Homework 2

The absolutely last date to hand in the homework is the 1:st of june, but please try to do it as early as possible. If you get stuck on something do not hesitate to ask me.

1 An eigth order linear model has been used to generate the data sequence y that is included in the file "y_data.mat" that can be obtained from the course homepage.

Use that data to estimate covariances by the truncated ergodic sum:

$$\hat{r}_k = \frac{1}{N} \sum_{\ell=1}^{N-k} y_\ell y_{\ell+k}.$$

Then use exercise 6.7 to find a realization of a linear model, with transfer function Φ_+, matching the covariances $\hat{r}_0/2, \hat{r}_1, \cdots, \hat{r}_{16}$.

Show that the function Φ_+ is positive real. Then determine the spectral factor W_- of $\Phi(z) = \Phi_+(z) + \Phi_+(z^{-1})$ corresponding to the forward predictor space.

Hint: To do the last step, it could be useful to iterate equation (6.9.8), or to use the command dare *in Matlab.*

The spectral density of the data sequence y can be estimated in Matlab using the command "psd(y)". Compare this estimate with the spectral density corresponding to W_- and the spectral density of the generating model. The spectral density of W_- can be plotted using the program "modelmag.m" available at the homepage. The data for the spectral density of the generating model is included in the data file "y_data.mat".

What happens if you try to match a fourth order model instead ?

2 Is the Frame space $\mathbf{H}^\square$ a minimal Markovian splitting subspace ?

(In general? / In some special situation ?)

We know that $\mathbf{H}^\square \sim (\mathbf{S}, \bar{\mathbf{S}})$ for $\mathbf{S} = (\mathbf{N}^+)^\perp$ and $\bar{\mathbf{S}} = (\mathbf{N}^-)^\perp$.

Use Theorem 7.4.3 to determine $(\mathbf{S}_1, \bar{\mathbf{S}}_1)$, and then $\mathbf{X}_1 = \mathbf{S}_1 \cap \bar{\mathbf{S}}_1$.

3 Use the same data as in 1, i.e. the data in "y_data.mat". Another approach to realizing a system from the data is to do something like this:

We would like to define a state. Let

$$x(t) = \begin{bmatrix} y(t-1) \\ y(t-2) \\ \vdots \\ y(t-n) \end{bmatrix},$$

and we will assume that $\mathbf{X} = \{a'x(0) | a \in \mathbb{R}^n\}$ is a Markovian splitting subspace. Is $\mathbf{X}$ an internal state ?

We could estimate a dynamics matrix A by minimizing

$$\left\| \sum_{t=n}^{N-1} x(t+1) - Ax(t) \right\|^2 .$$

To solve this least-squares problem it can be useful to form the matrices:

$$H_0 = \begin{bmatrix} y(n) & y(n+1) & \cdots & y(N-1) \\ y(n-1) & y(n) & \cdots & y(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ y(1) & y(2) & \dots & y(N-n-1) \end{bmatrix},$$

and

$$H_1 = \begin{bmatrix} y(n+1) & y(n+2) & \cdots & y(N) \\ y(n) & y(n+1) & \cdots & y(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ y(2) & y(3) & \dots & y(N-n) \end{bmatrix}.$$

This method for determining A should be equivalent to approximating (8.3.19) with truncated ergodic sums, i.e.

$$\begin{aligned} A &= \mathrm{E}\{x(1)x(0)'\}P^{-1} \\ &\approx \frac{1}{N-n-1} \sum_{t=n+1}^{N-1} x(t+1)x(t)' \left(\frac{1}{N-n} \sum_{t=n+1}^{N} x(t)x(t)' \right)^{-1} . \end{aligned}$$

Use one of these methods to estimate A for the case $n = 8$.

To estimate B, we could again proceed in different ways. First, define $w(t) = x(t+1) - Ax(t)$. Assuming that the system is driven by normalized white noise, B can be estimated by factoring

$$\mathrm{E}w(t)w(t)' \approx \frac{1}{N-n} \sum_{t=n+1}^{N} w(t)w(t)'.$$

Finally, C and D can be obtained by noting that $y(t)$ is the first component of $x(t+1)$, and extracting the corresponding parts of A and B.

As in 1, the spectral density of the data sequence y can be estimated in Matlab using the command "psd(y)". Compare this estimate with the spectral density corresponding to the model determined here and the spectral density of the generating model.

4 Consider a state space (forward) system with matrices A, B, C and D given in the file "model4.mat" available at the homepage.

Generate a stationary stochastic process y by feeding white noise through this system. This can be acchieved by using "dlsim(A,B,C,D,U,X0)" in Matlab, where U is the white noise that can determined by "randn(1,1000)", and X0 is the initial condition that can be chosen as "randn(4,1)".

Plot the output y.

Then determine the system in a basis adapted to the decomposition of the signal y into one p.d and one p.n.d part as in Theorem 8.4.8.

Then plot $y_0(t)$ and $y_\infty(t)$ in separate subplots.

Determine the corresponding backward model.