{x}_M^\top & 0 & \hat{x}_{M+1|M}^\top \end{bmatrix}^\top,
\]

with the assumptions that \(A_k \approx A_k|M\) and \(C_k \approx C_k|M\), which we call “Fast Algorithm 4.2”.

A different approach for estimating the covariances for nonlinear systems is provided in [5], which replaces steps 3 to 5 in Algorithm 4.2 by EKF/EKS, assumes \((G_k)^M\) and \((H_k)^M\) and provides an equivalent expression of \(\bar{C}_Q\) and \(\bar{C}_R\), such that

\[
\bar{C}_Q := \sum_{k=1}^{M-1} P_{k+1|M} - A_{k|M} P_{k+1,k|M} - P_{k+1,k|M} A_{k|M}^\top + A_{k|M} P_{k,M} A_{k|M}^\top + \hat{x}_{k+1|M} \hat{x}_{k+1|M}^\top - \hat{x}_{k+1|M} f^\top(\hat{x}_{k|M}) - f^\top(\hat{x}_{k|M}) \hat{x}_{k+1|M} + f^\top(\hat{x}_{k|M}) f(\hat{x}_{k|M}), \tag{4.12a}
\]

\[
\bar{C}_R := \sum_{k=1}^{M} C_{k|M} P_k C_{k|M}^\top + y_k y_k^\top - y_k h^\top(\hat{x}_{k|M}) - h^\top(\hat{x}_{k|M}) y_k + h^\top(\hat{x}_{k|M}) h(\hat{x}_{k|M}), \tag{4.12b}
\]

For determining the noise covariances, the analytical expression (4.8) is used in [5], the main problem is that (4.12) involves addition of large numbers with opposite sign (plus/minus), which cannot always guarantee the positive definiteness of \(\bar{C}_Q\) and \(\bar{C}_R\) due to numerical errors.

An MATLAB example in Listing 4.1 demonstrates that numerical error can be eliminated by reducing the numbers with opposite sign in the equation.

Therefore, in order to improve the estimation accuracy, expression (4.6) is preferred instead of (4.12). Moreover, the covariance estimates \(Q_i\) and \(R_i\) are guaranteed to be positive definite, if an SDP solver is applied rather than using (4.8). More details about this problem will be discussed in Section 4.3.2.

The main advantage of using the EKF/EKS is that this is faster than the MHE/FIE; however, the EKF/EKS suffers from instability and estimation accuracy problems. By using the MHE algorithm given in flow chart 2.1, if both \(\|w_k\|\) and \(\|v_k\|\) are bounded, then stability of state estimation can be guaranteed by Theorem 2.3. Propositions 2.6 and 4.2 ensure the
MHE/FIE can provide more accurate state estimates \((\hat{x}_k)_{k=1}^{M}\) and \((\hat{x}_{k|M})_{k=1}^{M}\) than the EKF and EKS.

### 4.3 Numerical Examples

We present three nonlinear examples to investigate the performance of our ALS and EM methods. In the first example, we assume the state noise \(w_k\) is not additive to the nonlinear dynamic, which is incompatible with the EM method, thus only the ALS method will be used. All results in this section are based on an Intel i7 5960X Octa-core CPU at 4.43 GHz, MATLAB 2015a, IPOPT v3.12.1 and SDPT3 v4.0 solver. Figure 4.1 shows the time taken versus the number of CPU cores when computing the ALS algorithm with \(M_e = 1000\), \(N = 150\) and \(n = 3\), based on an Intel i7 5960X CPU.

#### 4.3.1 Nonlinear System 1

Consider a nonlinear system, for \(k = 1, \cdots, 500\)

\[
\begin{align*}
    x_{k+1} &= \sin(x_k T_s + w_k), \\
    y_k &= x_k + v_k
\end{align*}
\]

where \(T_s = 0.1\) and noises \(w_k\) and \(v_k\) satisfy Assumption 2.6. We generate reference output \((y_k)_{k=1}^{500}\) with initial guess \(x_1 = 1\) and noise covariances \(Q = 0.01\) and \(R = 0.001\). The initial state and corresponding error covariance are set to \(\hat{x}_{1|0} = 0\) and \(P_{1|0} = 1\), respectively.
Algorithm 4.2 Noise Covariance Estimation using the EM combined with the MHE/FIE

\[ O = \text{EM/FIE} \left( f(\cdot), h(\cdot), O_1, (y_k)_{k=1}^M, H_l, \zeta \right) \]

1: Let \( b \leftarrow 0 \) and \( i \leftarrow 1 \),
2: \textbf{while} \( b = 0 \) \textbf{do}
3: \hspace{1em} Estimate sequences \((\hat{w}_k)_{k=1}^{M-1}\) and \((\hat{x}_k, \hat{v}_k)_{k=1}^M\) using the MHE algorithm provided in Figure 2.1 with horizon \( H_l \).
4: \hspace{1em} Let \( \chi_{1,M}^{(1)} := \begin{bmatrix} \hat{x}_1^T & \hat{v}_1^T & \hat{w}_1^T & \hat{x}_2^T & \cdots & \hat{w}_{M-1}^T & \hat{x}_M^T & \hat{v}_M^T \end{bmatrix}^T \).
5: \hspace{1em} Obtain the smoothed state sequence \((\hat{x}_k|_M)_{k=1}^M\) by solving (2.23) using \( \chi_{1,M}^{(1)} \) obtained from the previous step as the initial guess.
6: \hspace{1em} Calculate \((A_k, C_k)_{k=1}^M\) using (2.15).
7: \hspace{1em} Calculate \((P_k|_M, P_{k,k-1}|_M)_{k=1}^M\) using (2.4) and (2.53).
8: \hspace{1em} Calculate \((A_k|_M, C_k|_M)_{k=1}^M\) using (4.2).
9: \hspace{1em} Formulate \( O(\cdot|O_{i-1}) \) using (4.6) and hence obtain \( O_{i+1} \) by solving (2.54).
10: \hspace{1em} \textbf{if} \( \|\hat{Q}_i - \hat{Q}_i\|_F < \zeta \) and \( \|\hat{R}_i - \hat{R}_i\|_F < \zeta \) \textbf{then}
11: \hspace{1.5em} Let \( b \leftarrow 1 \).
12: \hspace{1em} \textbf{else}
13: \hspace{1.5em} Let \( i \leftarrow i + 1 \), \( \hat{x}_{i|0,i+1} \leftarrow \hat{x}_{1|_M} \) and \( P_{1|0,i+1} \leftarrow P_{1|_M} \).
14: \hspace{1em} \textbf{end if}
15: \textbf{end while}

Fig. 4.1 Time taken vs. the number of CPU cores for computing the ALS algorithm
Fig. 4.2 Estimation of noise covariances $Q^*$ and $R^*$

The guessed noise covariances are set to $Q_g = 1$ and $R_g = 1$. Similar to the LTV example in Section 3.5.1, we repeat the simulation 200 times using the ALS-EKF method mentioned in Section 4.1.1 with $N = 10$, $M_e = 500$. Figure 4.4 contains scatter plots of 200 estimates of the noise covariances $Q^*$ and $R^*$, as well as the average of all 200 estimates, which shows a good estimate of both $Q$ and $R$.

### 4.3.2 Nonlinear System 2

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

$$
\begin{align*}
    a_{k+1} &:= +a_k \cos(c_k T_s) + b_k \sin(c_k T_s) \\
    b_{k+1} &:= -a_k \sin(c_k T_s) + b_k \cos(c_k T_s) + c_k \\
    c_{k+1} &:= f(x_k) + w_k \\
    y_k &:= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_k + v_k
\end{align*}
$$
4.3 Numerical Examples

where $T_s = 0.1$ sec is the sampling time and $w_k$ and $v_k$ satisfy Assumption 2.6. The time-varying frequency $c_k$ is now part of the unknown state $x_k$, which transfers our previous LTV example in Section 3.5.1 into a nonlinear example. We now randomly pick an initial state from a uniform distribution, such that

$$x_1 := \begin{bmatrix} x_1^1 \sim \mathcal{U}(2,4) \\ x_2^2 \sim \mathcal{U}(-4,-2) \\ x_3^3 \sim \mathcal{U}(0.3,0.7) \end{bmatrix},$$

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

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

The guessed initial state error covariance is $P_{1|0} = P_g = 0.1 \times I_3$ and the guessed initial state is $\hat{x}_{1|0} = [15 \quad -15 \quad 15]^T$, the guessed noise covariance matrices are set to $Q_g = I_3$ and $R_g = 1$.

Similar to the LTV example in Section 3.5.1, we repeat the simulation 200 times using both the MHE and EKF methods mentioned in Sections 4.1.1 and 4.1.2 with $N = 150$, $M_e = 1000$ and horizon length $H_l = 300$ for MHE. We also compare ALS based methods with three different EM based covariance estimation methods introduce in Section 4.2 and [5].

Figures 4.3 to 4.5 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 4.1. Figures 4.6, 4.7 and 4.8 are the plots of the system noise covariance estimates using Algorithm 4.2, Fast Algorithm 4.2 and the EKF+EKS based EM method, respectively, over 1500 iterations. Figure 4.9 is a plot of the log-likelihood function $\Omega$ versus the number of iterations. Because there are few estimates of $Q_i$ between $i = 275$ and $i = 309$ are not positive definite, for the EKF+EKS method, only the real part of complex log-likelihood function $\Omega$ is plotted.

Table 4.1 presents the mean error percentage $\bar{p}_Q$, $\bar{p}_R$ and the variances of $e_Q$ and $e_R$ using different $M_e$ as well as $\tilde{T}_c$ and $\tilde{T}_s$ of the optimization problem (3.9). Table 4.1 clearly shows that as $M_e$ goes to $M$, error percentage $\bar{p}_Q$ and the variance of $e_Q$ are getting smaller, but the $\tilde{T}_c$ are dramatically increased.

