Signals & Systems

Laboratory no. 8 – Parametric models identification

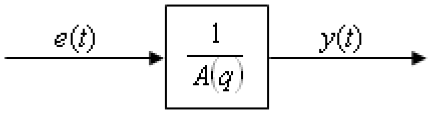
AR Model

AR Model (Auto Regressive) is defined as:

$$y(t) = -\sum_{i=1}^{n} a_i y(t-i) + e(t) \qquad (1)$$

where:

- $y(t)$ – system output,
- a_i – model parameters.
- n – order of the system,
- $y(t-1)$ – system output in the previous moment,
- $e(t)$ – white noise.



Equation (1) can be rewritten:

$$A(q)y(t) = e(t), \qquad A(q) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$$

where: q^{-1}- backward translation operator
Parameters estimation terror is defined as:

$$e(t) = y(t) - \hat{y}(t)$$

where $y(t)$ is an input signal to the estimation process and $\hat{y}(t)$ is an estimated signal.

ARMA Model

ARMA Model (Auto Regressive Moving Average) is an extension of the AR model and is defined as:

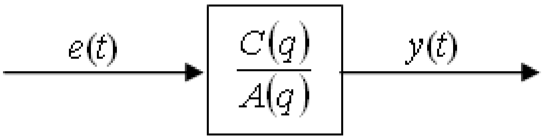
$$y(t) + a_1 y(t-1) + \ldots + a_n y(t-n) = e(t) + c_1 e(t-1) + \ldots + c_{n_c} y(t-n_c)$$

or

$$A(q)y(t) = C(q)e(t), \qquad A(q) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$$

$$C(q) = 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}$$

where $A(q)$ and $C(q)$ are polynomials of q.



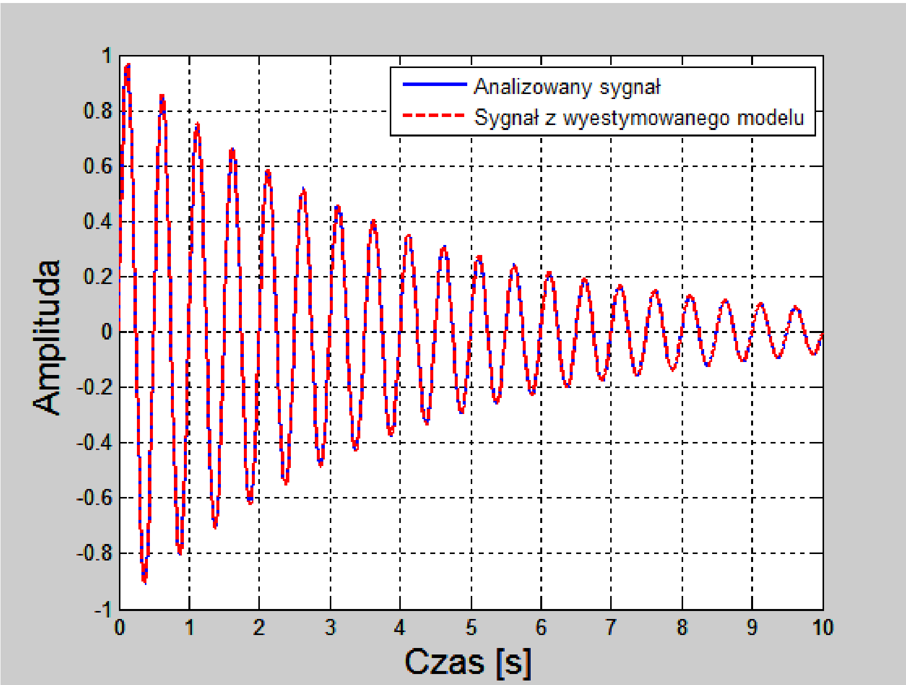
EXAMPLE 1

AR Model

Let us assume the 1DOF system response signal. Using the AR model let us estimate the system model. The exemplary code in Matlab can be found below.

```
clear
close all
t = 0:0.01:10;
y = exp(-0.02*2*pi*2*t).*sin(2*pi*2*t);
[mod] = ar(y,2);
Y_es = predict(mod,y');
figure
plot(t,y,'LineWidth',2)
hold on
plot(t,Y_es{1},'--r','LineWidth',2)
xlabel('Czas [s]','FontSize',16)
ylabel('Amplituda ','FontSize',16)
grid on
legend('Analizowany sygnal','Sygnal z wyestymowanego modelu')
```

