



Time Series Analysis

Henrik Madsen

`hm@imm.dtu.dk`

Informatics and Mathematical Modelling
Technical University of Denmark
DK-2800 Kgs. Lyngby



Outline of the lecture

Input-Output systems

- The z -transform – important issues from Sec. 4.4
- Cross Correlation Functions – from Sec. 6.2.2
- Transfer function models; identification, estimation, validation, prediction, Chap. 8



The z -transform

- A way to describe dynamical systems in discrete time

$$Z(\{x_t\}) = X(z) = \sum_{t=-\infty}^{\infty} x_t z^{-t} \quad (z \text{ complex})$$

- The z -transform of a time delay: $Z(\{x_{t-\tau}\}) = z^{-\tau} X(z)$

- The *transfer function* of the system is called $H(z) = \sum_{t=-\infty}^{\infty} h_t z^{-t}$

$$y_t = \sum_{k=-\infty}^{\infty} h_k x_{t-k} \Leftrightarrow Y(z) = H(z)X(z)$$

- Relation to the *frequency response function*: $\mathcal{H}(\omega) = H(e^{i\omega})$



Cross covariance and cross correlation functions

Estimate of the cross covariance function:

$$C_{XY}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (X_t - \bar{X})(Y_{t+k} - \bar{Y})$$

$$C_{XY}(-k) = \frac{1}{N} \sum_{t=1}^{N-k} (X_{t+k} - \bar{X})(Y_t - \bar{Y})$$

Estimate of the cross correlation function:

$$\hat{\rho}_{XY}(k) = C_{XY}(k) / \sqrt{C_{XX}(0)C_{YY}(0)}$$

If at least one of the processes is white noise and if the processes are uncorrelated then $\hat{\rho}_{XY}(k)$ is approximately normally distributed with mean 0 and variance $1/N$



Systems without measurement noise



$$Y_t = \sum_{i=-\infty}^{\infty} h_i X_{t-i}$$

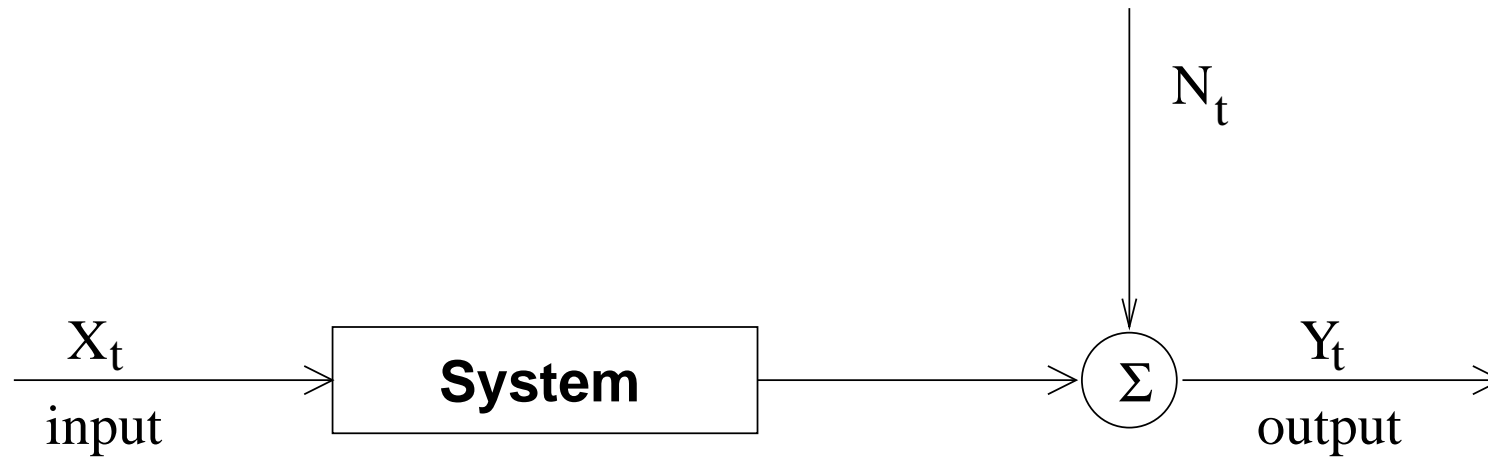
Given γ_{XX} and the system description we obtain

$$\gamma_{YY}(k) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_i h_j \gamma_{XX}(k - j + i) \quad (1)$$

$$\gamma_{XY}(k) = \sum_{i=-\infty}^{\infty} h_i \gamma_{XX}(k - i). \quad (2)$$



Systems with measurement noise



$$Y_t = \sum_{i=-\infty}^{\infty} h_i X_{t-i} + N_t.$$



Time domain relations

Given γ_{XX} and γ_{NN} we obtain

$$\gamma_{YY}(k) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_i h_j \gamma_{XX}(k - j + i) + \gamma_{NN}(k) \quad (3)$$

$$\gamma_{XY}(k) = \sum_{i=-\infty}^{\infty} h_i \gamma_{XX}(k - i). \quad (4)$$

IMPORTANT ASSUMPTION: No feedback in the system.