Table 4.2 compares noise covariance estimation errors and error percentages using 5 different estimation algorithms that introduced in Sections 4.1, 4.2 as well as in [5]. Table 4.2 shows that Algorithm 4.1, Algorithm 4.2 and fast Algorithm 4.2 can provide good estimates with similar accuracy. The EM algorithm given by [5] has larger estimation error on $Q$, because [5] calculates $Q$ based on the EKF/EKS rather than the MHE. The ALS+EKF method...
Noise Covariance Identification for Nonlinear Systems using Auto-covariance Least Squares and Expectation Maximization Method

Fig. 4.3 Estimation of noise covariances $Q^*$ using algorithm 4.1

Fig. 4.4 Estimation of noise covariances $Q^*$ and $R^*$ using algorithm 4.1
4.3 Numerical Examples

Fig. 4.5 Estimation of noise covariances $Q^*$ using algorithm 4.1

Fig. 4.6 Noise covariance estimation using algorithm 4.2

Fig. 4.7 Noise covariance estimation using fast algorithm 4.2

Fig. 4.8 Noise covariance estimation using the EM+EKS
4.3 Numerical Examples

Fig. 4.9 Log-likelihood function $\mathcal{Q}$ versus number of iterations

Table 4.1 Noise covariance estimation results using different $M_e$

<table>
<thead>
<tr>
<th>$M_e$</th>
<th>$\bar{p}_Q(%)$</th>
<th>Variance of $e_Q$</th>
<th>$\bar{p}_R(%)$</th>
<th>Variance of $e_R$</th>
<th>$T_e$ (sec)</th>
<th>$T_s$ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>282.8</td>
<td>$4.637 \times 10^{-6}$</td>
<td>11.32</td>
<td>$1.823 \times 10^{-9}$</td>
<td>0.158</td>
<td>0.3292</td>
</tr>
<tr>
<td>400</td>
<td>57.28</td>
<td>$1.210 \times 10^{-7}$</td>
<td>7.51</td>
<td>$6.384 \times 10^{-10}$</td>
<td>4.384</td>
<td>0.3634</td>
</tr>
<tr>
<td>600</td>
<td>36.68</td>
<td>$3.309 \times 10^{-8}$</td>
<td>4.91</td>
<td>$3.928 \times 10^{-10}$</td>
<td>25.69</td>
<td>0.3482</td>
</tr>
<tr>
<td>800</td>
<td>30.80</td>
<td>$1.781 \times 10^{-8}$</td>
<td>2.70</td>
<td>$2.883 \times 10^{-10}$</td>
<td>81.31</td>
<td>0.3435</td>
</tr>
<tr>
<td>1000</td>
<td>26.93</td>
<td>$1.329 \times 10^{-8}$</td>
<td>3.99</td>
<td>$2.392 \times 10^{-10}$</td>
<td>185.5</td>
<td>0.3563</td>
</tr>
</tbody>
</table>

Table 4.2 Noise covariance estimation results

<table>
<thead>
<tr>
<th></th>
<th>$e_Q(10^{-5})$</th>
<th>$e_R(10^{-5})$</th>
<th>$\bar{p}_Q(%)$</th>
<th>$\bar{p}_R(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM based Algorithm</td>
<td></td>
<td></td>
<td>$7.79$</td>
<td>$1.82$</td>
</tr>
<tr>
<td>Alg. 4.2</td>
<td></td>
<td></td>
<td>$7.97$</td>
<td>$1.80$</td>
</tr>
<tr>
<td>Fast Alg. 4.2</td>
<td></td>
<td></td>
<td>$49.9$</td>
<td>$1.2$</td>
</tr>
<tr>
<td>EM+EKS [5]</td>
<td></td>
<td></td>
<td>$49.9$</td>
<td>$1.2$</td>
</tr>
<tr>
<td>ALS based Algorithm</td>
<td></td>
<td></td>
<td>$1.32$</td>
<td>$3.99$</td>
</tr>
<tr>
<td>Alg. 4.1</td>
<td></td>
<td></td>
<td>$2.243 \times 10^4$</td>
<td>$158.7$</td>
</tr>
<tr>
<td>ALS+EKF</td>
<td></td>
<td></td>
<td>$2.243 \times 10^4$</td>
<td>$158.7$</td>
</tr>
</tbody>
</table>
fails to provide acceptable results, due to the divergence of the state estimate and “one-off” nature of the ALS based estimation algorithm.

By given the same output measurements and initial guesses of unknown parameters, we determined $Q^*$, $R^*$, $P^*_{1|0}$ and $\hat{x}_{1|0}$ using 5 different algorithms, hence used the EKF to estimate system states $x_k$. Table 4.3 compares covariance estimates, $\bar{p}_x$ and time taken $T$ amongst 5 different noise covariance estimation algorithms. Table 4.3 shows that all EM based algorithm are able provide the same $\bar{p}_x$ as the true system; however, the best choice of covariance estimation algorithm for this example is the ALS based Alg. 4.1, because this provides a much faster estimation with a small sacrifice in the state estimation accuracy.

As we discussed at the end of Section 4.2, table 4.4 examines the numerical robustness based on the EM+EKS method with different ways to maximize the $Q(\theta|\theta_{i-1})$. $T_{np}$ is number of times that $Q_i$ is NOT positive definite within 1500 iterations. Table 4.4 shows that by replacing (4.12) with (4.7), and/or using the SDP solver, all the $Q_i$ within 1500 iterations are positive definite and has a smaller error percentage $p_Q$.

### 4.3.3 Nonlinear System 3

Consider the nonlinear chemical process introduced in [17]:

$$
\begin{bmatrix}
x_k^{(1)} \\
x_k^{(2)} \\
x_{k+1}
\end{bmatrix} := \begin{bmatrix}
x_k^{(1)} \\
x_k^{(2)}
\end{bmatrix} + \begin{bmatrix}
\frac{x_k^{(1)}}{1+2k_rT_s x_k^{(1)}} \\
\frac{k_r T_s x_k^{(1)}}{1+2k_r T_s x_k^{(1)}}
\end{bmatrix} + \begin{bmatrix}
w_k^{(1)} \\
w_k^{(2)}
\end{bmatrix},
$$

$$y_k := C x_k + v_k$$

where $x_k \in \mathbb{R}^{2 \times 1}$, $k_r = 0.16$, the sampling period $T_s = 0.01$ s, $C = [1 \ 1]^T$, the initial state $x_0 = [3 \ 1]^T$, $w_k \in \mathbb{R}^{2 \times 1}$ and $v_k \in \mathbb{R}^1$ satisfy Assumption 2.6, which $w_k^{(1)}$ and $w_k^{(2)}$ are uncorrelated. We generate 7.5 seconds of output measurements $(y_k)_{k=1}^{750}$ based on noise covariance matrices $Q = 10^{-6} \times I_2$ and $R = 1 \times 10^{-2}$.

We will use Algorithm 4.2, Fast Algorithm 4.2 and the EKF/EKS based EM method provided in [5] to estimate the true noise covariances $Q$ and $R$. Because noise covariances are determined by an SDP solver rather than an analytical expression (4.8), we are able to add an extra constraint such that decision variable $Q$ in (2.54) has zero off-diagonal entries. For easier comparison, rather than setting up a terminal condition $\zeta$, the maximum number of iterations $N$ is fixed to 3000. Let $H_I = 200$, the initial guess of $P_{1|1}$, $Q_1$ and $R_1$ are $50 \times I_2$, $Q_1 = I_2$ and $R_1 = 1$, respectively, with initial state guess $\hat{x}_1 = [1 \ 45]^T$. 

### Table 4.3 Noise covariance estimation results

|                  | $P_1$         | $\hat{x}_{1|0}$ | $Q$          | $R$          | $\bar{p}_x(\%)$ |
|------------------|---------------|-----------------|--------------|--------------|-----------------|
| Initial Guess    | $0.1 \times I_3$ | $\begin{bmatrix} 15 \\ -15 \\ 15 \end{bmatrix}$ | $I_3$        | 1            | 185.40          |
| True System      | $I_3 \times 10^{-6}$ | $\mathcal{N}(x_1, P_1)$ | $\begin{bmatrix} 3 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix} \times 10^{-4}$ | $1 \times 10^{-4}$ | 2.83            |

<table>
<thead>
<tr>
<th></th>
<th>$P^{*}_{10}(\times 10^{-6})$</th>
<th>$Q^{*}(\times 10^{-3})$</th>
<th>$R^{*}(\times 10^{-4})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg. 4.2</td>
<td>0.38 -0.15 0.21</td>
<td>0.30 -0.01 0.15</td>
<td>1.18</td>
</tr>
<tr>
<td></td>
<td>-0.15 15.25 2.11</td>
<td>-0.01 0.29 0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.21 2.11 0.60</td>
<td>0.15 0.01 0.21</td>
<td></td>
</tr>
<tr>
<td>Fast Alg. 4.2</td>
<td>0.06 -0.04 0.03</td>
<td>0.30 -0.01 0.15</td>
<td>1.18</td>
</tr>
<tr>
<td></td>
<td>-0.04 15.37 2.25</td>
<td>-0.01 0.30 0.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.03 2.25 0.53</td>
<td>0.15 0.00 0.21</td>
<td></td>
</tr>
<tr>
<td>EM+EKS [5]</td>
<td>0.06 -0.17 0.00</td>
<td>0.32 0.09 0.13</td>
<td>1.12</td>
</tr>
<tr>
<td></td>
<td>-0.17 27.19 5.09</td>
<td>0.09 0.78 0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.00 5.09 1.54</td>
<td>0.13 0.01 0.20</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$P^{*}_{10}(\times 10^{-2})$</th>
<th>$Q^{*}(\times 10^{-3})$</th>
<th>$R^{*}(\times 10^{-4})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg. 4.1</td>
<td>0.05 0.69 0.12</td>
<td>0.31 -0.12 0.15</td>
<td>1.14</td>
</tr>
<tr>
<td></td>
<td>0.69 11.18 2</td>
<td>0.12 0.09 0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.12 2 0.36</td>
<td>0.15 0.01 0.24</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$P_1$</th>
<th>$Q^*$</th>
<th>$R^{*}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALS+EKF</td>
<td>$\begin{bmatrix} 149.7 \ -181.1 \ -8.29 \end{bmatrix}$</td>
<td>$\begin{bmatrix} -181.1 \ 310.0 \ 20.0 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.11 \ 0.47 \ 0.07 \end{bmatrix}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Alg. 4.2</th>
<th>Fast Alg. 4.2</th>
<th>EM+EKS</th>
<th>Alg. 4.1</th>
<th>ALS+EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{x}_{1</td>
<td>0}$</td>
<td>$\begin{bmatrix} 2.94 \ -3.01 \ 0.54 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 2.94 \ -2.99 \ 0.54 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 2.94 \ -2.49 \ 0.61 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 2.94 \ -3.41 \ 0.48 \end{bmatrix}$</td>
</tr>
<tr>
<td>$\bar{p}_x(%)$</td>
<td>2.95</td>
<td>2.97</td>
<td>3.13</td>
<td>3.49</td>
<td>156.76</td>
</tr>
<tr>
<td>$T$ mins</td>
<td>1317</td>
<td>11.40</td>
<td>5.03</td>
<td>1.85</td>
<td>0.35</td>
</tr>
</tbody>
</table>
Table 4.4 Robustness tests for EM+EKS based noise covariance estimation

| How to maximize $\mathbf{Q}(O|O_{i-1})$ | $T_{np}$ | $PQ(\%)$ |
|----------------------------------------|--------|--------|
| (4.12)+(4.8)                           | 6      | 101.8  |
| (4.12)+SDP solver                      | 0      | 93.16  |
| (4.7)+(4.8)                            | 0      | 93.16  |
| (4.7)+SDP solver                       | 0      | 93.16  |

Figure 4.10 is the plot of the smoothed state estimates using FIE and EKS with guessed noise covariance. Figure 4.10 clearly shows that FIE converges to the neighborhood of the true state trajectory very quickly, while EKS estimates are still far away from the true state trajectory. The main reason behind this is because FIE is able to add the constraint on the system state $x_k$ that the partial pressure $x_k$ is non-negative.