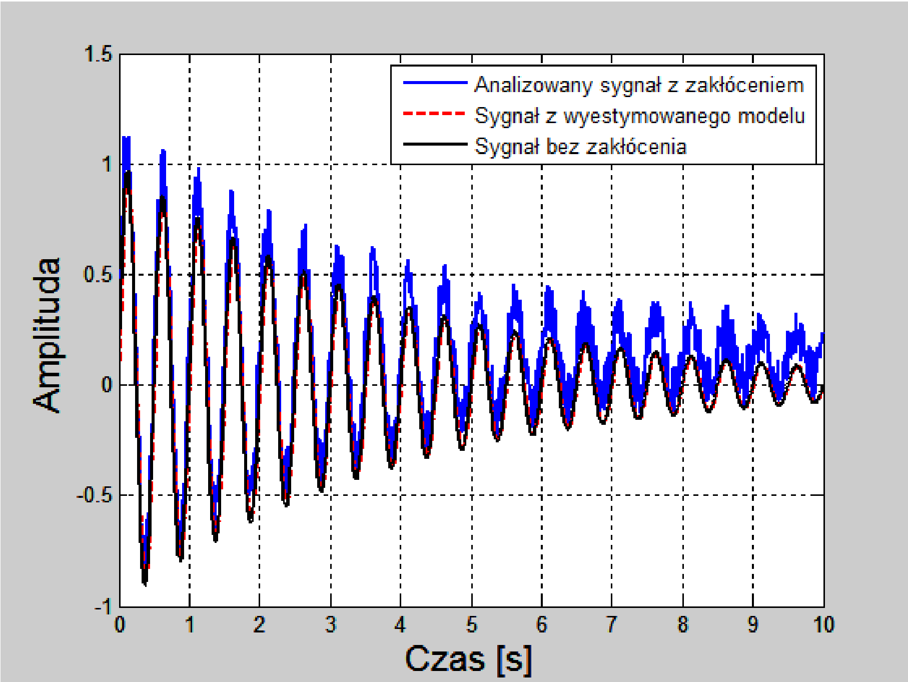
Result of the above Matlab cod eis presented in the figure.



Additionally we can add a noise to the signal.

```
y_nois = exp(-0.02*2*pi*2*t).*sin(2*pi*2*t)+rand(1,length(t))*0.25;
```

And repeat the model estimation process. Figure comparing obtained results is placed below.



EXAMPLE 2

ARMA Model

Let us create the state space model of the 1DOF system:

```
m=1;
k=3000;
c=2;
A = [0 1;-k/m -c/m];
B = [-1/m 0]';
C = [0 1];
D = 0;
sys = ss(A,B,C,D);
```

to obtain the opodal parameters of the system we use the function `damp`.