Spectral relations

$$\begin{aligned}f_{YY}(\omega) &= H(e^{-i\omega})H(e^{i\omega})f_{XX}(\omega) + f_{NN}(\omega) \\ &= G^2(\omega)f_{XX}(\omega) + f_{NN}(\omega),\end{aligned}$$

$$f_{XY}(\omega) = H(e^{i\omega})f_{XX}(\omega) = \mathcal{H}(\omega)f_{XX}(\omega).$$

The frequency response function, which is a complex function, is usually split into a modulus and argument

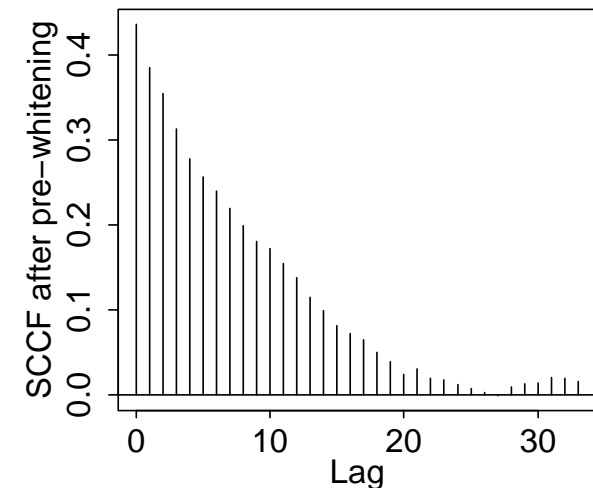
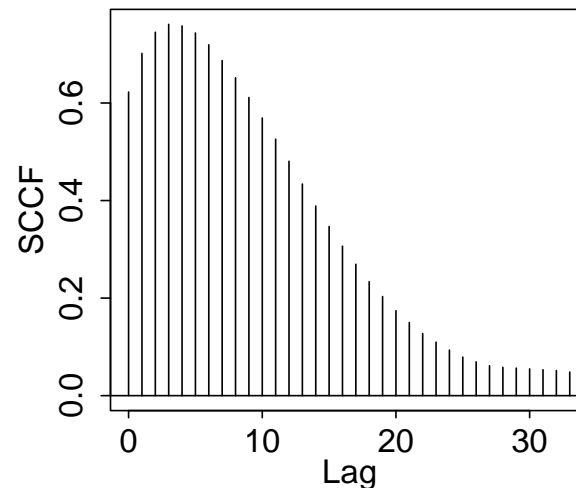
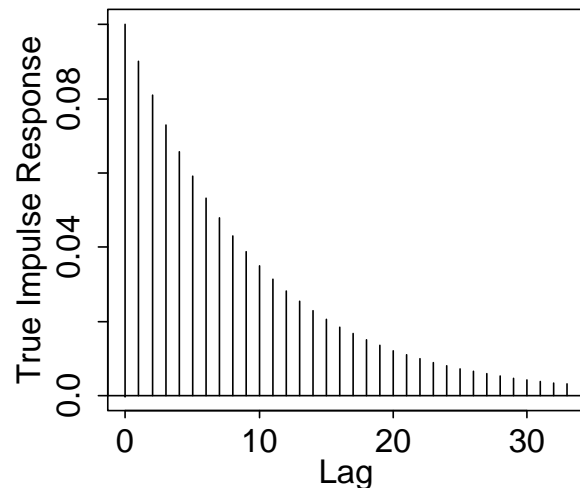
$$\mathcal{H}(\omega) = |\mathcal{H}(\omega)| e^{i \arg\{\mathcal{H}(\omega)\}} = G(\omega)e^{i\phi(\omega)},$$

where $G(\omega)$ and $\phi(\omega)$ are the *gain* and *phase*, respectively, of the system at the frequency ω from the input $\{X_t\}$ to the output $\{Y_t\}$.



Estimating the impulse response

- The poles and zeros characterize the impulse response (Appendix A and Chapter 8)
- If we can estimate the impulse response from recordings of input an output we can get information that allows us to *suggest a structure for the transfer function*





Estimating the impulse response

- On the previous slide we saw that we got a good picture of the true impulse response when pre-whitening the data
- The reason is

$$\gamma_{XY}(k) = \sum_{i=-\infty}^{\infty} h_i \gamma_{XX}(k-i)$$

- and only if $\{X_t\}$ is white noise we get $\gamma_{XY}(k) = h_k \sigma_X^2$
- Therefore if $\{X_t\}$ is white noise the SCCF $\hat{\rho}_{XY}(k)$ is proportional to \hat{h}_k
- Normally $\{X_t\}$ is not white noise – we fix this using pre-whitening



