Noise Covariance Estimation for an Air Separation Plant

Travis J. Arnold

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

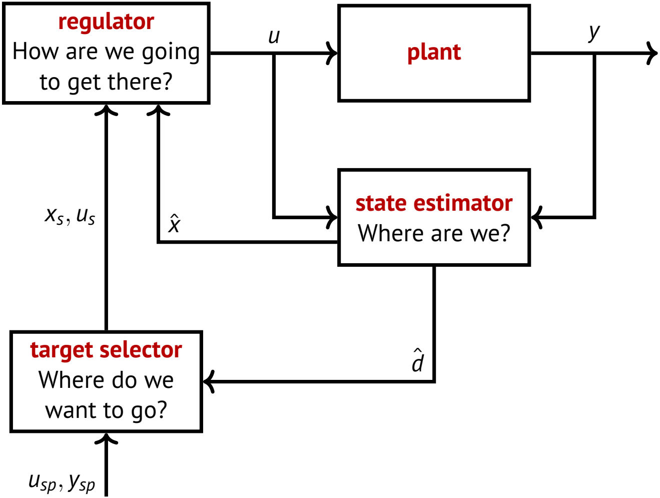
Department of Chemical and Biological Engineering

Systems Seminar

January 20, 2017

MPC Controller

MPC Controller



Motivation

Motivation

- Noise and unmodeled disturbances are ubiquitous in practice in systems and controls engineering.

Motivation

- Noise and unmodeled disturbances are ubiquitous in practice in systems and controls engineering.
- However, dynamic system modeling often focuses primarily on identifying a deterministic (input-output) model.

Motivation

- Noise and unmodeled disturbances are ubiquitous in practice in systems and controls engineering.
- However, dynamic system modeling often focuses primarily on identifying a deterministic (input-output) model.

Quantifying the noises that affect the system is necessary in order to build an optimal state estimator.

Two common assumptions

Two common assumptions

1. There are two distinct types of noise: process noise and measurement noise.

Two common assumptions

1. There are two distinct types of noise: process noise and measurement noise.
2. Noises can be modeled as zero-mean Gaussian random variables.

Two common assumptions

1. There are two distinct types of noise: process noise and measurement noise.
2. Noises can be modeled as zero-mean Gaussian random variables.

Under these assumptions, the task of noise model identification reduces to identifying the process and measurement noise covariance matrices.

Linear dynamic model

Linear dynamic model

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{y}_k &= C\mathbf{x}_k + \mathbf{v}_k \\ \mathbf{w}_k &\overset{iid}{\sim} \mathcal{N}(0, Q) \\ \mathbf{v}_k &\overset{iid}{\sim} \mathcal{N}(0, R) \end{aligned}$$

State:	$\mathbf{x} \in \mathbb{R}^n$
Input:	$\mathbf{u} \in \mathbb{R}^m$
Output:	$\mathbf{y} \in \mathbb{R}^p$
Process noise:	$\mathbf{w} \in \mathbb{R}^n$
Measurement noise:	$\mathbf{v} \in \mathbb{R}^p$

Linear dynamic model

$$x_{k+1} = Ax_k + Bu_k + w_k$$
$$y_k = Cx_k + v_k$$
$$w_k \overset{iid}{\sim} \mathcal{N}(0, Q)$$
$$v_k \overset{iid}{\sim} \mathcal{N}(0, R)$$

State: $x \in \mathbb{R}^n$

Input: $u \in \mathbb{R}^m$

Output: $y \in \mathbb{R}^p$

Process noise: $w \in \mathbb{R}^n$

Measurement noise: $v \in \mathbb{R}^p$

Our goal:

Linear dynamic model

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k$$
$$\mathbf{y}_k = C\mathbf{x}_k + \mathbf{v}_k$$
$$\mathbf{w}_k \overset{iid}{\sim} \mathcal{N}(0, Q)$$
$$\mathbf{v}_k \overset{iid}{\sim} \mathcal{N}(0, R)$$

State: $\mathbf{x} \in \mathbb{R}^n$

Input: $\mathbf{u} \in \mathbb{R}^m$

Output: $\mathbf{y} \in \mathbb{R}^p$

Process noise: $\mathbf{w} \in \mathbb{R}^n$

Measurement noise: $\mathbf{v} \in \mathbb{R}^p$

Our goal: Given a set of of input-output data $u_{0:N-1}$ and $y_{0:N}$, estimate (Q, R).