Figures 4.11 to 4.13 are the plots of the system noise covariance estimates using Algorithm 4.2, Fast Algorithm 4.2 and the EKF/EKS based EM method, respectively, over 3000 iterations. In Figures 4.11 and 4.13, the estimated noise covariances using Algorithm 4.2 and Fast Algorithm 4.2 are almost identical and much closer to the true values, compared with results using the EKF/EKS based EM method. The estimated $Q_i$ and $R_i$ using (4.8) in Figures 4.12 and 4.13 are much worse than others, this is because the positive definiteness of $Q_i$ using (4.8) has been violated several times due to the numerical errors.

Figure 4.14 is a plot of the log-likelihood function $Q$ versus the number of iterations. For the EKF/EKS method using (4.8), only the real part of complex $Q$ is shown. Figure 4.14 shows that only the trajectories using Algorithm 4.2 and fast Algorithm 4.2 are asymptotically converging to the neighborhood of the reference line, which is calculated using the function $Q$ with true covariances.

It is also interesting to compare the EM method with the ALS method introduced in Section 4.1. We keep the data length $M$ to 750 and the sub-optimal filter gains $(L^s_k)_{k=1}^M$ and the state error covariance $(P_k)_{k=1}^M$ are obtained from the EKF equations from the same initial conditions and guessed noise covariance matrices $Q_g = Q_1$ and $R_g = R_1$. We re-generate output measurements from the same initial state $x_0 = \begin{bmatrix} 3 & 1 \end{bmatrix}^T$ and different noise sequences based on noise covariances $Q$ and $R$, hence repeat the simulation 200 times using both the EKF and MHE methods mentioned in Section 4.1 with $N = 100$, $M_e = 750$ and $H_l = 250$, and calculate the average values of $e_Q$ and $e_R$ over 200 simulations as well as the error percentages.

Table 4.5 compares the estimation error norms and percentages using different noise covariance estimation methods. For the nonlinear system (4.13), Table 4.5 clearly shows that
4.3 Numerical Examples

Fig. 4.10 Estimates of the state trajectory using $Q_1$ and $R_1$

Fig. 4.11 Estimates of the state noise covariance
Fig. 4.12 Estimates of the state noise covariance using EKF/EKS in [5]

Fig. 4.13 Estimates of output noise covariance
noise covariance estimation using Algorithm 4.2 and Fast Algorithm 4.2 give almost identical results, which are more accurate than the ALS based methods. The reason behind this is that the ALS method estimates noise covariances by minimizing the difference between the auto-covariance of the estimates (as a function of $Q$ and $R$) and true innovation sequence. Unfortunately, if the system is nonlinear with a bad initial guess of noise covariances, then the auto-covariance of the estimated innovation sequence may not be approximated by a linear function of true noise covariances, which will significantly affect the ALS estimations. On the other hand, although our EM based algorithms involve approximations of the $Q$ function and the global optimum is not guaranteed for any initial guess [53], the EM method will keep increasing the likelihood function $\mathcal{L}_\ell (\cdot | Y_M)$ until it has converged. This property could help us to find a better estimate of noise covariances for some nonlinear systems.

Table 4.6 compares estimated $P_1$, $\hat{x}_1$, $\bar{p}_x$ and the time taken $T$ amongst 5 different noise covariance estimation algorithms. Table 4.6 shows that MHE/FIE-based methods have relatively small state estimation errors compared with EKF/EKS-based methods, which indicates that the initial conditions $P_1$ and $\bar{x}_1$ also play an important role in nonlinear state estimation. The MHE-based method is more time consuming than the EKF method and the EM-based method takes much longer than the ALS method due to its iterative nature. In our
example, for 3000 iterations, “Fast Algorithm 4.2” greatly speeds up the EM procedure by about 50 times compared to the original Algorithm 4.2.

4.4 Conclusions

In this chapter, we introduced two ALS and EM based noise covariance estimation algorithms for nonlinear systems. We extended the ALS algorithm introduced in Chapter 3 by linearizing nonlinear functions around the current state estimate. Instead of using EKF/EFS to estimate system states, we combined the MHE/FIE with the ALS and EM algorithm so the stability and accuracy of noise covariance estimation can be improved. The first numerical example has shown that the ALS-EKF based noise covariance estimation method is able to handle a nonlinear system with non-additive noise terms. Another two numerical examples were used to investigate and compare the performance amongst different estimation algorithms, both examples have shown that by using MHE/FIE the performance of noise covariance estimation can be significantly improved, especially when guessed initial state and noise covariances are far from their true values. For the second example, only our EM algorithm is able to provide accurate covariance estimates. In order to further improve the accuracy and ensure the positive definiteness, our EM problem is formulated into a constrained optimization problem and uses a semi-definite programming (SDP) solver to estimate the covariances. We also introduced a fast EM algorithm that used EKS instead of MHE to provide an initial guess for the FIE. Both example have shown that our fast algorithm will speed up the EM process by about 50 times compared with the EM-FIE method, with almost identical results.
Table 4.6 Noise covariance estimation results

| Algorithm | $P_1$ | $\hat{x}_{1|0}$ | $\hat{Q}$ | $R$ | $\bar{p}_{x}(\%)$ |
|-----------|-------|-----------------|-----------|-----|------------------|
| Initial Guess | $50 \times I_2$ | $\frac{1}{45}$ | $I_2$ | 1 | 206.6 |
| True System | $I_2 \times 10^{-6}$ | $N(x_1, P_1)$ | $I_2 \times 10^{-6}$ | $1 \times 10^{-2}$ | 0.344 |
| Alg. 4.2 | $P_1^* (\times 10^{-5})$ | $\hat{x}_{1|0}$ | $Q^* (\times 10^{-5})$ | $R^*$ | $\bar{p}_{x}(\%)$ | $T$ mins |
| Fast Alg. 4.2 | 0.59 | -0.28 | 2.97 | 0.89 | 0.01 | 0.54 | 3543 |
| | -1.80 | 0.18 | 1.01 | 0.41 | |
| EM+EKS [5] | 25.9 | -27.9 | -1.73 | 3.9 | -3.7 | 0.01 | 91.22 | 5.4 |
| | -27.9 | 30.4 | 5.64 | -3.7 | 3.5 | |
| Alg. 4.1 | 0.078 | -0.079 | 2.96 | 0.352 | 0.01 | 2.12 | 2.34 |
| | -0.079 | 0.079 | 1.02 | 0.001 | |
| ALS+EKF | $P_1$ | $\hat{x}_{1|0}$ | $Q^* (\times 10^{-4})$ | $R^*$ | $\bar{p}_{x}(\%)$ | $T$ mins |
| | 17.2 | -18.1 | -8.3 | 0.0002 | 0.007 | 139.1 | 0.44 |
| | -18.1 | 19.1 | 13.04 | 0 | 6.91 | |
In order to predict future observations of a noise-driven system, we have to find a model that exactly or at least approximately describes the behavior of the system so that the current system state can be recovered from passed observations. However, sometimes it is very difficult to model a system accurately, such as real ocean waves, thus it is particularly interesting to analyze ocean waves’ properties in the time-domain using autoregressive (AR) models.

In this chapter, we firstly study the power spectral density (PSD) and signal smoothing of ocean waves. We introduce AR models and their equivalent state space representations for predicting future ocean wave elevations, where unknown parameters will be determined using ALS- and EM-based estimation algorithms. Finally, we compare our prediction model with an existing model given in [15].

5.1 Ocean Wave Spectrum, Cut-off Frequency and Prediction Horizon

In the literature, two types of spectrum are mainly used to show the energy distributions across different frequency components of ocean waves: the Pierson-Moskowitz (P-M) and Joint North Sea Wave Project (JONSWAP) spectrum. The JONSWAP spectrum is given
Fig. 5.1 Spectrum of measured ocean waves

by [18, pp. 160-161]:

$$S_f(\omega) := \frac{\alpha g^2}{\omega^5} e^{-\frac{5}{4} \left(\frac{\omega}{\omega_p}\right)^4} \left[ e^{-\frac{(\omega - \omega_p)^2}{2\sigma^2\omega_p^2}} \right],$$

where $\alpha$ is a constant scaling value that relates to the wind speed and fetch length. Typical values for $\alpha$ in the northern North Sea are in the range of 0.0081 to 0.01 [54]. $g$ is the gravitational acceleration $g = 9.81 \text{m/s}^2$, $\omega_p$ is the peak frequency $\omega_p = \frac{2\pi}{T_p}$, the parameter $\sigma$ is equal to 0.07 if $\omega < \omega_p$ and 0.09 otherwise, $T_p$ is the typical period.

The P-M spectrum is given by: [29, pp. 33-34]

$$S_p(\omega) := \frac{\alpha g^2}{\omega^5} e^{-\frac{5}{4} \left(\frac{\omega}{\omega_p}\right)^4}.$$ 

Figure 5.1 is the wave spectrum that was recorded at Galway at 5:20 on the 10th of February 2005 with a sampling frequency of 2.56Hz. When predicting a signal from historical data using an AR-based model, it is often the case that high frequency disturbances in the historical data may significantly affect the prediction performance [15]. Thus, historical data is often smoothed using a finite impulse response (FIR) low-pass filter before passing through the AR model for forecasting.
Figure 5.1 shows that the energy spectrum is spread over 0 to 3 rad/s, where the high energy components are located within a lower frequency range between 0 and 1.5 rad/s. Hence, our main focus would be on analyzing lower frequency components. If a low-pass filtered historical waveform is used in the wave prediction model, then the WEC will not be able to extract energy from those frequency components that are ignored by the prediction model, although very high frequencies may be lost anyway due to the low-pass filtering dynamics of the WEC device itself [15]. Therefore, the choice of cut-off frequency $f_c$ of the low pass filter can be seen as a compromise between the performance of wave height forecasting and the loss of energy carried by higher frequency components of the incident wave. The difference between the original and smoothed signal are measured using the percentage of fitness according to

$$\text{Fit\%} := 100\% - \frac{\|\text{Original Signal} - \text{Smoothed Signal}\|_2}{\|\text{Original Signal}\|_2} \times 100\%,$$

with fitness usually between 70% and 75%. Figure 5.2 shows an example of a 72% fitness smoothed wave measurements with cut-off frequency $f_c = 1.3$ rad/s.