Pre-whitening

a) A suitable ARMA-model is applied to the input series:

$$\varphi(B)X_t = \theta(B)\alpha_t.$$

b) We perform a *prewhitening* of the input series

$$\alpha_t = \theta(B)^{-1}\varphi(B)X_t$$

c) The output-series $\{Y_t\}$ is filtered with the same model, i.e.

$$\beta_t = \theta(B)^{-1}\varphi(B)Y_t.$$

d) Now the *impulse response function* is estimated by

$$\hat{h}_k = C_{\alpha\beta}(k)/C_{\alpha\alpha}(0) = C_{\alpha\beta}(k)/S_{\alpha}^2.$$



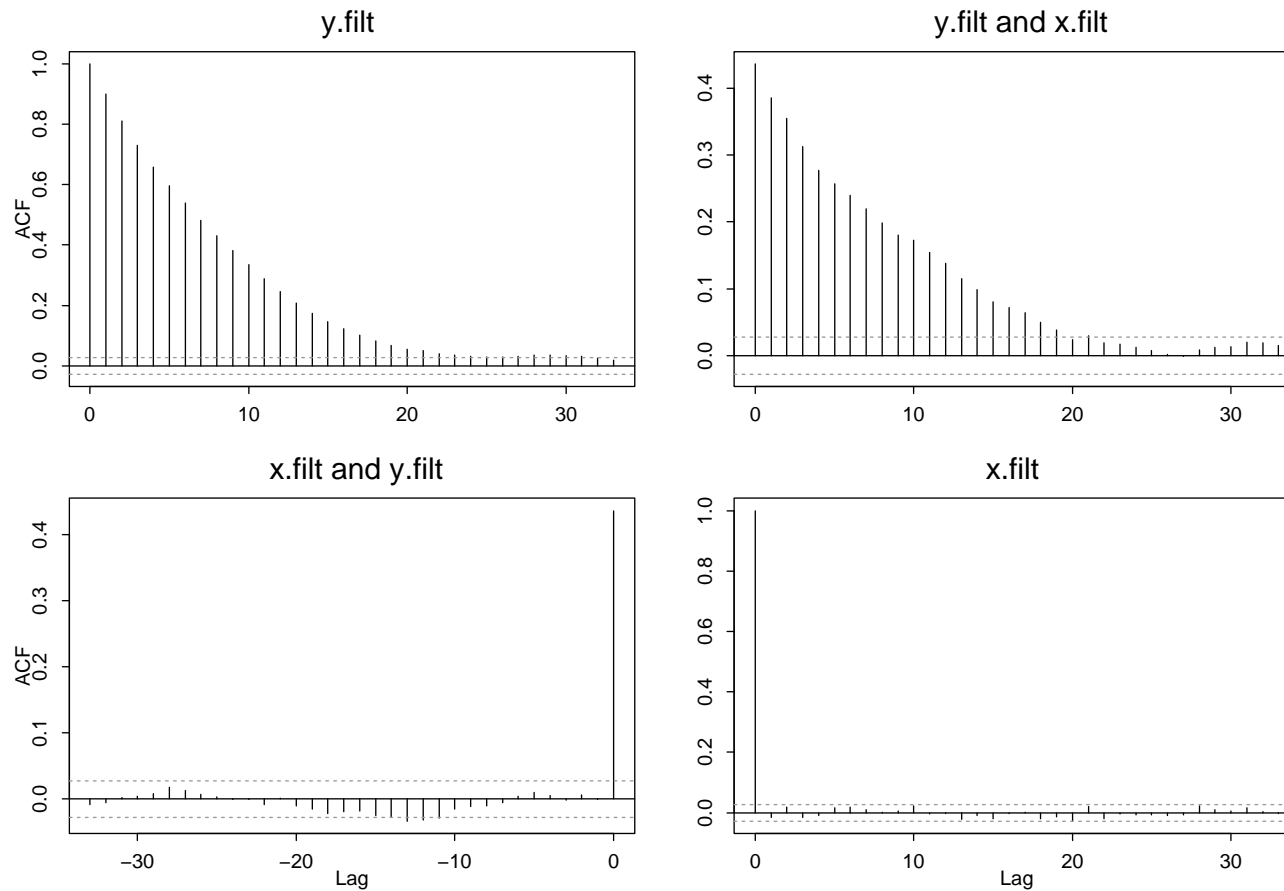
Example using S-PLUS

```
## ARMA structure for x; AR(1)
x.struct <- list(order=c(1,0,0))
## Estimate the model (check for convergence):
x.fit <- arima.mle(x - mean(x), model=x.struct)
## Extract the model:
x.mod <- x.fit$model
## Filter x:
x.start <- rep(mean(x), 1000)
x.filt <- arima.sim(model=list(ma=x.mod$ar),
                   innov=x, start.innov = x.start)
## Filter y:
y.start <- rep(mean(y), 1000)
y.filt <- arima.sim(model=list(ma=x.mod$ar),
                   innov=y, start.innov = y.start)
## Estimate SCCF for the filtered series:
acf(cbind(y.filt, x.filt))
```