Innovations form

Innovations form

- Estimate the states using a stable (but not necessarily optimal) linear state estimator with gain L, giving estimates:

Innovations form

- Estimate the states using a stable (but not necessarily optimal) linear state estimator with gain L, giving estimates:

$$\begin{aligned}\widehat{x}_{k+1|k} &= A\widehat{x}_{k|k} + Bu_k \\ \widehat{x}_{k|k} &= \widehat{x}_{k|k-1} + L(y_k - C\widehat{x}_{k|k-1})\end{aligned}$$

Innovations form

- Estimate the states using a stable (but not necessarily optimal) linear state estimator with gain L, giving estimates:

$$\begin{aligned} \widehat{x}_{k+1|k} &= A\widehat{x}_{k|k} + Bu_k \\ \widehat{x}_{k|k} &= \widehat{x}_{k|k-1} + L(y_k - C\widehat{x}_{k|k-1}) \end{aligned}$$

- Define the state estimate error $\varepsilon_k := x_k - \widehat{x}_{k|k-1}$ and the L-innovations $\mathscr{Y}_k = y_k - C\widehat{x}_{k|k-1}$. These evolve according to

Innovations form

- Estimate the states using a stable (but not necessarily optimal) linear state estimator with gain L, giving estimates:

$$\widehat{x}_{k+1|k} = A\widehat{x}_{k|k} + Bu_k$$
$$\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + L(y_k - C\widehat{x}_{k|k-1})$$

- Define the state estimate error $\varepsilon_k := x_k - \widehat{x}_{k|k-1}$ and the L-innovations $\mathscr{Y}_k = y_k - C\widehat{x}_{k|k-1}$. These evolve according to

$$\varepsilon_{k+1} = \bar{A}\varepsilon_k + \bar{G}\bar{w}_k$$
$$\mathscr{Y}_k = C\varepsilon_k + v_k$$

  - $\bar{A} = A - ALC, \qquad \bar{G} = [I \quad -AL], \qquad \bar{w}_k = \begin{bmatrix} w_k \\ v_k \end{bmatrix}$

Innovations form

- Estimate the states using a stable (but not necessarily optimal) linear state estimator with gain L, giving estimates:

$$\begin{aligned} \widehat{x}_{k+1|k} &= A\widehat{x}_{k|k} + Bu_k \\ \widehat{x}_{k|k} &= \widehat{x}_{k|k-1} + L(y_k - C\widehat{x}_{k|k-1}) \end{aligned}$$

- Define the state estimate error $\varepsilon_k := x_k - \widehat{x}_{k|k-1}$ and the L-innovations $\mathscr{Y}_k = y_k - C\widehat{x}_{k|k-1}$. These evolve according to

$$\begin{aligned} \varepsilon_{k+1} &= \bar{A}\varepsilon_k + \bar{G}\bar{w}_k \\ \mathscr{Y}_k &= C\varepsilon_k + v_k \end{aligned}$$

  - $\bar{A} = A - ALC, \qquad \bar{G} = \begin{bmatrix} I & -AL \end{bmatrix}, \qquad \bar{w}_k = \begin{bmatrix} w_k \\ v_k \end{bmatrix}$

- $E[\mathscr{Y}_{k+j}\mathscr{Y}_k']$ is the autocovariance of the innovations at lag j.

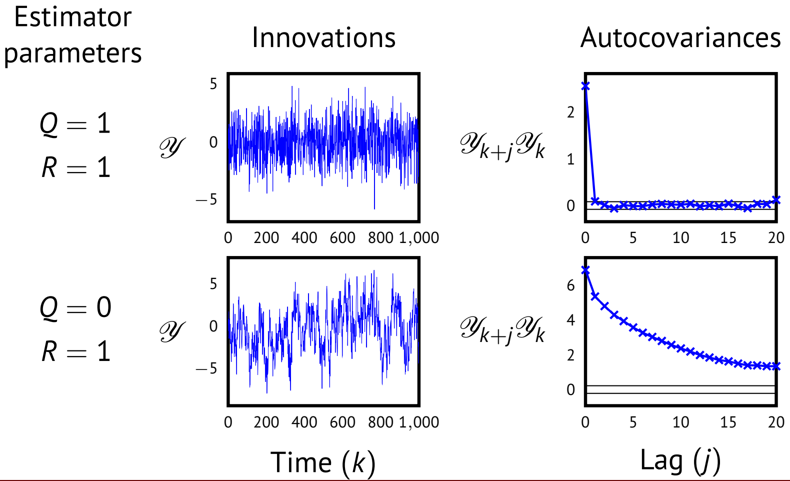
Optimal estimators produce white innovations

Optimal estimators produce white innovations

$$A = 0.9, \quad C = 1, \quad Q = 1, \quad R = 1$$

Optimal estimators produce white innovations

$$A = 0.9, \quad C = 1, \quad Q = 1, \quad R = 1$$



The autocovariance least squares (ALS) method

The autocovariance least squares (ALS) method

- Introduced by Odelson et al. (2006), improved by Rajamani and Rawlings (2009) and Zagrobelny and Rawlings (2015).