The prediction horizon $T_f$ is one of the important parameters in ocean wave forecasting, which indicates how many seconds we would like to predict the wave elevations into the
future; we defined that \( T_f = KT_s \), where \( K \) is a positive integer. The prediction horizon should be long enough for the WEC to respond; typically this should be approximately equal to half of the typical period \( T_p \) \cite{14}, i.e.

\[
T_f = \frac{T_p}{2} = \frac{\pi}{\omega_p},
\]

Figure 5.1 indicates that the peak frequency \( \omega_p \) is located at 0.6 rad/s, thus the prediction horizon \( T_f \approx 5 \text{ sec} \) should be long enough.

### 5.2 Autoregressive Prediction Model

Suppose we have a sufficient amount of historical wave data \( \{\eta_k\}_{k=1}^M \), and assume the current wave height to be linearly dependent on a number \( n \) \( (n \ll M) \) of its past values \cite{15}:

\[
\eta_{k+n} := \sum_{i=1}^{n} \tilde{a}_i \eta_{k+n-i} + v_{k+n}, \tag{5.1}
\]

where \( \tilde{a}_i \in \mathbb{R} \) is the parameter of the AR model and \( v_k \sim \mathcal{N}(0, R) \). If we let \( y_k := \eta_k \),

\[
x_k := \begin{bmatrix} \tilde{a}_1 & \tilde{a}_2 & \cdots & \tilde{a}_n \end{bmatrix}^\top \quad \text{and} \quad C_k := \begin{bmatrix} \eta_{k-1} & \eta_{k-2} & \cdots & \eta_{k-n} \end{bmatrix},
\]

then an equivalent LTV state space representation of (5.1) can be written as

\[
x_{k+1} = x_k, \quad y_k = C_k x_k + v_k, \quad k = m, m+1, \cdots, M, \tag{5.2}
\]

where \( 2n < m \ll M \). Hence, the initial state \( \hat{x}_m \) can be determined by solving a linear least squares (LLS) problem

\[
\hat{x}_m := \arg \min_{x_m} \| \tilde{\eta}_{n+1,m} - \tilde{H}_{1,m-1} x_m \|_2^2, \tag{5.3}
\]

where the matrix \( \tilde{H}_{1,m-1} \in \mathbb{R}^{(m-n) \times n} \) and vector \( \tilde{\eta}_{n+1,m} \in \mathbb{R}^{m-n} \) are determined from historical wave heights \( \{\eta_k\}_{k=1}^m \) such that

\[
\tilde{H}_{1,m-1} := \begin{bmatrix} \eta_n & \eta_{n-1} & \eta_{n-2} & \cdots & \eta_1 \\ \eta_{n+1} & \eta_n & \eta_{n-1} & \cdots & \eta_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \eta_{m-1} & \eta_{m-2} & \eta_{m-3} & \cdots & \eta_{m-n} \end{bmatrix}, \quad \tilde{\eta}_{n+1,m} := \begin{bmatrix} \eta_{n+1} \\ \eta_{n+2} \\ \vdots \\ \eta_{m} \end{bmatrix}, \quad \tilde{v}_{n+1,m} := \begin{bmatrix} v_{n+1} \\ v_{n+2} \\ \vdots \\ v_{m} \end{bmatrix}.
\]
Hence, if $\tilde{H}_{1,m-1}$ is full column rank, then $\hat{x}_1$ can be estimated by solving (5.3), i.e.

$$\hat{x}_m = \tilde{H}_{1,m-1}^{\top} \tilde{H}_{n+1,m}.$$

After the estimate of the initial state $\hat{x}_1$, we are able to track the constant state sequence $(x_k)_{k=2}^M$ using the recursive least squares (RLS) equations [15]

$$L_k = \tilde{P}_{k+1} C_k^{\top} \lambda^{-1}, \quad (5.4a)$$
$$\tilde{P}_{k+1} = \lambda \tilde{P}_k \left( C_k \tilde{P}_k C_k^{\top} + \lambda \right)^{-1}, \quad (5.4b)$$
$$\hat{x}_k = \hat{x}_{k-1} + L_k (y_k - C_k \hat{x}_{k-1}), \quad (5.4c)$$

where the initial covariance matrix is set to $\tilde{P}_m = I_n$.

**Proposition 5.1.** For the LTV system (5.2), the Kalman filter equations (2.4) can be simplified as the RLS equations (5.4) if $\tilde{P}_k = P_{k|k-1}$ and $\lambda = R$.

**Proof.** See Appendix E. \qed

After the optimal AR parameter vector $\hat{x}_M$ is determined, we could recursively predict future observations $\{\hat{\eta}_k\}_{k=M+1}^{M+T_f}$ by using

$$\hat{\eta}_{M+t} = \hat{x}_{M-t}^{\top} \tilde{H}_{n+1+t,M+t}, \quad t = 1, \cdots, T_f/T_s. \quad (5.5)$$

When the new observation $\eta_k, k > M$, becomes available, the AR parameter vector $\hat{x}_k$ will be updated using an RLS filter (5.4) based only on the most recent observation data $\eta_k$.

Alternatively, if the evolution of the state vector $x_k$ in (5.2) is assumed to be a random walk, then (5.2) can be replaced by [15]

$$x_{k+1} = x_k + w_k,$$
$$y_k = C_k x_k + v_k, \quad (5.6)$$

where $x_k := [\tilde{a}_{k,1} \quad \tilde{a}_{k,2} \cdots \quad \tilde{a}_{k,n}]^{\top}$ and Assumption 2.6 is hold. Since $y_k$ is a scalar, the state in (5.6) can be estimated using the following simplified Kalman filter equations:

$$L_k = P_{k|k-1} C_k^{\top} \left( C_k P_{k|k-1} C_k^{\top} + R \right)^{-1}, \quad (5.7a)$$
$$P_{k+1|k} = P_{k|k-1} + Q - P_{k|k-1} C_k^{\top} \left( C_k P_{k|k-1} C_k^{\top} + R \right)^{-1} C_k P_{k|k-1} \quad (5.7b)$$
$$= P_{k|k-1} + Q - \frac{P_{k|k-1} C_k P_{k|k-1} C_k^{\top}}{C_k P_{k|k-1} C_k^{\top} + R}. \quad (5.7c)$$
In [15], \( \hat{x}_M \) is estimated by solving a WLLS problem
\[
\hat{x}_M := \arg \min_{x_M} \| \tilde{\eta}_{1,M} - \tilde{H}_{1,M-1} x_M \|^2_{\Lambda},
\]
where the weight matrix \( \Lambda \in \mathbb{R}^{M \times M} \) is defined as \( \Lambda := \bigoplus_{k=0}^{M-1} \lambda^k \) and \( \lambda \in [0.97, 0.995] \) is the forgetting factor, so that more weight is given to recent observations according to an exponential law [15]. Hence, if \( \tilde{H}_{1,M-1} \) is a full column rank matrix, then
\[
\hat{x}_M = \left( \tilde{H}_{1,M-1}^\top \Lambda^{-1} \tilde{H}_{1,M-1} \right)^{-1} \tilde{H}_{1,M-1}^\top \Lambda^{-1} \tilde{\eta}_{n+1,M}.
\]
Once \( x_M \) is determined, future wave heights can be predicted using (5.5). When the new observation \( \eta_k \) becomes available, the AR parameters \( \hat{x}_k \) will be updated by [15]
\[
L_k = \tilde{P}_{k+1} C_k^\top, \tag{5.8a}
\]
\[
\tilde{P}_{k+1} = \tilde{P}_k \left( C_k \tilde{P}_k C_k^\top + \lambda \right)^{-1}, \tag{5.8b}
\]
\[
\hat{x}_k = \hat{x}_{k-1} + L_k (y_k - C_k \hat{x}_{k-1}). \tag{5.8c}
\]
with state error covariance \( \tilde{P}_M = \hat{x}_M \hat{x}_M^\top \).

There are two mistakes with the approach in [15]. Firstly, incorrect RLS equations (5.8) are used, which is only correct when \( \lambda = 1 \); secondly, an inappropriate state error covariance \( \tilde{P}_M \) is applied, because
\[
\tilde{P}_M = \mathbb{E} \left\{ (x_M - \hat{x}_M) (x_M - \hat{x}_M)^\top \right\} = \mathbb{E} \left\{ x_M x_M^\top \right\} - \hat{x}_M \hat{x}_M^\top = \hat{x}_M \hat{x}_M^\top.
\]
Our choice of \( \tilde{P}_M \) may not be correct either, but \( \tilde{P}_M \) will become accurate after several iterations of state estimation. Most of the time, these two mistakes may not lead to a serious problem, because [15] limits the choice of \( \lambda \) within 0.97 and 0.995, which makes (5.8) approximately equal to (5.4). \( P_M \) is a positive definite matrix, which may only have a limited effect on system stability and prediction accuracy. However, these two mistakes will cause a robustness problem if the measurements do not add new information to the system [15]. We will discuss more details in Section 5.3.2.

In [15], covariances \( Q, R \) and \( \lambda \) in (5.7) and (5.4), respectively, are all user-defined matrices. Inappropriate choices could result in poor or even unstable wave height predictions. In order to improve the prediction performance, one could use the ALS-based covariance estimation algorithms introduced in previous chapter to provide optimal covariance matrices for estimation models (5.4) and (5.6).
5.3 Performance of Wave Prediction

In this section, the performance of ocean wave prediction using models (5.4) and (5.6) will be examined and compared with existing methods. Unless otherwise stated, the ocean wave will be mainly based on the recording at Galway at 5:20 on the 10th of February 2005 with sampling frequency 2.56Hz [20]. The data file contains 3072 points (20 mins); the first 1400 points will be used as the historical wave. Because estimating noise covariance matrices requires some time, only the first 1000 points \((m < 250)\) will be used for estimating noise covariances. We are going to use the remaining points as the reference wave to examine the performance of ocean wave prediction.

For the ALS-based covariance estimation algorithms, we let \(M_e = 1000\), \(N\) and the initial state \(\hat{x}_m\) are determined by plotting the auto-correlation function of the innovation sequence (Figure 3.5) and solving (5.3), respectively. The initial guesses of \(P_m\), \(Q\) and \(R\) are set to \(I_n \times 10^{-7}\), \(I_n \times 10^{-7}\) and 10, respectively. In order to reduce the computational complexity, we assume that all covariances are diagonal matrices.

Forecasting accuracy is the way to determine the similarity between the predicted and true wave heights. The accuracy of prediction results are determined by the percentage of fitness:

\[
\text{Fit\%} := 100\% - \frac{\|\eta - \hat{\eta}\|_{\ell_2}}{\|\eta\|_{\ell_2}} \times 100\% \quad (5.9)
\]

where \(\eta\) is the measured wave heights, \(\hat{\eta}\) is the predicted wave heights \(T_f\) seconds into the future.

5.3.1 Cut-off Frequency and the Length of the AR Parameter Vector

Figures 5.3 and 5.4 show the prediction performance versus the cut-off frequency and size of the AR parameter vector using model (5.2) and (5.6), respectively. The best result for model (5.2) appears when \(f_c = 1.3\) rad/s and \(n = 33\); the best result for model (5.6) appears when \(f_c = 1.6\) rad/s and \(n = 80\).