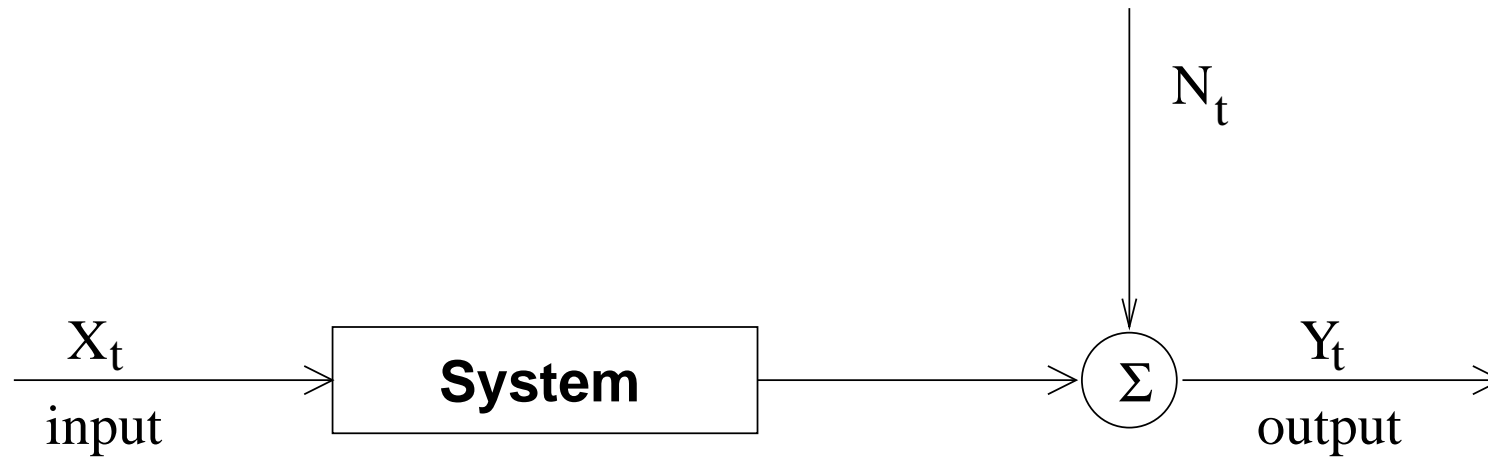
Graphical output

Multivariate Series : `cbind(y.filt, x.filt)`





Systems with measurement noise

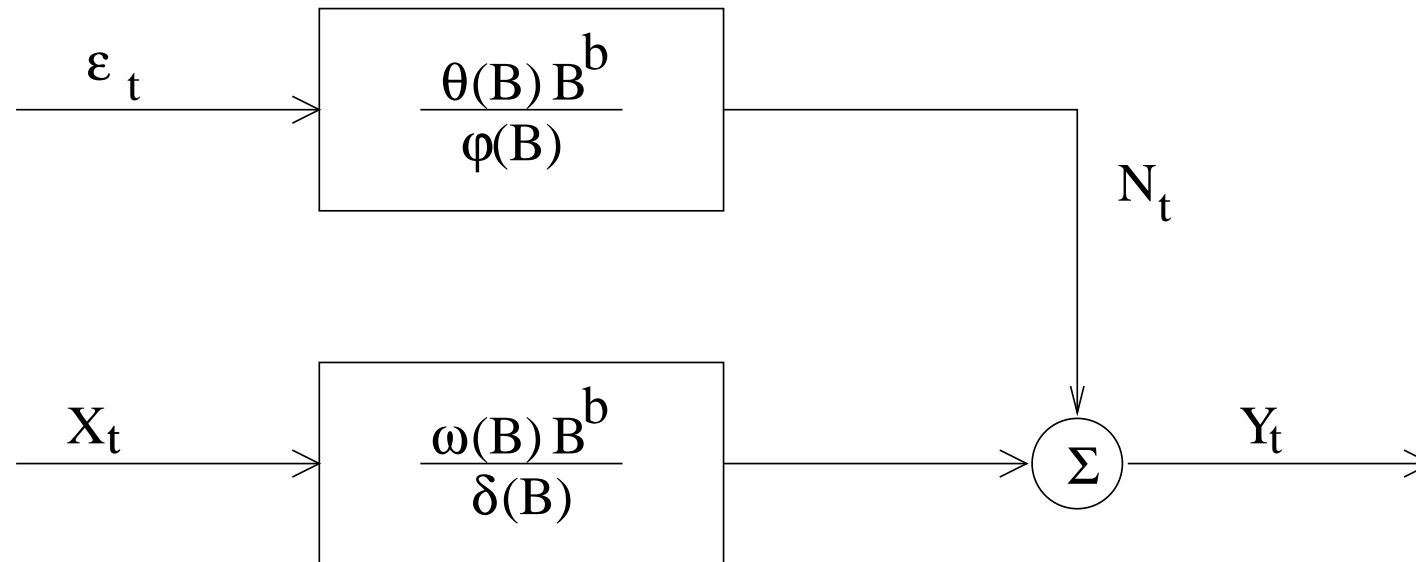


$$Y_t = \sum_{i=-\infty}^{\infty} h_i X_{t-i} + N_t.$$

$$\gamma_{XY}(k) = \sum_{i=-\infty}^{\infty} h_i \gamma_{XX}(k-i)$$



Transfer function models



$$Y_t = \frac{\omega(B)}{\delta(B)} B^b X_t + \frac{\theta(B)}{\varphi(B)} \epsilon_t$$

- Also called Box-Jenkins models
- Can be extended to include more inputs – see the book.



Some names

- FIR: Finite Impulse Response
- ARX: Auto Regressive with eXternal input
- ARMAX/CARMA: Auto Regressive Moving Average with eXternal input / Controlled ARMA
- OE: Output Error
- Regression models with ARMA noise



