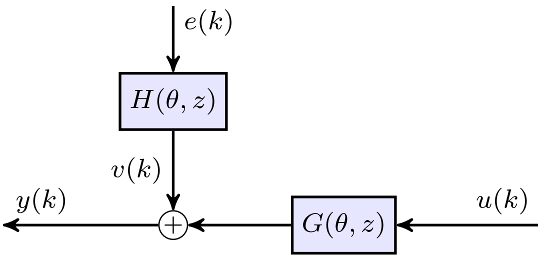
System Identification

Lecture 11: Parametrised transfer function models

Roy Smith

Transfer function models



Transfer function models

Plant model form:

$$G(z) = \frac{b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}.$$

Input output relationship (noise-free case):

$$y(k) \;=\; G(z)u(k) \;=\; -a_1 y(k-1) - \cdots - a_n y(k-n) + b_1 u(k-1) + \cdots + b_m u(k-m)$$

Prediction error & data-fitting

Prediction error formulation

When is PE equivalent to fitting the data?

Transfer function models

Transfer function parametrisation

Plant model form:

$$G(\theta, z) = \frac{b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}, \qquad \theta = \begin{bmatrix} a_1 & \cdots & a_n & b_1 & \cdots & b_m \end{bmatrix}^T.$$

Is the estimation linear or non-linear in θ?

Transfer function models

Is the prediction error-based estimation linear or non-linear in θ?

Prediction error based identification

Prediction error

The one-step ahead predictor is parametrised by θ,

$$\hat{y}(k|\theta, Z_K) = H_{\mathsf{inv}}(\theta, z)G(\theta, z)u(k) + (1 - H_{\mathsf{inv}}(\theta, z))y(k)$$

Parametrised prediction error,

$$\epsilon(k, \theta) = y(k) - \hat{y}(k, \theta).$$

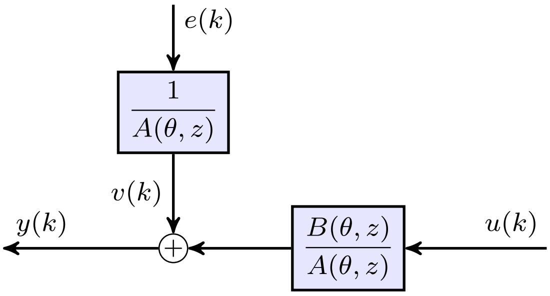
Prediction error-based identification

Typical cost function: $J(\theta, Z_K) = \|\epsilon(k, \theta)\|_2.$

Pick θ to minimise the error.

Optimisation formulation: $\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\ J(\theta, Z_K).$

Prediction error methods: ARX models (Equation error model)



$$G(\theta, z) = \frac{B(\theta, z)}{A(\theta, z)},$$

$$H(\theta, z) = \frac{1}{A(\theta, z)},$$

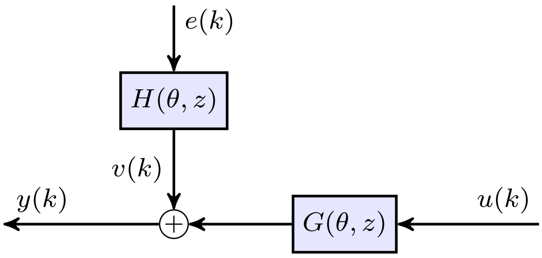
$$\begin{aligned}\hat{y}(k|\theta) &= H_{\mathsf{inv}}(\theta, z)G(\theta, z)u(k) + (1 - H_{\mathsf{inv}}(\theta, z))y(k) \\ &= B(z)u(k) + (1 - A(z))y(k) \\ &= \theta^T \phi(k) = \phi^T(k)\theta.\end{aligned}$$

So,

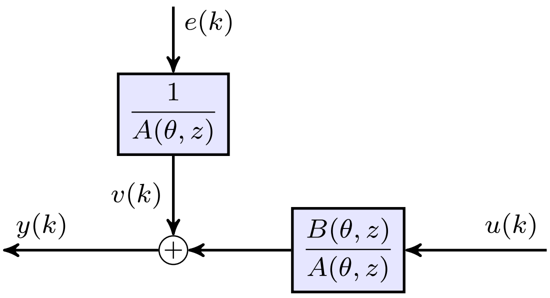
$Y - \Phi\theta = \epsilon \quad \longleftarrow$ vector of prediction errors

Least squares regression approach minimises the prediction errors.