The autocovariance least squares (ALS) method

- Introduced by Odelson et al. (2006), improved by Rajamani and Rawlings (2009) and Zagrobelny and Rawlings (2015).
- Takes advantage of the linear relationship between the autocovariance matrix $\mathscr{R}_1(M)$ and (Q, R):

The autocovariance least squares (ALS) method

- Introduced by Odelson et al. (2006), improved by Rajamani and Rawlings (2009) and Zagrobelny and Rawlings (2015).
- Takes advantage of the linear relationship between the autocovariance matrix $\mathscr{R}_1(M)$ and (Q, R):

$$\mathscr{R}_1(M) := E \begin{bmatrix} \mathscr{Y}_k \mathscr{Y}_k' \\ \cdots \\ \mathscr{Y}_{k+M-1} \mathscr{Y}_k' \end{bmatrix} = \mathscr{A} \begin{bmatrix} \text{vec}\, Q \\ \text{vec}\, R \end{bmatrix}$$

  - $\mathscr{A}$ is a constant matrix depending on A, C, and L.

The autocovariance least squares (ALS) method

- Introduced by Odelson et al. (2006), improved by Rajamani and Rawlings (2009) and Zagrobelny and Rawlings (2015).
- Takes advantage of the linear relationship between the autocovariance matrix $\mathscr{R}_1(M)$ and (Q, R):

$$\mathscr{R}_1(M) := E\begin{bmatrix} \mathscr{Y}_k \mathscr{Y}_k' \\ \cdots \\ \mathscr{Y}_{k+M-1} \mathscr{Y}_k' \end{bmatrix} = \mathscr{A}\begin{bmatrix} \operatorname{vec} Q \\ \operatorname{vec} R \end{bmatrix}$$

  - $\mathscr{A}$ is a constant matrix depending on A, C, and L.
- We define the ALS problem as

$$\min_{Q,R} \left\| \mathscr{A}\begin{bmatrix} \operatorname{vec} Q \\ \operatorname{vec} R \end{bmatrix} - \widehat{\mathscr{R}}_1(M) \right\|^2$$

  - $\widehat{\mathscr{R}}_1(M)$ is estimated from the data.

Maximum likelihood estimation

Maximum likelihood estimation

- If we model the initial state as a Gaussian random variable $x_0 \sim \mathcal{N}(\mu, \Sigma)$, we can write down the probability density of the outputs, $p(y_{0:N}; \mu, \Sigma, Q, R)$.

Maximum likelihood estimation

- If we model the initial state as a Gaussian random variable $x_0 \sim \mathcal{N}(\mu, \Sigma)$, we can write down the probability density of the outputs, $p(y_{0:N}; \mu, \Sigma, Q, R)$.
  - This notation is used to indicate that (μ, Σ, Q, R) parameterize the density of $y_{0:N}$.

Maximum likelihood estimation

- If we model the initial state as a Gaussian random variable $x_0 \sim \mathcal{N}(\mu, \Sigma)$, we can write down the probability density of the outputs, $p(y_{0:N}; \mu, \Sigma, Q, R)$.
  - This notation is used to indicate that (μ, Σ, Q, R) parameterize the density of $y_{0:N}$.
- The maximum likelihood problem is

$$\max_{\mu, \Sigma, Q, R} p(y_{0:N}; \mu, \Sigma, Q, R)$$

Expectation maximization (EM)

Expectation maximization (EM)

- Iterative scheme for solving maximum likelihood problems.

Expectation maximization (EM)

- Iterative scheme for solving maximum likelihood problems.
- Particularly useful in situations where there are "hidden" states (these are the x's in our case).