Identification of transfer function models

$$h(B) = \frac{\omega(B)B^b}{\delta(B)} = h_0 + h_1B + h_2B^2 + h_3B^3 + h_4B^4 + \dots$$

- Using pre-whitening we estimate the impulse response and “guess” an appropriate structure of $h(B)$ based on this (see page 197 for examples).
- It is a good idea to experiment with some structures. Matlab (use q^{-1} instead of B):

```
A = 1; B = 1; C = 1; D = 1;  
F = [1 -2.55 2.41 -0.85];  
mod = idpoly(A, B, C, D, F, 1, 1)  
impulse(mod)
```

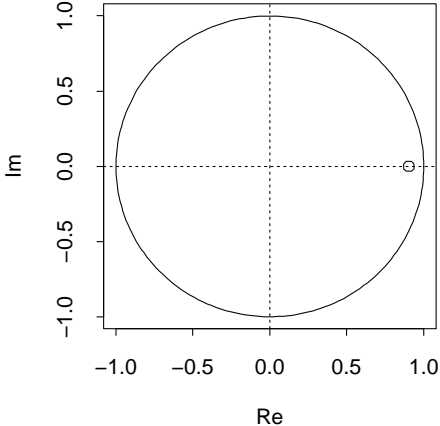
- PEZdemo (complex poles/zeros should be in pairs):
<http://users.ece.gatech.edu/mcclella/matlabGUIs/>



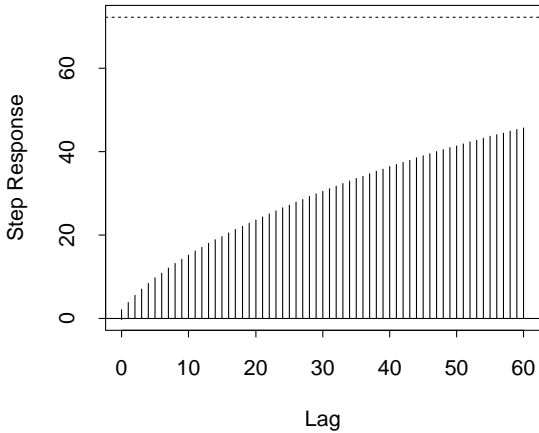
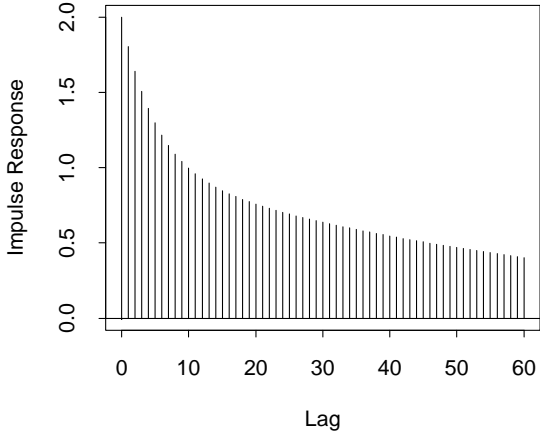
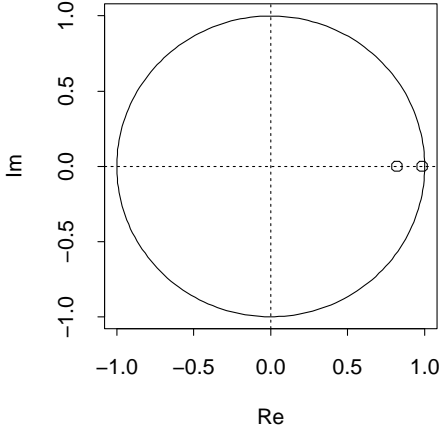
2 exponential