Model structures



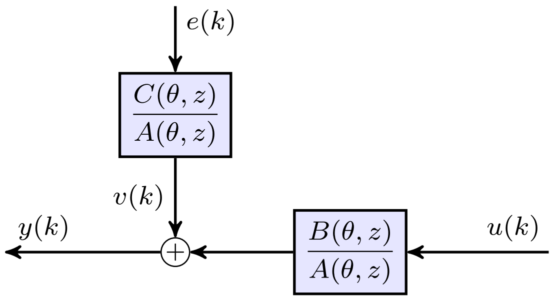
ARX model structure (equation error)



$$G(\theta, z) = \frac{B(\theta, z)}{A(\theta, z)},$$

$$H(\theta, z) = \frac{1}{A(\theta, z)},$$

ARMAX model structure



$$G(\theta, z) = \frac{B(\theta, z)}{A(\theta, z)},$$

$$H(\theta, z) = \frac{C(\theta, z)}{A(\theta, z)},$$

with $A(z)$, $C(z)$ monic.

Prediction error structure

$$\hat{y}(k|\theta) = \frac{B(z)}{C(z)} u(k) + \left(1 - \frac{A(z)}{C(z)}\right) y(k)$$

$$C(z)\hat{y}(k|\theta) = B(z)u(k) + (C(z) - A(z))\, y(k)$$

$$\hat{y}(k|\theta) = B(z)u(k) + (1 - A(z))y(k) + (C(z) - 1)\ \underbrace{(y(k) - \hat{y}(k|\theta))}_{\epsilon(k)}$$

Nonlinear regression

One-step ahead ARMAX predictor

$$\begin{aligned}\hat{y}(k|\theta) &= B(z)u(k) + (1 - A(z))y(k) + (C(z) - 1)\epsilon(k) \\ &= \begin{bmatrix} b_1 & \ldots & a_1 & \ldots & c_1 & \ldots \end{bmatrix} \\ &\qquad\qquad \begin{bmatrix} u(k-1) & \ldots & -y(k-1) & \ldots & \epsilon(k-1) & \ldots \end{bmatrix}^T \\ &= \varphi^T(\theta, k)\theta.\end{aligned}$$

This is not linear in θ.

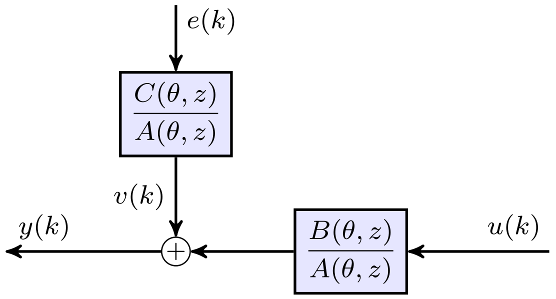
Optimisation-based algorithm

$$\begin{aligned} \underset{\theta,\epsilon}{\text{minimise}} \quad & \|\epsilon\|_2 \\ \text{subject to} \quad & Y = \Phi(\epsilon)\theta + \epsilon \end{aligned}$$

(nonlinear equality constraint)

ARMAX example

ARMAX model structure



$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$$
$$B(z) = b_1 z^{-1} + b_2 z^{-2}$$
$$C(z) = 1 + c_1 z^{-1} + c_2 z^{-2}$$

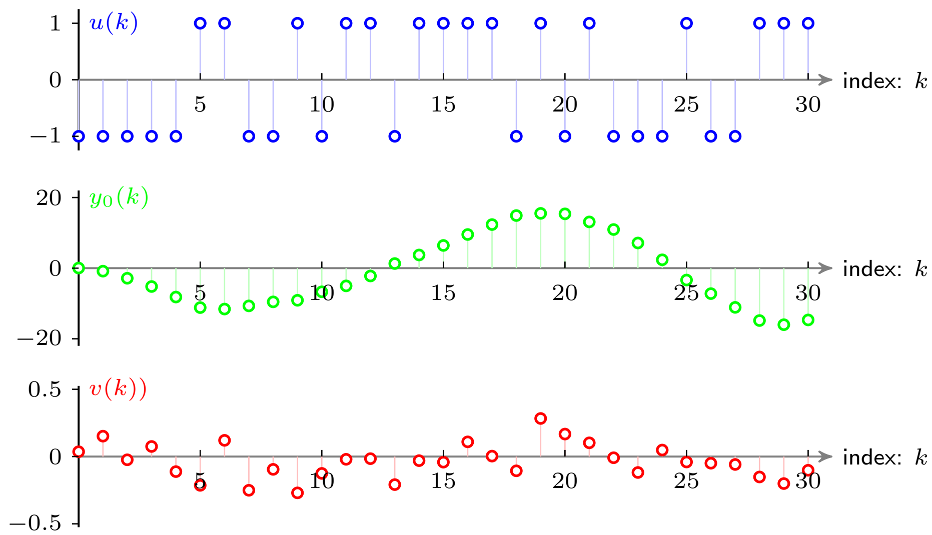
$$\theta = \begin{bmatrix} b_1 & b_2 & a_1 & a_2 & c_1 & c_2 \end{bmatrix}^T.$$

Experiments

- The plant is "at rest".
- Data length $K = 31$.
- PRBS input signal, $u(k)$.