Expectation maximization (EM)

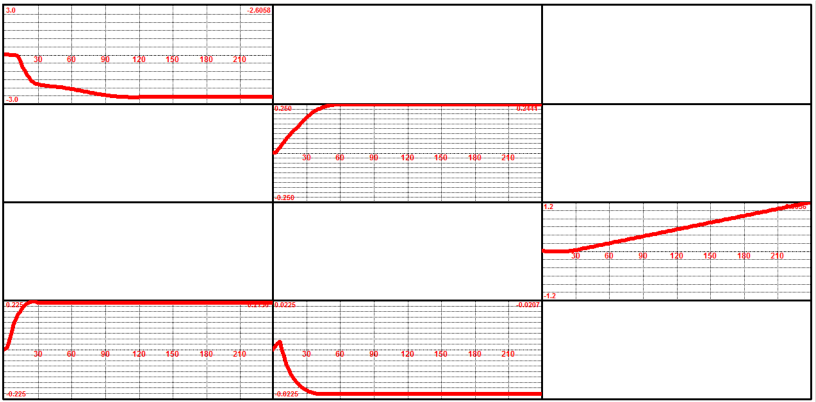
- Iterative scheme for solving maximum likelihood problems.
- Particularly useful in situations where there are "hidden" states (these are the x's in our case).
- First described by Dempster et al. (1977).

Expectation maximization (EM)

- Iterative scheme for solving maximum likelihood problems.
- Particularly useful in situations where there are "hidden" states (these are the x's in our case).
- First described by Dempster et al. (1977).
- Shumway and Stoffer (1982) were the first to use an EM scheme to estimate Q and R. Researchers have remained interested: see Li and Badgwell (2014) for a recent example.

Case study: air separation plant

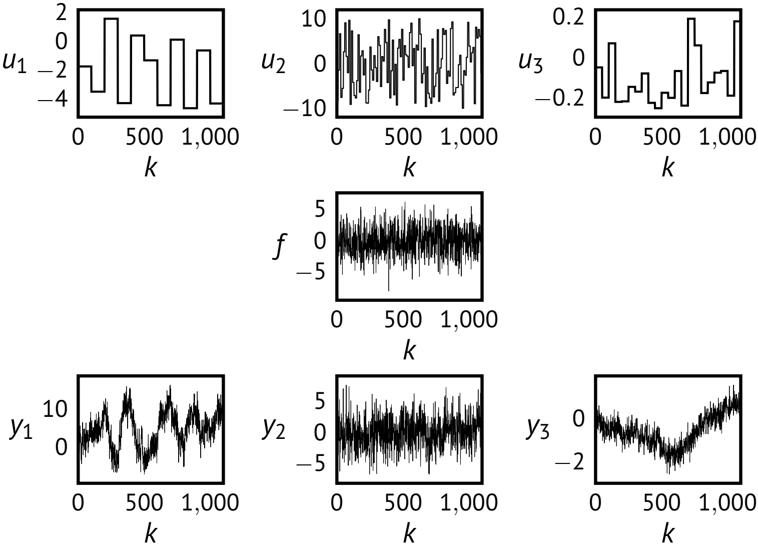
Case study: air separation plant



3 inputs, 1 feedforward, 3 outputs. State space realization ($n = 7$) generated with Matlab's system identification toolbox.