If we keep the best settings of \(f_c\) and \(n\) for model (5.2) and (5.6), Figure 5.5 shows the predicted performance versus the size \(m\) for estimating the initial state \(x_m\). Figure 5.5 indicates that \(m = 140\) for model (5.2) and \(m = 185\) for model (5.6).

5.3.2 Robustness of AR Prediction Models

We proceed to test and compare the robustness of AR prediction models (5.2) and (5.6) with the AR model of [15] when there is a period of time without any signal. Figure (5.6)
Fig. 5.3 Percentage of fitness vs cut-off frequency and size of AR parameters using (5.2)

Fig. 5.4 Percentage of fitness vs cut-off frequency and size of AR parameters using (5.6)
5.3 Performance of Wave Prediction

![Graph showing percentage of fitness vs m for estimating the initial state $\hat{x}_m$](image)

Fig. 5.5 Percentage of fitness vs $m$ for estimating the initial state $\hat{x}_m$

illustrates that when the signal is resumed after approximately a 38 sec signal loss, wave prediction using the AR prediction model (5.2) and (5.6) show a strong tracking ability and good prediction performance, while the wave prediction using the AR model in [15] tends to infinity, due to mistakes we discussed in the previous section.

### 5.3.3 Which AR Prediction Model is Better?

In this section, we compare prediction results and computation time based on model (5.2) and (5.6). Both ALS and EM based covariance estimation methods will be used for model (5.2). However, only the ALS based method will be applied to model (5.6), for the following reasons: firstly, for a higher order system, using an iterative EM method will be very time consuming; secondly, because the determinant of the noise covariance $Q$ will be very small ($|Q| \ll 1 \times 10^{-200}$), calculating $\log |Q|$ in the EM algorithm may cause numerical problems.

Tables 5.1 and 5.2 show the accuracy Fit% and the computation time taken $T$ for noise covariance estimation between model (5.2) and (5.6) using 10 different ocean wave data. Tables 5.1 and 5.2 show that both models could reach a very similar accuracy level. However, model (5.2) results in a much faster computational time and slightly better results. Hence,
Fig. 5.6 Robustness tests for AR prediction with models

Table 5.1 Ocean wave prediction results with model (5.2)

<table>
<thead>
<tr>
<th>Time, Date</th>
<th>$f_c$ (rad/s)</th>
<th>$n$</th>
<th>$m$</th>
<th>Fit%</th>
<th>T (sec)</th>
<th>ALS (Chapter 3)</th>
<th>Fit%</th>
<th>T (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5:20 on 10/02/05</td>
<td>1.3</td>
<td>55</td>
<td>140</td>
<td>71.34</td>
<td>24.77</td>
<td>71.82</td>
<td>6.42</td>
<td></td>
</tr>
<tr>
<td>22:20 on 30/11/04</td>
<td>1</td>
<td>70</td>
<td>240</td>
<td>78.09</td>
<td>38.20</td>
<td>76.46</td>
<td>5.96</td>
<td></td>
</tr>
<tr>
<td>8:20 on 01/11/04</td>
<td>1.1</td>
<td>85</td>
<td>230</td>
<td>73.13</td>
<td>55.51</td>
<td>72.73</td>
<td>7.05</td>
<td></td>
</tr>
<tr>
<td>12:20 on 20/02/05</td>
<td>1.1</td>
<td>65</td>
<td>245</td>
<td>72.46</td>
<td>32.69</td>
<td>72.00</td>
<td>7.20</td>
<td></td>
</tr>
<tr>
<td>17:20 on 31/01/05</td>
<td>1.2</td>
<td>60</td>
<td>250</td>
<td>71.66</td>
<td>25.04</td>
<td>71.43</td>
<td>5.89</td>
<td></td>
</tr>
<tr>
<td>9:20 on 18/03/05</td>
<td>1.2</td>
<td>55</td>
<td>255</td>
<td>69.24</td>
<td>28.75</td>
<td>68.07</td>
<td>6.74</td>
<td></td>
</tr>
<tr>
<td>9:20 on 01/10/04</td>
<td>1.5</td>
<td>80</td>
<td>245</td>
<td>70.57</td>
<td>45.25</td>
<td>70.68</td>
<td>6.19</td>
<td></td>
</tr>
<tr>
<td>16:20 on 15/10/04</td>
<td>1.2</td>
<td>100</td>
<td>300</td>
<td>70.06</td>
<td>80.02</td>
<td>69.70</td>
<td>7.41</td>
<td></td>
</tr>
<tr>
<td>9:20 on 1/12/04</td>
<td>1</td>
<td>70</td>
<td>250</td>
<td>71.99</td>
<td>38.52</td>
<td>71.23</td>
<td>6.65</td>
<td></td>
</tr>
<tr>
<td>15:20 on 31/12/04</td>
<td>0.95</td>
<td>35</td>
<td>225</td>
<td>75.6</td>
<td>8.57</td>
<td>73.61</td>
<td>4.65</td>
<td></td>
</tr>
</tbody>
</table>
Table 5.2 Ocean wave prediction results with model (5.6)

<table>
<thead>
<tr>
<th>Time, Date</th>
<th>$f_c$ (rad/s)</th>
<th>$n$</th>
<th>$m$</th>
<th>Fit%</th>
<th>$T$ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5:20 on 10/02/05</td>
<td>1.3</td>
<td>85</td>
<td>300</td>
<td>67.56</td>
<td>109.91</td>
</tr>
<tr>
<td>22:20 on 30/11/04</td>
<td>0.95</td>
<td>65</td>
<td>160</td>
<td>75.62</td>
<td>65.84</td>
</tr>
<tr>
<td>8:20 on 01/11/04</td>
<td>1.1</td>
<td>50</td>
<td>140</td>
<td>68.92</td>
<td>34.65</td>
</tr>
<tr>
<td>12:20 on 20/02/05</td>
<td>1.1</td>
<td>75</td>
<td>260</td>
<td>71.57</td>
<td>69.37</td>
</tr>
<tr>
<td>17:20 on 31/01/05</td>
<td>1.3</td>
<td>40</td>
<td>240</td>
<td>66.53</td>
<td>46.78</td>
</tr>
<tr>
<td>9:20 on 18/03/05</td>
<td>0.95</td>
<td>35</td>
<td>150</td>
<td>65.33</td>
<td>46.03</td>
</tr>
<tr>
<td>9:20 on 01/10/04</td>
<td>1.4</td>
<td>60</td>
<td>260</td>
<td>68.92</td>
<td>75.83</td>
</tr>
<tr>
<td>16:20 on 15/10/04</td>
<td>1.1</td>
<td>100</td>
<td>265</td>
<td>65.52</td>
<td>148.57</td>
</tr>
<tr>
<td>9:20 on 1/12/04</td>
<td>1</td>
<td>60</td>
<td>250</td>
<td>70.09</td>
<td>67.52</td>
</tr>
<tr>
<td>15:20 on 31/12/04</td>
<td>0.95</td>
<td>35</td>
<td>225</td>
<td>75.53</td>
<td>36.50</td>
</tr>
</tbody>
</table>

The AR model (5.2) will be used for further tests and comparisons. Figure 5.7 shows that the objective function value of the EM method has converged after about 10 iterations.

### 5.3.4 Prediction Performance Comparison

In this section, we are going to compare the wave prediction performance based on the following 5 methods:

1. AR model in [15] with $\lambda = 0.97$

2. AR model (5.2) without estimating noise covariance (using only the initial guess $R = 10$)

3. AR model (5.2) with the ALS method of [35]

4. AR model (5.2) with ALS method of Chapter 3

5. AR model (5.2) with EM method of Chapter 2

Figure 5.8 shows that, in this case, the eigenvalues of $\prod_{k=2}^{2+N_k} (b_k A_k)$ do not converge to zero as $N_k$ increases to 1000, hence (3.11) cannot be fulfilled. Therefore, in order to keep enough data for the ALS estimation, we let $N_k = 100$. Hence, the estimation errors caused by model mismatch using the ALS method given in [35] will be investigated in the following tests.

Figure 5.9 illustrates 160 sec of ocean wave prediction with $T_f \approx 5$ sec prediction horizon using the 5 different methods. The percentage of fitness of the different methods are 56.56%, 51.64%, 58.54%, 71.34% and 71.82%, respectively. The AR model (5.2) with EM method
Fig. 5.7 Log-likelihood function $Q$ versus number of iterations

Fig. 5.8 Eigenvalues of matrix $\prod_{k=2}^{2+N_k} \bar{A}_k$ as $N_k$ increases
5.4 Conclusions

In this chapter, we focused on forecasting ocean wave elevations for wave energy converters. Two existing AR wave prediction models, as well as the prediction methods given in [15] were firstly reviewed. This was followed by discussions of the accuracy and stability issues of the wave prediction method of [15]. We provided solutions to improve robustness of the state estimation, then applied the ALS and the EM noise covariance estimation algorithms given in Chapter 3 and Section 2.9, respectively, to improve the accuracy of ocean wave prediction. After we tested the performance and efficiency of both model (5.2) and (5.6) using 10 different ocean wave data files, we found out that, by applying the ALS method, model (5.2) provided the fastest estimation time compared to model (5.6) with similar performance. Finally, we compared the wave prediction results using 5 different methods gives the best wave prediction result. Figure 5.10 is the percentage of fitness versus the prediction horizon, which shows that as the prediction horizon goes to 10 sec, only the AR model (5.2) with ALS and EM method are able to maintain a decent prediction performance.
Fig. 5.10 Wave prediction performances for different AR prediction models based on model (5.2). Results have shown that model (5.2) with EM gave the best prediction performance and is the fastest.
Chapter 6

Conclusions

If you love a flower that lives on a star, it is sweet to look at the sky at night. All the stars are a-bloom with flowers.

Le Petit Prince
Antoine de Saint-Exupéry

6.1 Contributions

In this thesis, we started with a brief review on state-of-the-art state estimation algorithms for linear and nonlinear systems, including the Kalman filter, Kalman smoother, EKF, FIE and MHE. We then provided proofs to show the properties and relationships between FIE and Kalman based filter/smoothers in both linear and nonlinear systems:

1. We proved that the FIE for an LTV system (2.1) is unbiased regardless of the choice of covariances $P_{1|0}$, $Q$ and $R$. However, if these covariances are accurate, then the FIE is BLUE for unknown system states. If matrices $G_k$ and $H_k$ are nonsingular, then the FIE problem (2.28) becomes an unconstrained WLLS problem and the proof becomes much easier, as has been done in [1]. We provided a generalized proof by letting $G_k$ and $H_k$ be non-square full column rank matrices.

2. We proved that solving the FIE problem (2.28) is equivalent to applying the Kalman filter and Kalman smoother for LTV system (2.1). The proof was inspired by [19].