ARMAX example

Typical experimental data



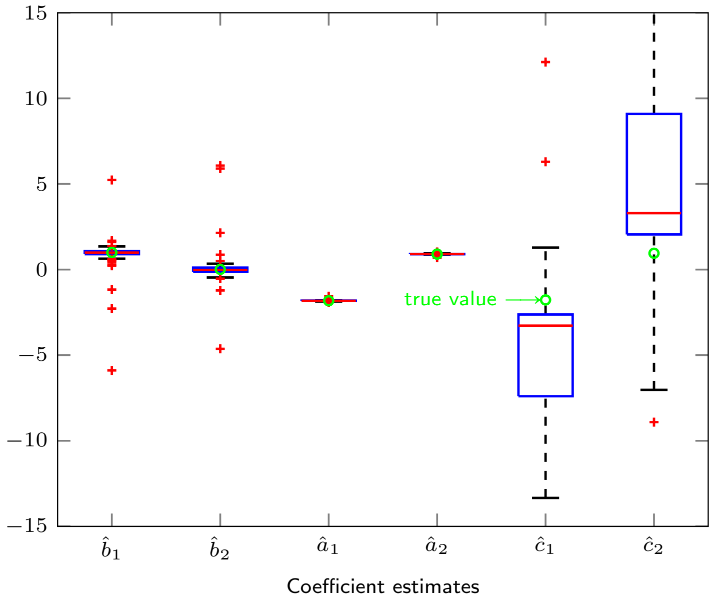
Constrained minimisation code (simplified for clarity)

```
PhiTyu(1,:)  = [0, 0, 0, 0];        % Regressor:  assume at rest
PhiTyu(2,:)  = [u(1),0, -y(1), 0];
for i = 3:K,
    PhiTyu(i,:)  = [u(i-1), u(i-2), -y(i-1), -y(i-2)];
end

[x,fval] = fmincon(@(x)ARMAXobjective(x),x0,...
          [],[],[],[],[],[],@(x)ARMAXconstraint(x,y,PhiTyu));

function [f] = ARMAXobjective(x)   % x = [theta; e]
f = sqrt(x(7:end)'*x(7:end));

function [c,ceq] = ARMAXconstraint(x,y,PhiTyu)
e = x(7:end);
PhiTe = zeros(K,2);
PhiTe(2,1) = e(1);
for j = 3:K,
   PhiTe(j,:)  = [e(j-1), e(j-2)];
end
ceq = y - [PhiTyu, PhiTe] * theta - e;    c = [];
```