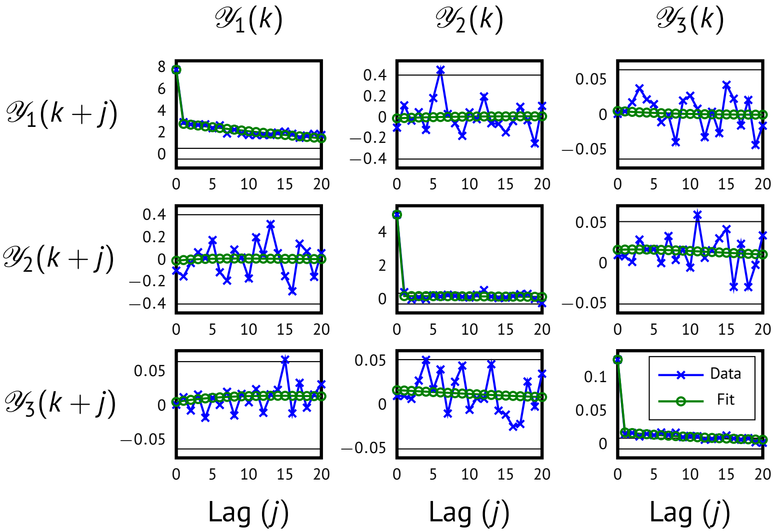
Simulated data plots

Simulated data plots



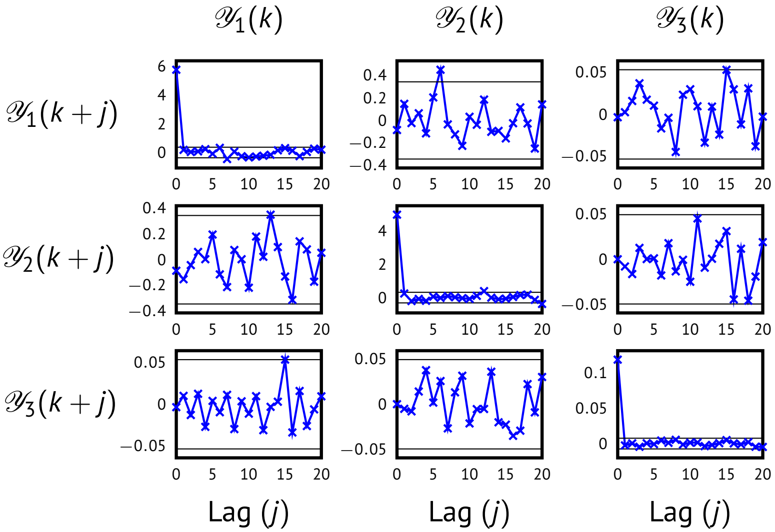
ALS results—fit

ALS results—fit



ALS results—new estimator performance

ALS results—new estimator performance



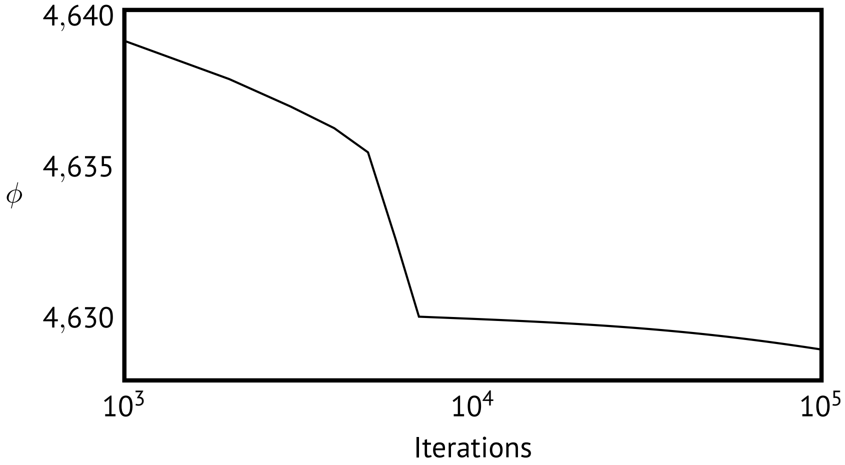
MLE results—objective function

MLE results—objective function

$\phi(\mu, \Sigma, Q, R) = -2 \ln p(y_{0:N}; \mu, \Sigma, Q, R)$ without the constant term.

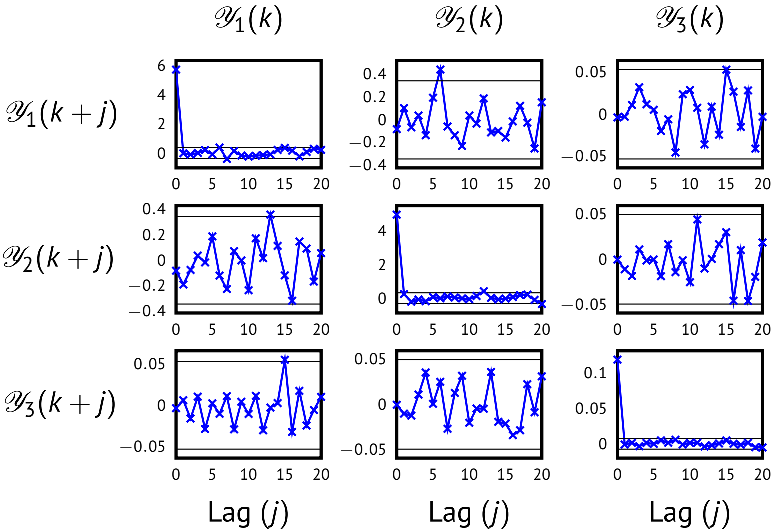
MLE results—objective function

$\phi(\mu, \Sigma, Q, R) = -2 \ln p(y_{0:N}; \mu, \Sigma, Q, R)$ without the constant term.



MLE results—estimator performance

MLE results—estimator performance



Comparison of ALS and MLE/EM

Comparison of ALS and MLE/EM

- ALS is a significantly faster algorithm than EM.