3. We proved that for nonlinear systems, estimating the state using the FIE will always get more accurate results than using the EKF and smoother due to the optimality. Again, the proof was inspired by [19].
We also reviewed existing noise covariance estimation methods for linear systems, such as the auto-covariance least squares (ALS) and the expectation maximization (EM) algorithm. The ALS-based algorithm estimate noise covariances by solving one optimization, whereas the EM-based algorithm estimate noise covariances by recursively solving several optimization problems.

This thesis mainly focused on developing and analyzing covariance estimation algorithms for Kalman-based state estimators for both LTV and nonlinear systems. The first algorithm is based on the constrained (positive-definite) ALS method, which was first introduced in [35]; the new algorithm made several improvements:

1. The Assumption 1 in [35] has been removed, which assumes matrix $\prod_{k=2}^{2+N_k} (b) A_k$ will quickly converge to zero as $N_k$ increases. In Figures 3.9 and 5.8 we have shown that the assumption may not hold, if the given historical data is not long enough. By removing this assumption, our algorithm does not involve any approximations for covariance estimation in LTV systems, no longer needs to determine the parameter $N_k$ that is associated with the assumption and is able to determine the initial state error covariance $P_1$ that is required by the Kalman filter.

2. The ALS method in [35] for higher order LTV systems could require large amount of memory due to the Kronecker products. By rearranging the original ALS formulation, Figures 3.1 and 3.2 have shown that the new ALS algorithm is 60 times faster than the original ALS method, using only 0.001% of the original memory consumption.

3. For nonlinear systems, instead of using the EKF in the ALS method as in [35], we use MHE to estimate the nonlinear system state. By Proposition 2.6, the MHE is able to provide more accurate state estimates for nonlinear systems and capable of adding state constraints. After all the states are determined, one can linearize the system and hence construct the SDP problem (3.9).

4. We also provided a necessary condition for finding a unique solution in our ALS method.

After the ALS method, we introduced another covariance estimation method based on the expectation maximization (EM) method combined with FIE/MHE for nonlinear systems. The advantages of using FIE/MHE include: more accurate and stable state estimates compared with the EKF/smoother, the ability to add state constraints. After all states are determined, one can linearize the system around the current state estimate, construct (4.6) and hence the SDP problem (2.54). We also introduced a fast EM algorithm for nonlinear systems that use
the EFK rather than MHE to provide initial guesses for the FIE, which could provide similar accuracy 50 times faster than the original EM algorithm.

Finally, we applied our noise covariance estimation algorithm to improve an AR prediction model for forecasting ocean wave heights for wave energy converters. We discussed and corrected two mistakes in the AR wave prediction method of [15], thus the accuracy and robustness of ocean wave prediction using our AR prediction methods had been significantly improved compared to [15].

In the following, we conclude our result in each of the numerical examples that we have investigated. For the ALS method, we used two LTV examples and one nonlinear example to demonstrate and compare our ALS method with the one in [35]:

1. In the first LTV example, Figure 3.4 has shown that Assumption 1 in [35] can be fulfilled if $N_k \geq 30$. Table 3.2 showed that, as $N_k$ increased from 0 to 30, the mean of the estimated noise covariances becomes closer to the true values and the variance of the estimated noise covariances becomes smaller. When $N_k = 30$, the method of [35] is still worse than our ALS algorithm due to the effective $M_e$ in [35] being less than our ALS method.

2. In the second LTV example, Figure 3.9 showed that the Assumption 1 in [35] cannot be fulfilled with a given amount of output measurements. Thus, our ALS method provided better covariance estimates than [35], because the method of [35] is unable to provide the estimates of initial state error covariance $P_{1|0}$. Figure 3.13 showed that the state estimate can be significantly improved, if an inappropriate $P_{1|0}$ is applied.

3. In the third nonlinear example, we showed that the ALS-EKF method is able to estimate noise covariances for a nonlinear system, where the state noise term $w_k$ is not additive to the nonlinear dynamic.

4. In the last nonlinear example, we demonstrated that, if the guesses for the initial state and its error covariances are bad, then the ALS algorithm combined with the EKF will not provide acceptable results. However, our algorithm combines ALS with MHE, which is still able to provide a good estimate of noise covariances.

For the EM method, we used two nonlinear examples to compare to the existing EM method, as well as the ALS methods.

1. The same nonlinear example that was employed by the ALS algorithm has been used to examine the EM algorithm. Results have shown that Algorithm 4.2, “fast” Algorithm 4.2 and the existing EM Algorithm given by [5] could provide accurate
covariance estimation. However, due to the iterative process of the EM algorithm, they will take much longer compared to ALS based Algorithm 4.1, which could provide similar good estimates.

2. The second nonlinear example showed that only Algorithm 4.2 and “fast” Algorithm 4.2 could provide relatively good estimates of covariances, compared to other EM and ALS based covariance estimation algorithms. The first reason is that our EM algorithms use FIE/MHE instead of the EKF, hence is able to add state constraints for better state estimation accuracy. The second reason is that ALS based algorithms estimate covariance by solving one optimization, which could cause a problem for nonlinear systems with bad guesses for noise covariances. In contrast to the ALS method, the EM based algorithm recursively maximizes the likelihood function until convergence, which resulted in a better estimate in this example.

Finally, for ocean wave prediction, both AR prediction models (5.2) and (5.6) provided very similar wave prediction performance based on 10 groups of ocean wave data recorded at different times. However, model (5.2) takes less time than model (5.6). Thus, we selected model (5.2) for further tests and comparisons. We compared prediction performance by varying the prediction horizon from 1 sec to 10 sec using AR model (5.2) with the ALS and EM method, as well as 3 other existing methods. Figure 5.10 showed that our AR prediction model (5.2) with the EM method provided the best wave prediction up to 10 seconds into the future.

6.2 Future Works

Extensions of this thesis could involve finding a way to determine the term \( p(x_{1:M}, y_{1:M}|\mathcal{O}) \) in the EM method, where

\[
p(x_{1:M}, y_{1:M}|\mathcal{O}) \propto p(x_{1:M}|y_{1:M}, \mathcal{O}) \propto p(y_{1:M}|x_{1:M}, \mathcal{O}) p(x_{1:M}, \mathcal{O})
\]

for nonlinear system (2.14), so that the EM method is able to estimate noise covariances for a more general nonlinear system. Because noise terms \( w_k \) and \( v_k \) are not additive to the nonlinear dynamics, evaluating the state and output transition probability density functions \( p(x_{k+1}|x_k) \) and \( p(y_k|x_k) \), respectively, will become difficult, which requires the solution of a functional difference equation, namely the discrete-time analog of the Fokker-Planck equation [38]. Alternatively, \( p(x_{k+1}|x_k) \) and \( p(y_k|x_k) \) can be approximated by first order
Taylor series:
\[
f(x_k, 0) = f(\hat{x}_k, 0) + \left. \frac{\partial f(\cdot)}{\partial x_k} \right|_{x_k = \hat{x}_k, \, w_k = 0} (x_k - \hat{x}_k) + \text{error}_{f_0}, \tag{6.1a}
\]
\[
h(x_k, 0) = h(\hat{x}_k, 0) + \left. \frac{\partial h(\cdot)}{\partial x_k} \right|_{x_k = \hat{x}_k, \, v_k = 0} (x_k - \hat{x}_k) + \text{error}_{h_0}. \tag{6.1b}
\]
Substituting (6.1) to (2.15) gives
\[
f(x_k, w_k) = f(x_k, 0) + G_kw_k + \text{error}_f - \text{error}_{f_0},
\]
\[
h(x_k, v_k) = h(x_k, 0) + H_kv_k + \text{error}_h - \text{error}_{h_0}.
\]
By ignoring all error terms, if \( G_k \) and \( H_k \) are both full column rank matrices, then
\[
f(x_k, w_k) \approx f(x_k, 0) + G_kw_k \implies p(x_{k+1}|x_k) \approx \frac{1}{\sqrt{(2\pi)^q|Q|}} e^{-\frac{1}{2}||w_k||_Q^2},
\]
\[
h(x_k, v_k) \approx h(x_k, 0) + H_kv_k \implies p(y_k|x_k) \approx \frac{1}{\sqrt{(2\pi)^q|R|}} e^{-\frac{1}{2}||v_k||_R^2}.
\]
Another possible extension of the EM method for nonlinear systems is finding a new way to determine the covariance terms \( C[v_k|Y_M, O_{i-1}] \) and \( C[w_k|Y_M, O_{i-1}] \) in the nonlinear EM method, without linearizing the system or any other approximations (i.e. unscented Kalman filter or particle filters), so the accuracy of the EM method can be further improved.

The efficiency of nonlinear noise covariance estimation can be improved by speeding up the MHE based state estimation. A possible approach is given in [43, p. 350], which combines the MHE with a particle filter [31]. Table 6.1 compares advantages and disadvantages between the MHE and particle filter. Table 6.1 shows that both MHE and particle filter do not linearize the nonlinear system, 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 [43, 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.

For the ocean wave prediction, we are already able to involve the noise covariance into the optimization. One could try to involve the size of the AR parameters \( n \) and the cut-off frequency \( f_c \) into the optimization as well. Moreover, instead of predicting ocean waves at one location, Figure 6.1 shows another possible approach introduced by [15], which
Table 6.1 Advantages and disadvantages of MHE and particle filter

<table>
<thead>
<tr>
<th></th>
<th>MHE</th>
<th>Particle Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>advantages</td>
<td>states propagate through original nonlinear system</td>
<td>samples propagate through original nonlinear system</td>
</tr>
<tr>
<td></td>
<td>physical constraints can be included</td>
<td>by using resampling, the sampled probability density can represent any general conditional probability density</td>
</tr>
<tr>
<td></td>
<td>smaller state error norms $|e_k|^M_{k=1}$ than the EKF</td>
<td>state estimation executes quickly for small sample sizes</td>
</tr>
<tr>
<td>disadvantages</td>
<td>state estimation may take a long time</td>
<td>performance of state estimation decreases significantly for higher-order nonlinear systems</td>
</tr>
<tr>
<td></td>
<td>global optimal is not guaranteed</td>
<td>poor robustness for unmodeled disturbance</td>
</tr>
</tbody>
</table>

predicts ocean waves based on reconstruction of the wave field from an array of distant measurements. Obviously, predicting ocean waves at one location is a simple and cheaper forecasting approach, however, the forecasting horizon is limited by signal coherence and the wave measurement has to rely on the motion (hydrodynamics) of the WEC devices. The alternative approach requires lots of instruments for measurements and communications, hence a more complicated system model, but it is possible to measure incoming waves from different directions and distances, hence provide a deterministic approach and more possible accurate prediction for long time horizons.
References