ARMAX example

Coefficient estimate statistics for 128 experiments

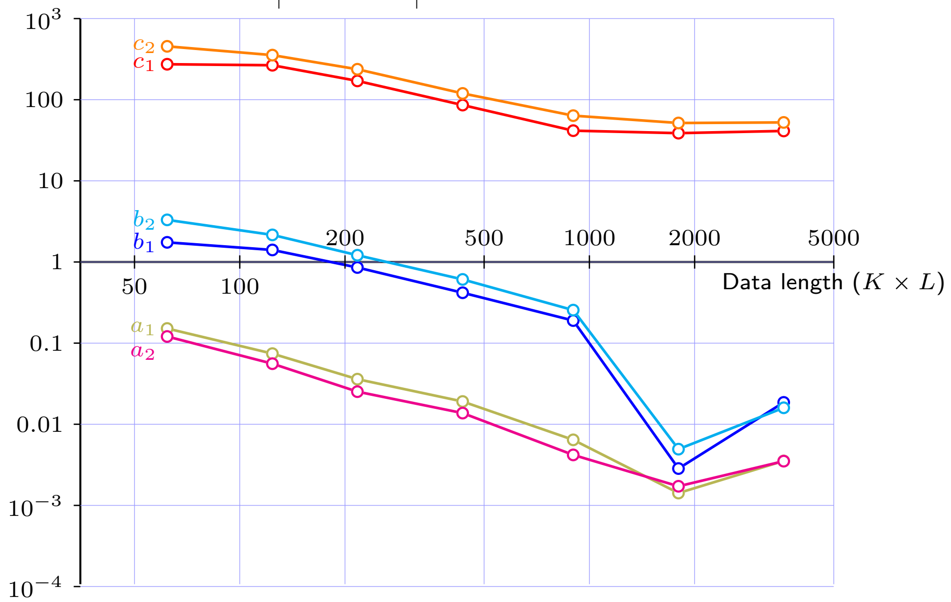
Experiment length: $K = 31$



ARMAX example

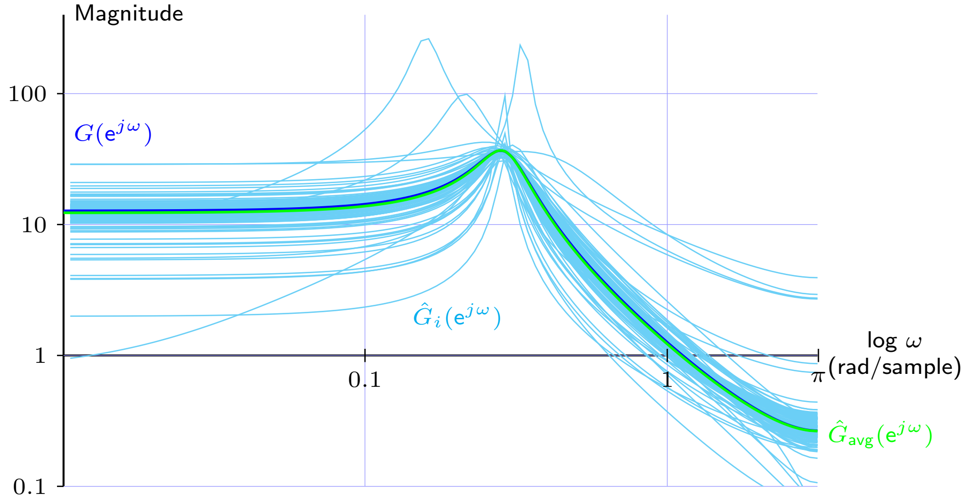
Coefficient error for averages: L = 2, 4, 8, . . . , 128 experiments

Error magnitude: $\left|\hat{\theta}(i) - \theta_0(i)\right|, \quad i = 1, \ldots, 6.$



ARMAX example

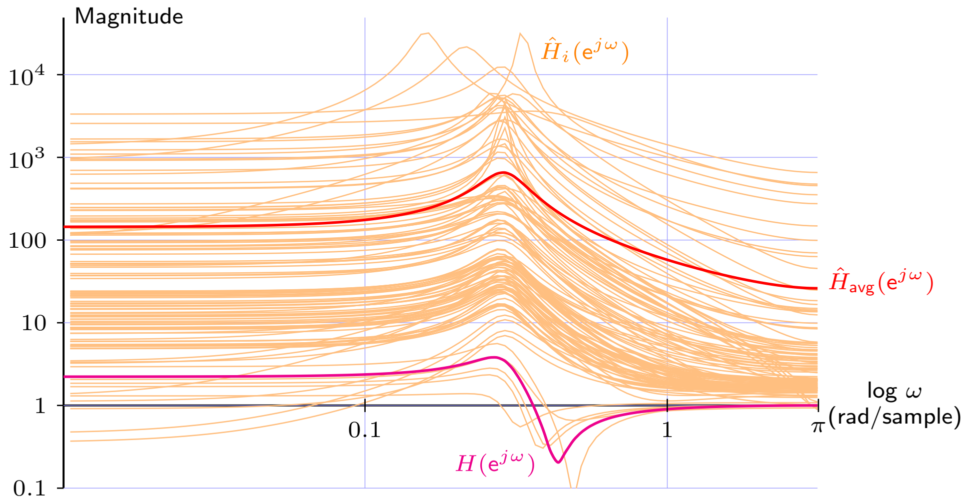
Transfer function estimates: 128 experiments (data length, $K = 31$)