$$\frac{2-1.8B}{1-1.8B+0.81B^2}$$

Zeros



Poles

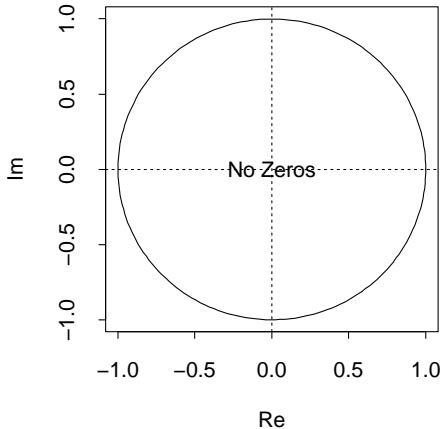




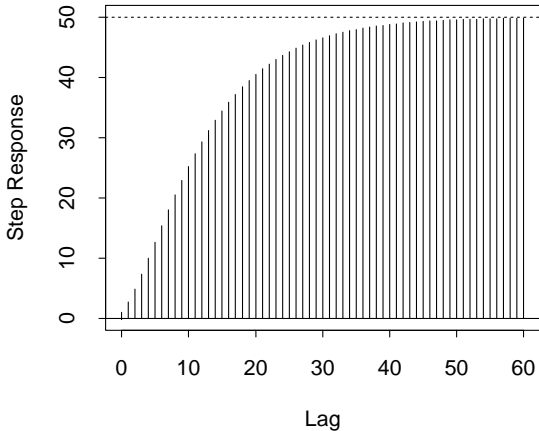
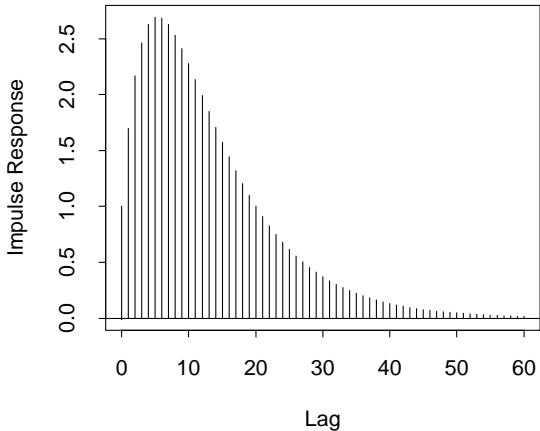
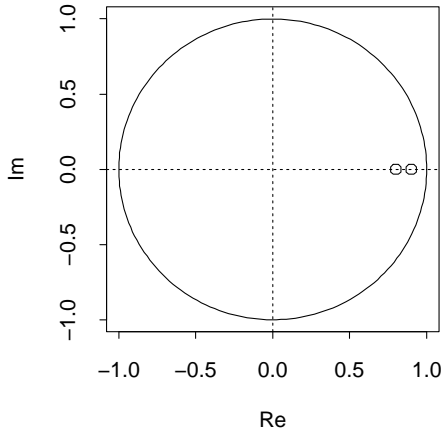
2 real poles