Appendix A

Proof of non-singularity of matrix $\mathcal{H}_{S_k}$ in Theorem 2.4

Lemma A.1. [6, Fact 2.14.9] Let $A$, $C$, $B$ be matrices of dimension $n \times n$, $m \times n$ and $m \times m$, respectively. If matrix $A$ is nonsingular, then

$$\det \left( \begin{bmatrix} A & C^\top \\ C & B \end{bmatrix} \right) = \det(A) \times \det \left( B - CA^{-1}C^\top \right)$$

and if matrix $B$ is nonsingular, then

$$\det \left( \begin{bmatrix} A & C^\top \\ C & B \end{bmatrix} \right) = \det(B) \times \det \left( A - C^\top B^{-1}C \right).$$

We start from deriving the determinate of matrix $\mathcal{H}_{S_1}$. By using Lemma A.1, we have

$$\det(\mathcal{H}_{S_1}) := \det \begin{pmatrix} P_{1|0}^{-1} & 0 & -I_n & 0 & 0 \\ 0 & R^{-1} & 0 & -H_1^\top & 0 \\ -I_n & 0 & 0 & 0 & -I_n \\ 0 & -H_1 & 0 & 0 & -C_1 \\ 0 & 0 & -I_n & -C_1^\top & 0 \end{pmatrix}$$

$$= \det \begin{pmatrix} P_{1|0}^{-1} & 0 \\ 0 & R^{-1} \end{pmatrix} \det \begin{pmatrix} -P_{1|0} & 0 & -I_n \\ 0 & -H_1RH_1^\top & -C_1^\top \\ -I_n & -C_1^\top & 0 \end{pmatrix}$$
Proof of non-singularity of matrix $\mathcal{H}_{S_1}$ in Theorem 2.4

\[
\det(\mathcal{H}_{S_1}) := \det\left(\begin{bmatrix} P_{1|0}^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix}\right) \det(P_{1|0}) \det\left(\begin{bmatrix} -H_1 R H_1^\top - C_1 \\ -C_1^\top - P_{1|0}^{-1} \end{bmatrix}\right)
\]

\[
= \det\left(\begin{bmatrix} P_{1|0}^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix}\right) \det\left(-H_1 R H_1^\top - C_1 P_{1|0} C_1^\top \right) \neq 0,
\]

since $H_1 R H_1^\top \succeq 0$ and $C P_{1|0} C^\top \succ 0$. For any nonzero column vector $z$, we have

\[
z^\top \left(-H_1 R H_1^\top - C_1 P_{1|0} C_1^\top \right) z = -z^\top \left(H_1 R H_1^\top \right) z - z^\top \left(-C_1 P_{1|0} C_1^\top \right) z < 0,
\]

hence $\left(-H_1 R H_1^\top - C_1 P_{1|0} C_1^\top \right) < 0 \Rightarrow \mathcal{H}_{S_1}$ is nonsingular. The determinant of matrix $\mathcal{H}_{S_2}$ is

\[
\det(\mathcal{H}_{S_2}) := \det\left(\begin{bmatrix} \mathcal{H}_{S_2}^{1,1} & \mathcal{H}_{S_2}^{1,2} \\ \mathcal{H}_{S_2}^{2,1} & \mathcal{H}_{S_2}^{2,2} \end{bmatrix}\right)
\]

\[
= \det\left(\begin{bmatrix} P_{1|0}^{-1} & 0 & -I_n & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & R^{-1} & 0 & -H_1^\top & 0 & 0 & 0 & 0 & 0 \\ -I_n & 0 & 0 & 0 & -I_n & 0 & 0 & 0 & 0 \\ 0 & -H_1 & 0 & 0 & -C_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I_n & -C_1^\top & 0 & 0 & 0 & -A_1^\top & 0 \\ 0 & 0 & 0 & 0 & Q^{-1} & 0 & 0 & -G_1^\top & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & R^{-1} & 0 \\ 0 & 0 & 0 & 0 & -A_1 & -G_1 & 0 & 0 & I_n \\ 0 & 0 & 0 & 0 & 0 & -H_2 & 0 & 0 & -C_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_n & -C_2^\top & 0 \end{bmatrix}\right).
\]

Let the inverse of $\mathcal{H}_{S_1}$ be

\[
\mathcal{H}_{S_1}^{-1} := \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} \\ h_{31} & h_{32} & h_{33} & h_{14} & h_{15} \\ h_{41} & h_{42} & h_{43} & h_{44} & h_{45} \\ h_{51} & h_{52} & h_{53} & h_{54} & h_{55} \end{bmatrix},
\]
where
\[
\begin{align*}
    h_{55} &= P_{1|0} - P_{1|0} C_1^\top \left( C_1 P_{1|0} C_1 + H_1 R H_1^\top \right)^{-1} L_1 C_1 P_{1|0}, \\
    &= (I_n - L_1 C_1) P_{1|0} = (I_n - L_1 C_1) P_{1|0} (I_n - L_1 C_1)^\top + L_1 H_1 R H_1^\top L_1^\top \succ 0.
\end{align*}
\]

Because $\mathcal{H}_{51}$ is nonsingular, hence $\mathcal{H}_{S_2}^{1,1}$ is nonsingular. By using Lemma A.1 we have
\[
\det(\mathcal{H}_{S_2}) = \det(\mathcal{H}_{S_2}^{1,1}) \det\left( \begin{bmatrix} -P_{2|1} & 0 & I_n \\ 0 & -H_2 R H_2^\top & -C_2 \\ I_n & -C_2^\top & 0 \end{bmatrix} \right),
\]
where $P_{2|1} := A_1 h_{55} A_1^\top + G_1 Q G_1^\top \succ 0$, hence $\mathcal{H}_{S_2}$ is nonsingular. Therefore, by induction, matrix $\mathcal{H}_{S_k}$ is nonsingular for all $k \leq M$. 
Appendix B

Proof of monotonicity of $\mathcal{L}_\ell(O_i|y_{1:M})$ in the EM method

This theorem is a key contribution of [11] and we are going to briefly summarize the proof here. By the definition of the conditional expectation of a discrete variable

$$J(O_i|O_{i-1}) = \sum_{x_{1:M} \in X} p(x_{1:M}|y_{1:M}, O_{i-1}) \log(p(x_{1:M}|y_{1:M}, O_i)),$$

Gibbs’ inequality [21, p. 291] ensures that

$$\sum_{x_{1:M} \in X} p(x_{1:M}|y_{1:M}, O_{i-1}) \log(p(x_{1:M}|y_{1:M}, O_i)) \leq \sum_{x_{1:M} \in X} p(x_{1:M}|y_{1:M}, O_{i-1}) \log(p(x_{1:M}|y_{1:M}, O_{i-1})), $$

with equality if and only if $O_i = O_{i-1}$.

In order to ensure the monotonic increase of the log-likelihood function $\mathcal{L}_\ell(O|y_{1:M})$, one requires $\mathcal{L}_\ell(O_i|y_{1:M}) - \mathcal{L}_\ell(O_{i-1}|y_{1:M}) \geq 0$, which gives:

$$Q(O_i|O_{i-1}) - Q(O_{i-1}|O_{i-1}) - (J(O_i|O_{i-1}) - J(O_{i-1}|O_{i-1})) \geq 0.$$

Since $J(O_i|O_{i-1}) - J(O_{i-1}|O_{i-1}) \leq 0$, the function $\mathcal{L}_\ell(O|y_{1:M})$ will converge to the maximum if and only if

$$Q(O_i|O_{i-1}) - Q(O_{i-1}|O_{i-1}) \geq 0.$$

This means the parameter set $O_i$ has to be estimated by maximizing the function $Q(O|O_{i-1})$. 
Appendix C

“Lag-n” smoothed state error covariance

Considering following smoothed state error covariance for LTV systems ($k_1 \leq k_2 \leq M$):

$$
P_{k_2,k_1 | M} = \mathbb{E} \left\{ (x_{k_2} - \mathbb{E}\{\hat{x}_{k_2} | y_{1:M}\}) (x_{k_1} - \mathbb{E}\{\hat{x}_{k_1} | y_{1:M}\})^\top \right\},
$$

$$
= \mathbb{E} \left\{ (x_{k_2} - \hat{x}_{k_2|M}) [x_{k_1} - \hat{x}_{k_1} - U_{k_1} (\hat{x}_{k_1+1|M} - \hat{x}_{k_1+1|k_1})]^\top \right\},
$$

$$
= \mathbb{E} \left\{ (x_{k_2} - \hat{x}_{k_2|M}) \begin{bmatrix} (x_{k_1} - \hat{x}_{k_1}) + U_{k_1} (\hat{x}_{k_1+1} - \hat{x}_{k_1+1|k_1}) - U_{k_1} (\hat{x}_{k_1+1} - \hat{x}_{k_1+1|k_1}) \\ e_{k_1} \end{bmatrix}^\top \right\},
$$

$$
= P_{k_2,k_1+1|M} U_{k_1}^\top.
$$

If let $k_1 = M - 1$, $k_2 = M$, then the “lag-one” smoothed state error covariance $P_{M,M-1|M}$ is given by

$$
P_{M,M-1|M} = P_{M,M|M} U_{M-1}^\top = P_{M} U_{M-1}^\top,
$$

$$
= (I_n - L_M C_M) P_{M|M-1} P_{M|M-2}^{-1} A_{M-1} P_{M-1} = (I_n - L_M C_M) A_{M-1} P_{M-1}.
$$

Hence, we have

$$
P_{k\!\!,k-1|M} = P_{k|M} U_{k-1}^\top = \left[ P_k + U_k (P_{k+1|M} - P_{k+1|k}) U_k^\top \right] U_{k-1}^\top,
$$

$$
= \left[ P_k + U_k (P_{k+1,k|M} - A_k P_k) \right] U_{k-1}^\top,
$$

$$
= P_k U_{k-1}^\top + U_k (P_{k+1,k|M} - A_k P_k) U_{k-1}^\top.
$$
Appendix D

Derivation of (3.5) and (3.6)

D.1 Derivation of (3.5)