$\hat{G}_{\mathsf{avg}}(z)$: transfer function with average value coefficients

ARMAX example

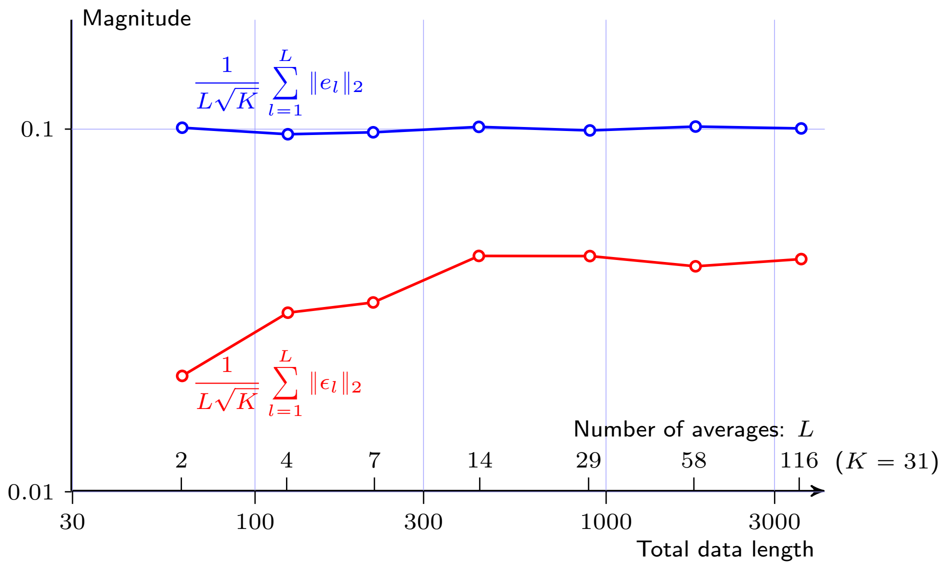
Transfer function estimates: 128 experiments (data length, $K = 31$)



$\hat{H}_{avg}(z)$: transfer function with average value coefficients

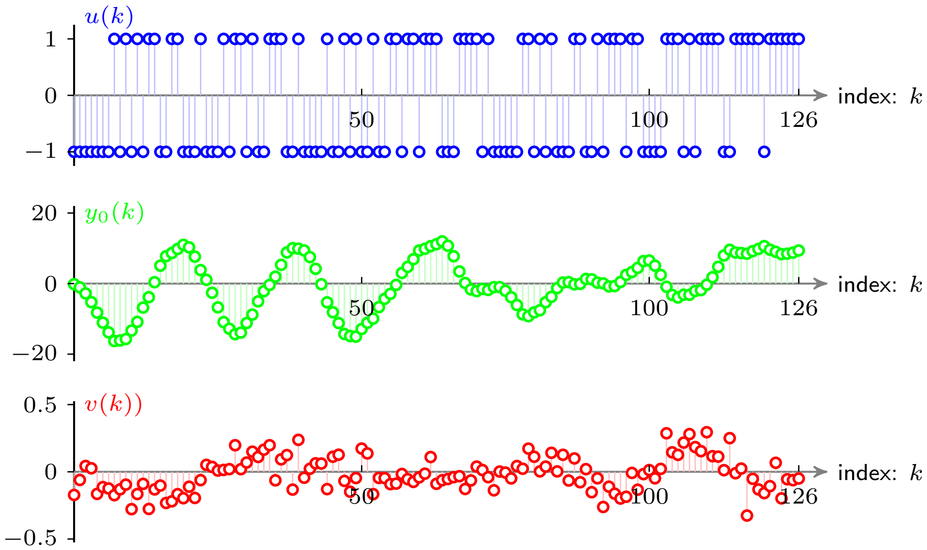
ARMAX example

Prediction errors and actual innovations (data length, $K = 31$)