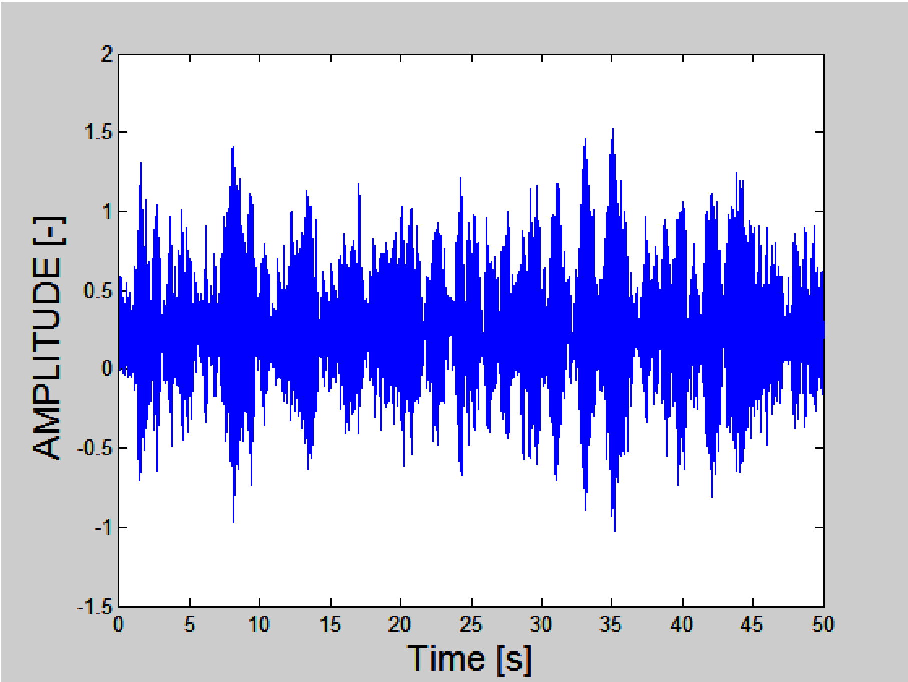
```
[Wn,Z] = damp(sys);
```

where: Wn natural frequencies of the system, Z – damping factors.

We will now simulate the system with use of the white noise excitation signal.

```
t = 0:0.01:50;
u = 0.5*rand(1,length(t));
Y_out = lsim(sys,u,t);
```

Result of the simulation can be seen in the figure.



Let us now estimate the ARMA model parameters and compare obtained results with the given ones:

```
model = arma([Y_out],[2 2]);
model_z = zpk(model)
[Wn_es, Z_es P] = damp(model_z);
[Wn(2)/2/pi Wn_es(1)/2/pi*100;Z(1) Z_es(1)]
```

Given natural frequencies [Hz]	8.7173
Estimated natural frequencies [Hz]	7.6344
Given damping factors	0.0183
Estimated damping factors	0.0912

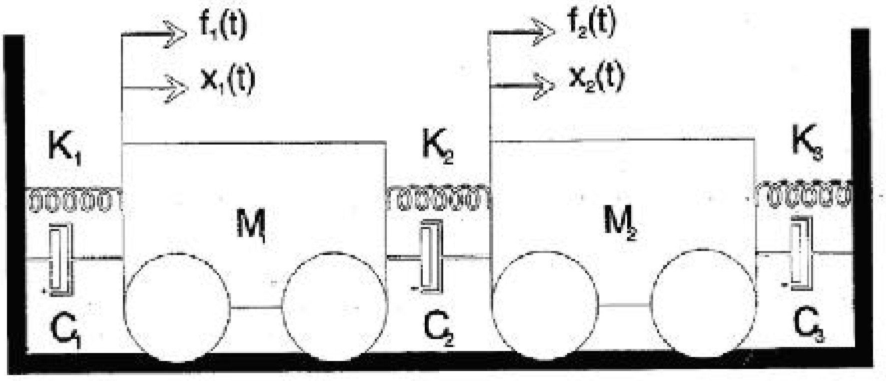
As it can be seen the model is not property estimated. We have to increase the model order.

```
model = armax([Y_out],[4 4]);

damp(model_z);

[Wn_es, Z_es P] = damp(model_z);
[Wn(1)/2/pi Wn_es(3)/2/pi*100;Z(1) Z_es(3)]
```

Given natural frequencies [Hz]	8.7173
Estimated natural frequencies [Hz]	8.7478
Given damping factors	0.0183
Estimated damping factors	0.0199

Exercise 1.

For the system, as shown in Figure below select arbitrary the parameters of the matrices *M*, *C* and *K* and build its model in Simulink for the following parameters.



Exercise 2.

Simulate the system (using the white noise input signal) and store the time histories of the excitation force f_1 and system response x_2.

Exercise 3.

Build the AR and ARMA models of the system from Exercise 1 basing on the data from Exercise 2. Compare obtained results with parameters of the system assumed in Exercise 1.

Exercise 4.

Add a different level of noise to the signals f_1 and x_2. Repeat Exercise 3 for noised data with different levels of noise.