$$\frac{1}{1-1.7B+0.72B^2}$$

Zeros



Poles

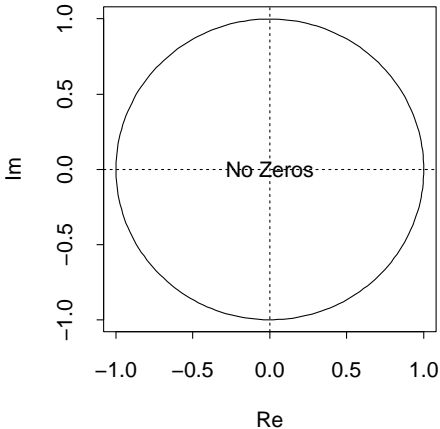




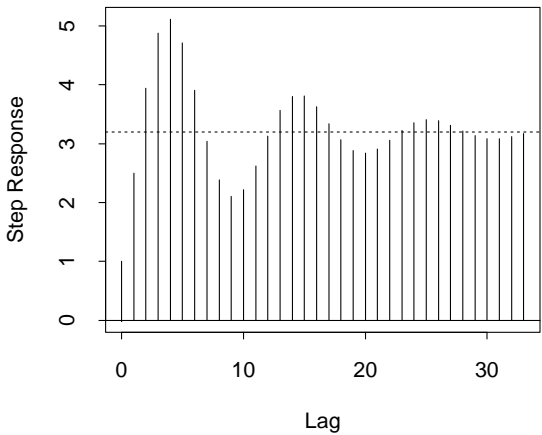
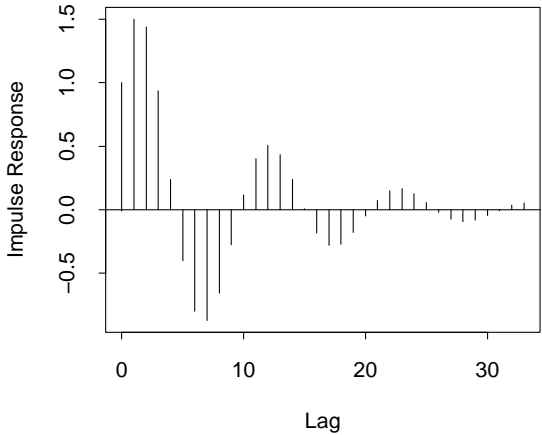
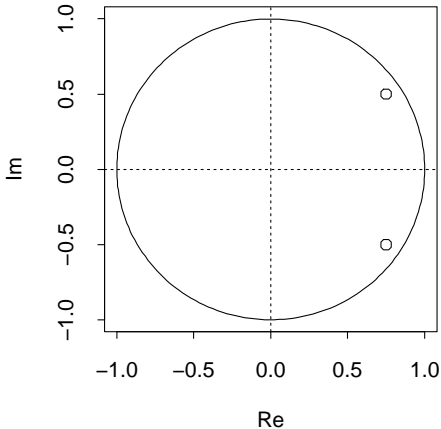
2 complex

$$\frac{1}{1-1.5B+0.81B^2}$$

Zeros



Poles



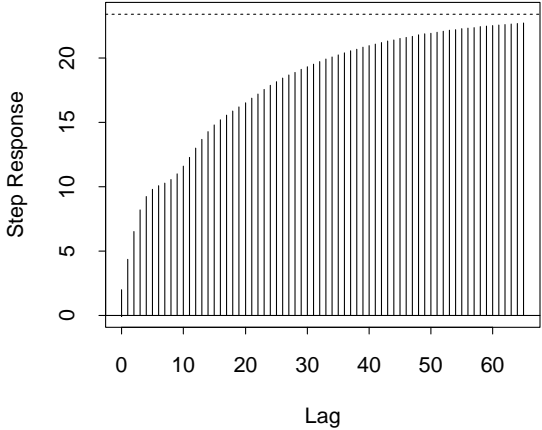
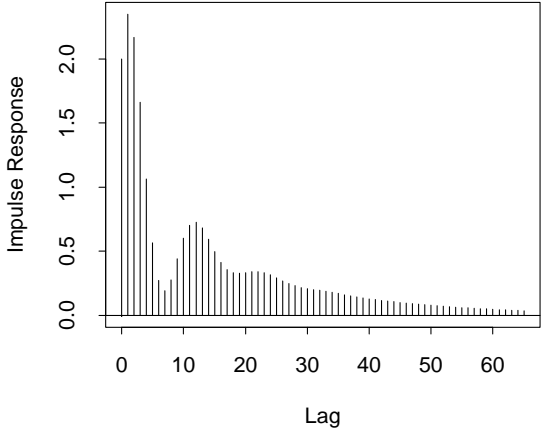
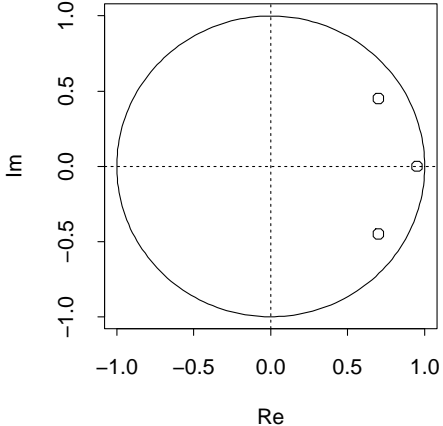
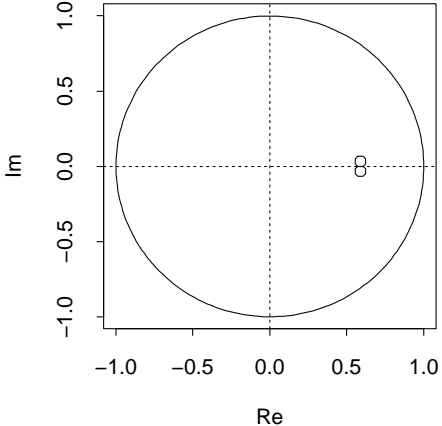


1 exp + 2 comp

$$\frac{2 - 2.35B + 0.69B^2}{1 - 2.35B + 2.02B^2 - 0.66B^3}$$

Zeros

Poles





Identification of the transfer function for the noise

- After selection of the structure of the transfer function of the input we estimate the parameters of the model

$$Y_t = \frac{\omega(B)}{\delta(B)} B^b X_t + N_t$$

- We extract the residuals $\{N_t\}$ and identifies a structure for an ARMA model of this series

$$N_t = \frac{\theta(B)}{\varphi(B)} \varepsilon_t \quad \Leftrightarrow \quad \varphi(B)N_t = \theta(B)\varepsilon_t$$