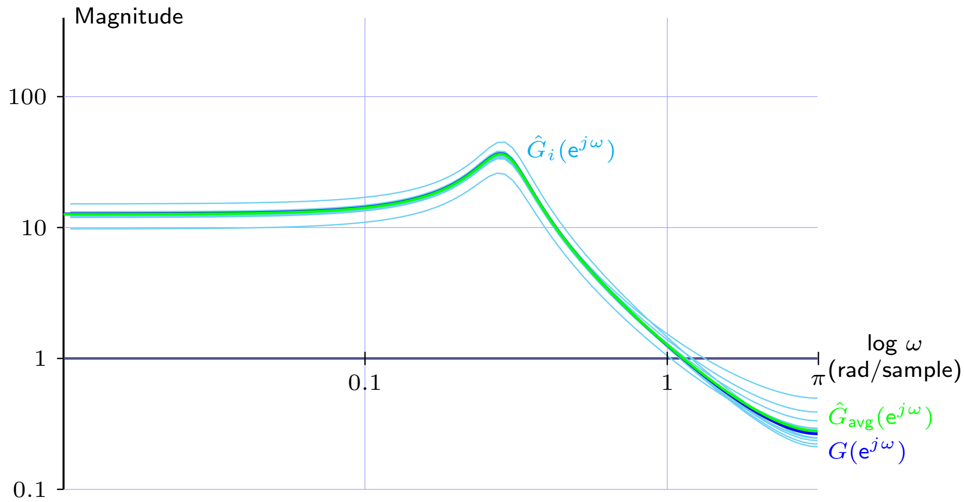
ARMAX example

Longer experiments: $K = 127$



ARMAX example

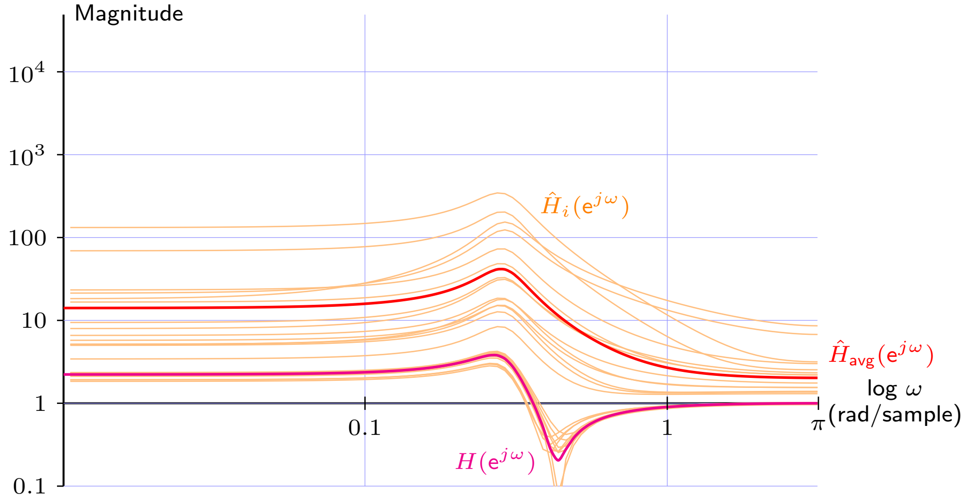
Transfer function estimates: 32 experiments (data length, $K = 127$)



$\hat{G}_{avg}(z)$: transfer function with average value coefficients

ARMAX example

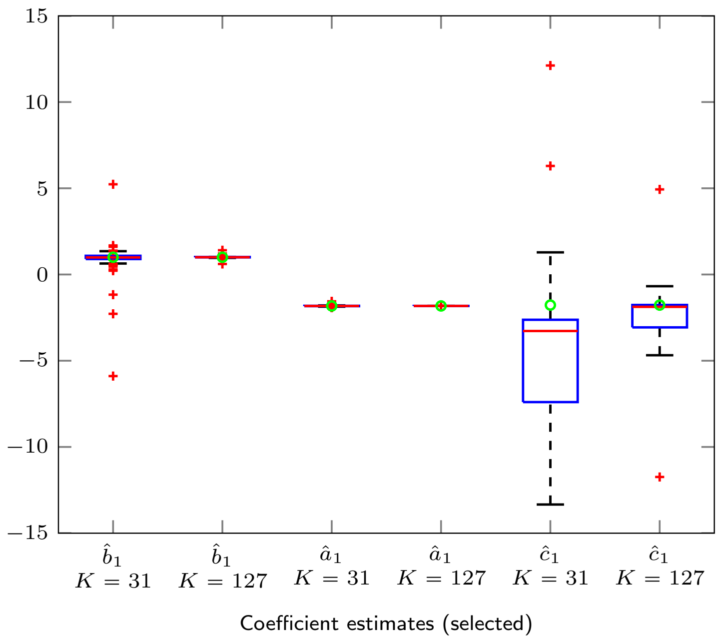
Transfer function estimates: 32 experiments (data length, $K = 127$)