Recall (3.2a) and (3.2b) gives

\[ \varepsilon_{k+1} = \bar{A}_k \varepsilon_k + \bar{G}_k \bar{w}_k, \quad k = 1, \ldots, M_e, \]

\[ z_k = C_k \varepsilon_k + H_k v_k, \]

hence,

\[
\begin{bmatrix}
I_n & 0 & \cdots & \cdots & \cdots & 0 \\
-\bar{A}_2 & I_n & 0 & : & : & : \\
0 & -\bar{A}_3 & I_n & 0 & : & : \\
: & 0 & \cdots & \cdots & \cdots & 0 \\
: & : & \cdots & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & 0 \\
\end{bmatrix}
\begin{bmatrix}
\varepsilon_2 \\
\varepsilon_3 \\
\vdots \\
\varepsilon_{M_e+1} \\
\end{bmatrix}
= 
\begin{bmatrix}
\bar{A}_1 \\
\bar{A}_1 \\
\vdots \\
\bar{A}_{M_e} \\
\end{bmatrix}
\begin{bmatrix}
\varepsilon_1 \\
\varepsilon_1 \\
\vdots \\
\varepsilon_1 \\
\end{bmatrix}
+ 
\begin{bmatrix}
\bar{G}_1 & 0 & \cdots & \cdots & \cdots & 0 \\
0 & \bar{G}_2 & 0 & : & : & : \\
: & 0 & \cdots & \cdots & \cdots & : \\
: & : & \cdots & \cdots & \cdots & : \\
: & : & \cdots & \cdots & \cdots & : \\
0 & \cdots & \cdots & \cdots & \cdots & 0 \\
\end{bmatrix}
\begin{bmatrix}
\bar{w}_1 \\
\bar{w}_2 \\
\vdots \\
\bar{w}_{M_e} \\
\end{bmatrix}
\]

and \( \varepsilon = \bar{A}^{-1} (\hat{\varepsilon} \varepsilon_1 + \tilde{G} \bar{w}) \). Finally,

\[
\begin{bmatrix}
z_2 \\
z_3 \\
\vdots \\
z_{M_e+1} \\
\end{bmatrix}
= 
\begin{bmatrix}
C_2 & 0 & \cdots & 0 \\
0 & C_3 & \cdots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & C_{M_e+1} \\
\end{bmatrix}
\begin{bmatrix}
\varepsilon \\
\hat{\varepsilon} \\
\vdots \\
\hat{\varepsilon} \\
\end{bmatrix}
+ 
\begin{bmatrix}
H_1 & 0 & \cdots & 0 \\
0 & H_2 & \cdots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & H_{M_e} \\
\end{bmatrix}
\begin{bmatrix}
\tilde{v}_2 \\
\tilde{v}_3 \\
\vdots \\
\tilde{v}_{M_e+1} \\
\end{bmatrix}
\]
and
\[ z = \tilde{V}(\tilde{\varepsilon}_1 + \tilde{G}\tilde{w}) + \tilde{F}\tilde{v}. \]

**D.2 Derivation of (3.6)**

Expanding \( \varepsilon = \tilde{A}^{-1}(\tilde{\varepsilon}_1 + \tilde{G}\tilde{w}) \) gives
\[
\varepsilon = \tilde{A}^{-1}\begin{pmatrix}
\begin{bmatrix}
G_1 & 0 & \ldots & 0 \\
0 & G_2 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & \ldots & G_{M_e}
\end{bmatrix} & \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{M_e} \end{bmatrix} \\
\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{M_e} \end{bmatrix}
\end{pmatrix} + \begin{pmatrix}
\begin{bmatrix}
A_1 L^s_1 H_1 & 0 & \ldots & 0 \\
0 & A_2 L^s_2 H_2 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & \ldots & A_{M_e} L^s_{M_e} H_{M_e}
\end{bmatrix} & \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{M_e} \end{bmatrix} \\
\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{M_e} \end{bmatrix}
\end{pmatrix},
\]

hence we have
\[
z = \tilde{V}(\tilde{\varepsilon}_1 + \tilde{G}\tilde{w} + \tilde{\mathcal{L}}\tilde{v}) + \tilde{F}\tilde{v} = \tilde{F}_x\varepsilon_1 + \tilde{B}w + \tilde{D}v + \tilde{F}\tilde{v},
\]

which for \( i = 0, \ldots, N_z - 1 \), gives
\[
z_{i+2} = \tilde{M}\tilde{S}_i z = \tilde{M}\tilde{S}_i \begin{pmatrix}
\begin{bmatrix}
\tilde{F}_x\varepsilon_1 + \tilde{B}\tilde{g}_i \\
\vdots \\
\tilde{w}_{i+1}
\end{bmatrix} + \tilde{D}\tilde{U}_i \\
\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{i+1} \end{bmatrix}
\end{pmatrix} + H_{i+2}v_{i+2}
\]

and
\[
\tilde{S}_i z = \begin{bmatrix}
z_{i+2} \\
z_{i+3} \\
\vdots \\
z_{i+N+1}
\end{bmatrix} = \tilde{S}_i \begin{pmatrix}
\begin{bmatrix}
\tilde{F}_x\varepsilon_1 + \tilde{B}\tilde{g}_{i+N-1} \\
\vdots \\
\tilde{w}_{i+N}
\end{bmatrix} + \tilde{D}\tilde{U}_{i+N-1} \\
\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{i+N} \end{bmatrix}
\end{pmatrix} + \begin{pmatrix}
\begin{bmatrix}
H_{i+2} & 0 & \ldots & 0 \\
0 & H_{i+3} & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & \ldots & H_{i+N+1}
\end{bmatrix} & \begin{bmatrix} v_{i+2} \\ v_{i+3} \\ \vdots \\ v_{i+N+1} \end{bmatrix}
\end{pmatrix}.
Hence, for $i = 0$, by removing all rows with pure zero entries,

\[
\mathcal{R}_0 = \mathbb{E} \begin{bmatrix} z_2 \\ z_3 \\ \vdots \\ z_{N+1} \end{bmatrix} = \tilde{s}_0 \left\{ \tilde{F}_s P_1 \tilde{F}_s^T + \tilde{B} \tilde{G}_0 (I_1 \otimes Q) \tilde{J}_0 \tilde{B}^T + \tilde{D} \tilde{U}_0 (I_1 \otimes R) \tilde{U}_0^T \tilde{D}^T \right\} \tilde{s}_0^T \tilde{M}^T \\
+ \tilde{s}_0 \tilde{D} \tilde{U}_1 \begin{bmatrix} 0 \\ R \end{bmatrix} H_2^T + \begin{bmatrix} H_2 R H_2^T \\ 0 \end{bmatrix},
\]

and for $i = 1$, by removing all rows with pure zero entries,

\[
\mathcal{R}_1 = \mathbb{E} \begin{bmatrix} z_3 \\ z_4 \\ \vdots \\ z_{N+2} \end{bmatrix} = \tilde{s}_1 \left\{ \tilde{F}_s P_1 \tilde{F}_s^T + \tilde{B} \tilde{G}_1 (I_2 \otimes Q) \tilde{J}_0 \tilde{B}^T + \tilde{D} \tilde{U}_1 (I_2 \otimes R) \tilde{U}_1^T \tilde{D}^T \right\} \tilde{s}_1^T \tilde{M}^T \\
+ \begin{bmatrix} H_3 R H_3^T \\ \tilde{P}_1 \tilde{D} \tilde{O}_1 \tilde{R} H_3^T \end{bmatrix}.
\]

Finally, for $i = N_z - 1$, by removing all rows with pure zero entries,

\[
\mathcal{R}_{N_z-1} = \mathbb{E} \begin{bmatrix} z_{N_z+1} \\ z_{N_z+2} \\ \vdots \\ z_{M_z+1} \end{bmatrix} = \tilde{s}_{N_z-1} \left\{ \tilde{F}_s P_1 \tilde{F}_s^T + \tilde{B} \tilde{G}_N (I_N \otimes Q) \tilde{J}_{N_z-1} \tilde{B}^T \\
+ \tilde{D} \tilde{U}_{N_z-1} (I_N \otimes R) \tilde{U}_{N_z-1}^T \tilde{D}^T \right\} \tilde{s}_{N_z-1}^T \tilde{M}^T + \begin{bmatrix} H_{N_z-1} R H_{N_z-1}^T \\ \tilde{P}_{N_z-1} \tilde{D} \tilde{O}_{N_z-1} \tilde{R} H_{N_z-1}^T \end{bmatrix}.
\]

Therefore,

\[
\mathcal{R}(P_1, Q, R) = \Gamma (I_{N_z} \otimes P_{1|0}) \tilde{\Gamma}^T + \Omega (I_{N_d} \otimes Q) \tilde{\Omega}^T + \Phi (I_{N_d} \otimes R) \tilde{\Phi}^T + \Psi (I_{N_z} \otimes R) \tilde{\Psi}^T.
\]
Appendix E

Proof of Proposition 5.1

Substituting (2.4b) into (2.4d) yields the Riccati equation

\[ P_{k+1|k} = A_k (I_n - L_k C_k) P_{k|k-1} A_k^T + G_k Q G_k^T \]
\[ = A_k P_{k|k-1} A_k^T + G_k Q G_k^T - A_k L_k C_k P_{k|k-1} A_k^T \]
\[ = A_k P_{k|k-1} A_k^T + G_k Q G_k^T - A_k P_{k|k-1} C_k^T \left( C_k P_{k|k-1} C_k^T + H_k R H_k^T \right)^{-1} C_k P_{k|k-1} A_k^T \]
\[ = A_k P_{k|k-1} A_k^T + G_k Q G_k^T - L_{k+1|k} \left( C_k P_{k|k-1} C_k^T + H_k R H_k^T \right) L_{k+1|k}^T, \]

(E.1)

where

\[ L_{k+1|k} = A_k L_k. \]

Hence, we have

\[ \hat{x}_{k+1|k} = A_k \hat{x}_{k|k-1} + L_{k+1|k} \left( y_k - C_k \hat{x}_{k|k-1} \right). \]

Since \( A = I_n, R = 1, \lambda = R \) and \( Q = 0 \), we have

\[ L_{k+1|k} = L_k = P_{k|k-1} C_k^T \left( C_k P_{k|k-1} C_k^T + \lambda \right)^{-1}. \]

(E.2)

Substituting (E.2) into (E.1) gives

\[ P_{k+1|k} = P_{k|k-1} - P_{k|k-1} C_k^T \left( C_k P_{k|k-1} C_k^T + \lambda \right)^{-1} C_k P_{k|k-1} = P_{k|k-1} - L_{k+1|k} C_k P_{k|k-1}. \]

(E.3)
Hence, we have

\[ L_{k+1|k} = P_{k|k-1}C_k^\top \lambda^{-1}\left(C_k P_{k|k-1} C_k^\top \lambda^{-1} + 1\right)^{-1}, \]

\[ L_{k+1|k} \left(C_k P_{k|k-1} C_k^\top \lambda^{-1} + 1\right) = P_{k|k-1} C_k^\top \lambda^{-1}, \]

\[ L_{k+1|k} = (P_{k|k-1} - L_{k+1|k} C_k P_{k|k-1}) C_k^\top \lambda^{-1} = P_{k+1|k} C_k^\top \lambda^{-1}. \]

Because \( \lambda \) is a scalar and

\[ L_{k+1|k} = P_{k|k-1} C_k^\top \left(C_k P_{k|k-1} C_k^\top + \lambda\right)^{-1} = P_{k+1|k} C_k^\top \lambda^{-1}, \]

we have

\[ P_{k+1|k} = \lambda P_{k|k-1} \left(C_k P_{k|k-1} C_k^\top + \lambda\right)^{-1} \]

and

\[ \hat{x}_k = \hat{x}_{k-1} + L_k (y_k - C_k \hat{x}_{k-1}). \]