- We then have the full structure of the model and we estimate all parameters simultaneously



Estimation

- Form 1-step predictions, treating the input $\{X_t\}$ as known in the future (if $\{X_t\}$ is really stochastic we *condition* on the observed values)
- Select the parameters so that the sum of squares of these errors is as small as possible
- If $\{\varepsilon_t\}$ is normal then the ML estimates are obtained
- For FIR and ARX models we can write the model as $Y_t = \mathbf{X}_t^T \boldsymbol{\theta} + \varepsilon_t$ and use LS-estimates
- Moment estimates: Based on the structure of the transfer function we find the theoretical impulse response and we make a match with the lowest lags in the estimated impulse response
- Output error estimates ...



Model validation

As for ARMA models with the additions:

- Test for cross correlation between the residuals and the input

$$\hat{\rho}_{\varepsilon X}(k) \sim \text{Norm}(0, 1/N)$$

which is (approximately) correct when $\{\varepsilon_t\}$ is white noise and when there is no correlation between the input and the residuals

- A *Portmanteau test* can also be performed



Prediction $\hat{Y}_{t+k|t}$

We must consider two situations

- The input is controllable, i.e. we can decide it and we can predict under different input-scenarios. In this case the prediction error variance is originating from the ARMA-part only (N_t).
- The input is only known until the present time point t and to predict the output we must predict the input. In this case the prediction error variance depend also on the autocovariance of the input process. In the book the case where the input can be modelled as an ARMA-process is considered.



Prediction (cont'nd)

$$\hat{Y}_{t+k|t} = \sum_{i=0}^{k-1} h_i \hat{X}_{t+k-i|t} + \sum_{i=k}^{\infty} h_i X_{t+k-i} + \hat{N}_{t+k|t}.$$

$$Y_{t+k} - \hat{Y}_{t+k|t} = \sum_{i=0}^{k-1} h_i (X_{t+k-i} - \hat{X}_{t+k-i|t}) + N_{t+k} - \hat{N}_{t+k|t}$$

- If the input is controllable then $\hat{X}_{t+k-i|t} = X_{t+k-i}$
- The book also considers the case where output is known until time t and input until time $t + j$



Prediction (cont'nd)

- We have

$$N_t = \sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i}$$

- And if we model the input as an ARMA-process we have

$$X_t = \sum_{i=0}^{\infty} \bar{\psi}_i \eta_{t-i}$$

- And thereby we get:

$$V[Y_{t+k} - \hat{Y}_{t+k|t}] = \sigma_{\eta}^2 \sum_{\ell=0}^{k-1} \left(\sum_{i_1+i_2=\ell} h_{i_1} \bar{\psi}_{i_2} \right)^2 + \sigma_{\varepsilon}^2 \sum_{i=0}^{k-1} \psi_i^2$$



$$Y_t = \frac{0.4}{1-0.6B} X_t + \frac{1}{1-0.4} \varepsilon_t, \quad \sigma_\varepsilon^2 = 0.036$$

	y	x	h	xf	N		y	x	h	xf	N
1	2.04	1.661	0.00403	2.21	-0.1645		2.04	1.661	0.00403	2.21	-0.1645
2	3.05	4.199	0.00672	3.00	0.0407		3.05	4.199	0.00672	3.00	0.0407
3	2.34	1.991	0.01120	2.60	-0.2566		2.34	1.991	0.01120	2.60	-0.2566
4	2.49	2.371	0.01866	2.51	-0.0186		2.49	2.371	0.01866	2.51	-0.0186
5	3.30	3.521	0.03110	2.91	0.3826		3.30	3.521	0.03110	2.91	0.3826
6	3.53	3.269	0.05184	3.06	0.4768		3.53	3.269	0.05184	3.06	0.4768
7	2.72	0.741	0.08640	2.13	0.5880		2.72	0.741	0.08640	2.13	0.5880
8	2.46	2.238	0.14400	2.17	0.2888		2.46	2.238	0.14400	2.17	0.2888
9	NA	2.544	0.24000	2.32	NA		2.44	2.544	0.24000	2.32	0.1155
10	NA	3.201	0.40000	2.67	NA		2.72	3.201	0.40000	2.67	0.0462

To forecast y (9,10) we must filter x as in xf , calc. N for the historic data, forecast N and add that to xf (future values)

```
> Nfc <- arima.forecast(N[1:8], model=list(ar=0.4), sigma2=0.036, n=2)
> Nfc$mean:
[1] 0.1155 0.0462
```



Intervention models

$$I_t = \begin{cases} 1 & t = t_0 \\ 0 & t \neq t_0 \end{cases}$$

$$Y_t = \frac{\omega(B)}{\delta(B)} I_t + \frac{\theta(B)}{\phi(B)} \varepsilon_t$$

See a real life example in the book.