$\hat{H}_{avg}(z)$: transfer function with average value coefficients

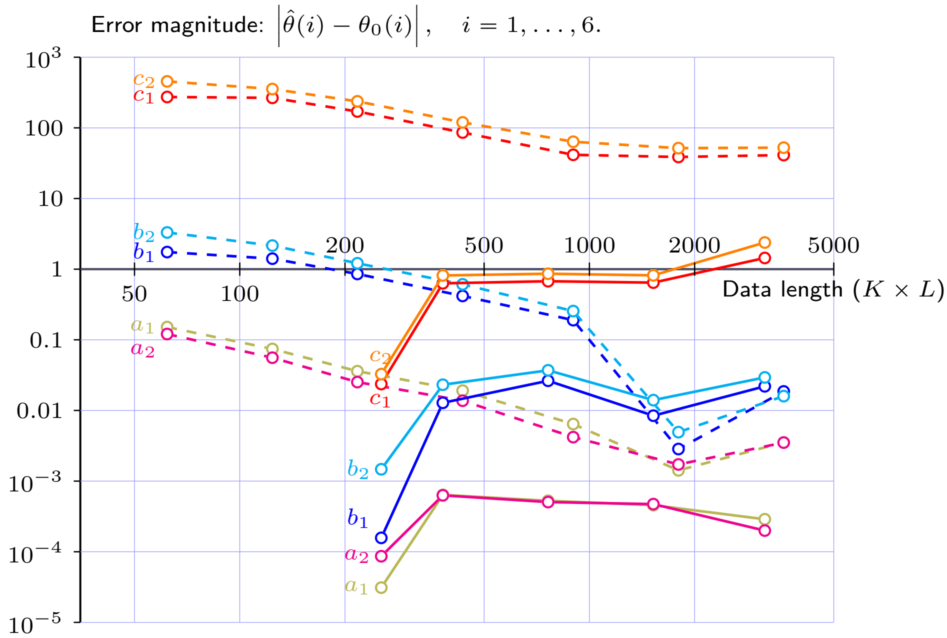
ARMAX example

Coefficient estimate statistics comparison: $K = 31$ and $K = 127$



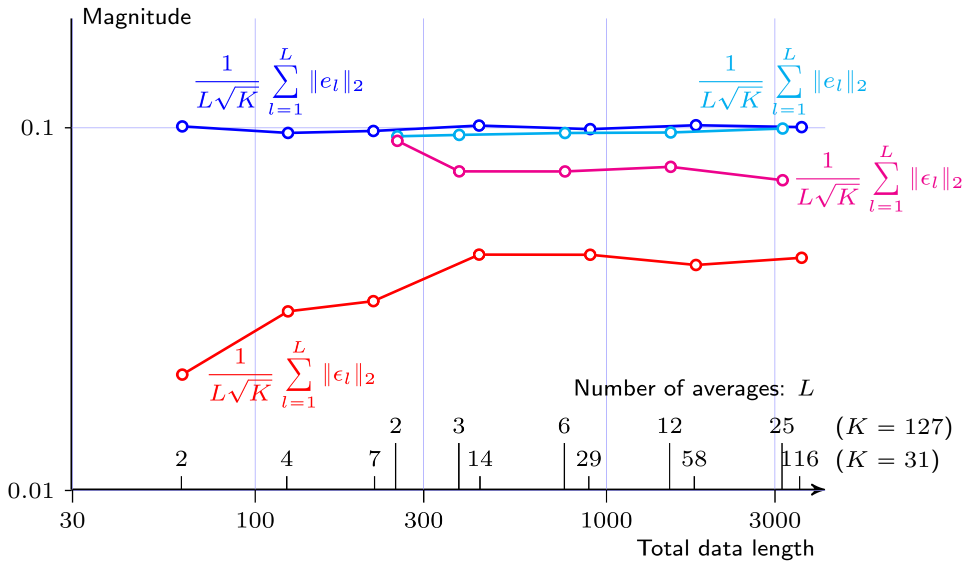
ARMAX example

Coefficient error comparison: $K = 31$ and $K = 127$

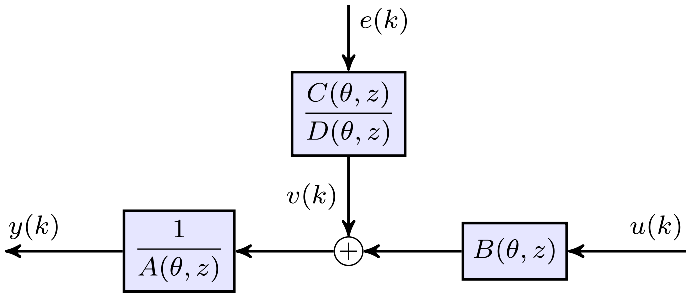


ARMAX example

Prediction error comparison: $K = 31$ and $K = 127$



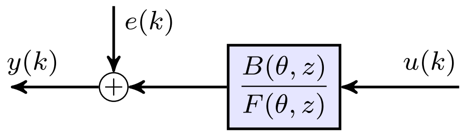
ARARMAX model structure



$$G(\theta, z) = \frac{B(\theta, z)}{A(\theta, z)}, \qquad H(\theta, z) = \frac{C(\theta, z)}{A(\theta, z)D(\theta, z)},$$

with $A(z)$, $C(z)$ and $D(z)$ monic.

Output error model structure



$$G(\theta, z) = \frac{B(\theta, z)}{F(\theta, z)},$$

$$H(\theta, z) = 1.$$

with $F(z)$ monic.