Comparison of ALS and MLE/EM

- ALS is a significantly faster algorithm than EM.
  - EM does not necessarily need to run to full optimality in order to give estimates of Q and R that lead to good estimator.

Comparison of ALS and MLE/EM

- ALS is a significantly faster algorithm than EM.
  - EM does not necessarily need to run to full optimality in order to give estimates of Q and R that lead to good estimator.
- $\phi(\mu, \Sigma, Q, R)$ is non-increasing with each step of EM, but this does not preclude the possibility of convergence to some stationary point other than the global minimum.

Comparison of ALS and MLE/EM

- ALS is a significantly faster algorithm than EM.
  - EM does not necessarily need to run to full optimality in order to give estimates of Q and R that lead to good estimator.
- $\phi(\mu, \Sigma, Q, R)$ is non-increasing with each step of EM, but this does not preclude the possibility of convergence to some stationary point other than the global minimum.
- It can be difficult to know when EM has converged because $\phi(\mu, \Sigma, Q, R)$ may be intractable for large data sets.

Comparison of ALS and MLE/EM

- ALS is a significantly faster algorithm than EM.
  - EM does not necessarily need to run to full optimality in order to give estimates of Q and R that lead to good estimator.
- $\phi(\mu, \Sigma, Q, R)$ is non-increasing with each step of EM, but this does not preclude the possibility of convergence to some stationary point other than the global minimum.
- It can be difficult to know when EM has converged because $\phi(\mu, \Sigma, Q, R)$ may be intractable for large data sets.
- Both algorithms can be modified to encourage the estimate of Q to be low rank:
  - ALS: Rajamani and Rawlings (2009)
  - EM: Li and Badgwell (2014)

Acknowledgments

- The authors gratefully acknowledge the financial support of the industrial members of the Texas-Wisconsin-California Control Consortium, and NSF through grant #CTS-1159088.
- Thanks to Dr. Gangshi Hu from Praxair for providing an industrial model to use.

Further reading I

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. Ser. B*, 39:1–38, 1977.

W. Li and T. A. Badgwell. Structured covariance estimation for state prediction. In *53rd IEEE Conference on Decision and Control*, pages 296–303, Los Angeles, CA, December 15–17 2014.

B. J. Odelson, M. R. Rajamani, and J. B. Rawlings. A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, February 2006.

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

Further reading II

R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *J. Time Series Anal.*, 3:253–264, 1982.

M. A. Zagrobelny and J. B. Rawlings. Practical improvements to autocovariance least-squares. *AIChE J.*, 61:1840–1855, 2015.

Questions?

MLE—details

$$y_{0:N} \sim \mathcal{N}(m, P)$$
$$m = \mathbb{A}_0 \mu + \mathbb{B} U_{0:N-1}$$
$$P = \mathbb{A}_0 \Sigma \mathbb{A}_0' + \mathbb{A}(I_M \otimes Q)\mathbb{A}' + (I_{M+1} \otimes R)$$

$$U_{0:N-1} = [u_0 \cdots u_{N-1}]$$

$$\mathbb{A}_0 = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} \qquad \mathbb{A} = \begin{bmatrix} 0 & & & \\ C & & & \\ CA & C & & \\ \vdots & \vdots & \ddots & \\ CA^{N-1} & CA^{N-2} & \cdots & C \end{bmatrix}$$

$$\mathbb{B} = \begin{bmatrix} 0 & & & \\ CB & & & \\ CAB & CB & & \\ \vdots & \vdots & \ddots & \\ CA^{N-1}B & CA^{N-2}B & \cdots & CB \end{bmatrix}$$

EM—details

- E-step:

$$G(\mu, \Sigma, Q, R) = E_{x_{0:N}|y_{0:N}}[\ln p(x_{0:N}, y_{0:N}; \mu, \Sigma, Q, R)]$$

  Take the conditional expectation of $\ln p(x_{0:N}, y_{0:N}; \mu, \Sigma, Q, R)$ given $y_{0:N}$ where $(\widehat{\mu}_n, \widehat{\Sigma}_n, \widehat{Q}_n, \widehat{R}_n)$ determine the conditional density of $x_{0:N}$. This requires that we run the Kalman smoother.

- M-step:

$$(\widehat{\mu}_{n+1}, \widehat{\Sigma}_{n+1}, \widehat{Q}_{n+1}, \widehat{R}_{n+1}) = \arg \min_{\mu, \Sigma, Q, R} G(\mu, \Sigma, Q, R)$$