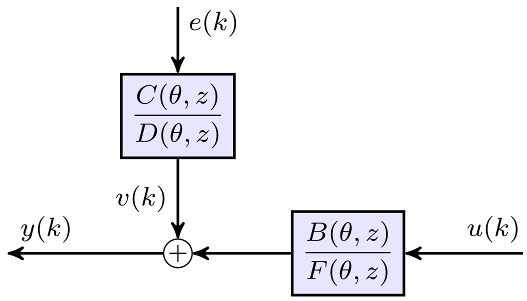
Pseudolinear predictor framework

$$\hat{y}(k|\theta) = \frac{B(\theta, z)}{F(\theta, z)} u(k) \quad = \phi(k, \theta)^T \theta.$$

where

$$\phi(k,\theta)^T = [\ u(k-1) \quad \dots \quad u(k-m) \quad \underbrace{-\hat{y}(k-1,\theta) \quad \dots \quad -\hat{y}(k-n_f,\theta)}_{\text{pseudolinear terms}}\ ].$$

Box-Jenkins model structure

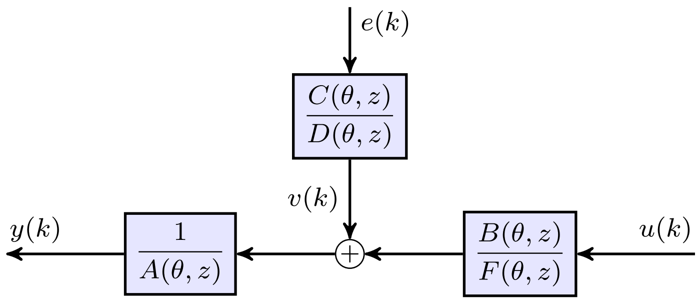


$$G(\theta, z) = \frac{B(\theta, z)}{F(\theta, z)}, \qquad H(\theta, z) = \frac{C(\theta, z)}{D(\theta, z)},$$

Predictor

$$\hat{y}(k|\theta) = \frac{D(z)}{C(z)}\frac{B(z)}{F(z)}u(k) + \left(1 - \frac{D(z)}{C(z)}\right) y(k)$$

General model structure



$$G(\theta, z) = \frac{B(\theta, z)}{A(\theta, z)F(\theta, z)}, \qquad H(\theta, z) = \frac{C(\theta, z)}{A(\theta, z)D(\theta, z)}$$

Predictor

$$\hat{y}(k|\theta) = \frac{D(z)}{C(z)}\frac{B(z)}{F(z)}u(k) + \left(1 - \frac{D(z)A(z)}{C(z)}\right) y(k)$$

A nonlinear regression can be derived for the prediction error.

Known noise model (with AR noise dynamics)

Assume that the MA part of the noise model is known,

$$v(k) = L(z)e(k)$$

So

$$A(z)y(k) = B(z)u(k) + L(z)e(k).$$

Filter signals via $L^{-1}(z)$,

$$y_L(k) = L^{-1}(z)y(k)$$
$$u_L(k) = L^{-1}(z)u(k)$$

Giving,

$$A(z)y_L(k) = B(z)u_L(k) + e(k),$$

for which LS methods give consistent estimates.

High-order model fitting

Assume that the noise is autoregressive (ARARX structure),

$$A(z)y(k) = B(z)u(k) + \frac{1}{D(z)}e(k) \qquad e(k) \sim \mathcal{N}(0, \lambda).$$

Fit a high order model (order of $D(z)$ is n_d):

$$A(z)D(z)y(k) = B(z)D(z)u(k) + e(k).$$

Least squares estimate with orders $n + n_d$ and $m + n_d$. This gives a consistent estimate of,

$$\frac{B(z)D(z)}{A(z)D(z)} = \frac{B(z)}{A(z)}.$$

This amounts to making the noise model sufficiently rich to capture additional autoregressive features in the noise.

In practice the cancellation will not be exact: $\hat{A}(z)$ and $\hat{B}(z)$ will be high order.

Bibliography

Prediction
Lennart Ljung, *System Identification;Theory for the User*, 2nd Ed., Prentice-Hall, 1999, [section 3.2].

Model parametrisations
Lennart Ljung, *System Identification;Theory for the User*, 2nd Ed., Prentice-Hall, 1999, [sections 1.3 and 4.2].

Nonlinear regression
Lennart Ljung, *System Identification;Theory for the User*, 2nd Ed., Prentice-Hall, 1999, [sections 10.1 and 10.2].