System identification and Signal Filtering
Laboratory work
Identification of a mechanical system

Alban-Félix Barreteau
Master 1 of Control and Robotics - Signal and Image Processing
Ecole Centrale de Nantes
Mail : alban-felix.barreteau@eleves.ec-nantes.fr

January 10, 2021

Professor : Saïd Moussaoui

Abstract

The aim of this project is to identify the continuous-time transfer function of a real system and to link the identified model parameters to some physical parameters of the system. A discrete-time model is firstly identified and then transformed to continuous-time domain.

Please find in appendix the full Matlab code for the laboratory work 2 and 3 about system identifications.

Contents

1 Modeling of the process

1.1 Theorical transfer function of the system

We express the transfer function of the system when considering the velocity as output variable in function of the physical parameters of the process (P, $T1$, $T2$, K_y, K_z) : *(See the calculations at the end of the document)*

$$\frac{Z(s)}{Y_c(s)} = P \times \frac{\frac{Z(s)}{U(s)}}{1+\frac{Y(s)}{U(s)}} = P \times \frac{\frac{k_z \cdot k}{(1+sT_1)(1+sT_2)}}{1+\frac{k_y \cdot k}{s(1+sT_1)(1+sT_2)}} \qquad \text{where} \qquad K_y = k \cdot k_y \quad \text{and} \quad K_z = k \cdot k_z$$

$$= P \times \frac{\frac{K_z}{s^2T_1T_2+s(T_1+T_2)+1}}{1+\frac{K_y}{s(s^2T_1T_2+s(T_1+T_2)+1)}} \times \frac{s(s^2T_1T_2+s(T_1+T_2)+1)}{s(s^2T_1T_2+s(T_1+T_2)+1)}$$

$$= P \times \frac{sK_z}{s(s^2T_1T_2+s(T_1+T_2)+1)+K_y}$$

$$= \frac{PK_zs}{s^3(T_1T_2)+s^2(T_1+T_2)+s+K_y}$$

But the transfer function is normalized on s^3 on Matlab $\frac{Z(s)}{Y_c(s)} = \frac{b_1s+b_0}{s^3+a_2s^2+a_1s+a_0}$ so finally :

$$\frac{Z(s)}{Y_c(s)} = \frac{s\frac{PK_z}{T_1T_2}}{s^3+s^2\frac{T_1+T_2}{T_1T_2}+s\frac{1}{T_1T_2}+\frac{K_y}{T_1T_2}}$$

We identify the physical parameters of the process in function of the coefficients :

$$b_1 = \frac{PK_z}{T_1T_2} \qquad b_0 = 0 \qquad a_2 = \frac{T_1+T_2}{T_1T_2} \qquad a_1 = \frac{1}{T_1T_2} \qquad a_0 = \frac{K_y}{T_1T_2}$$

$$K_z = \frac{b_1}{Pa_1} \qquad K_y = \frac{a_0}{a_1} \qquad \begin{cases} a_1 &= \frac{1}{T_1T_2} \\ a_2 &= \frac{T_1+T_2}{T_1T_2} \end{cases} \Leftrightarrow \begin{cases} T_1 &= \frac{1}{a_1T_2} \\ T_2 &= \frac{a_2 \pm \sqrt{a_2^2-4a_1}}{2a_1} \end{cases}$$

1.2 Identification of the transfer function

I choose to use a OE Output Error model. The orders of the numerator nb and the denominator nf of the discrete transfer function are 3. The delay nk is directly estimated by the Matlab function *delayest* in function of the data. The data is loaded from the *tergane20.mat* file. Once the model constructed, the scipt present several informations about this model.

We identify the parameters a_2, a_1, a_0, b_1 and b_0 from the constructed model, and the physical parameters are therefore calculated.

1.3 Results

I identified the following continuous transfer function $H(s)$ of the processus :

$$H(s) = \frac{10930s}{s^3 + 459,4s^2 + 28930s + 2102000}$$

We deduce the physical parameters of the process :

$$P = 2,91 \qquad K_z = 0,130 \qquad K_y = 72,7 \qquad T_1 = 0,0026 \qquad T_2 = 0,133$$

2 Laboratory work 2

I construct a data object on a BJ Box-Jenkins polynomial model. Then I identified 3 models : ARX, BJ and OE. The informations about those models are presented.

A Lab 3 Matlab code source

```matlab
% Alban-F lix Barreteau, M1 CORO SIP
% Mail : alban-felix.barreteau@eleves.ec-nantes.fr
% Matlab R2020b Update 5 (9.9.0.1538559), Student license
% Signal Filtering and System Identification (SISIF)
% Labwork 3 : Identification of a mechanical system (session 2)

% Professor : said.moussaoui@ec-nantes.fr

%-----------------------------------------------------------------%

clear all
close all
clc

% Definition of the screen :
hpixels=1440;
vpixels=900;


%% Parameters

fs=1000; %Sampling frequency in Hz
ts=1/fs; %Sampling period
P=2.91; %Gain of the proportionnal controller



%% Identification of the processus : Z as output

load tergane20.mat
data=iddata(z,yc,ts);

nb=3;
nf=3;
nk=delayest(data);

mod_oe=oe(data, [nb nf nk]);
present(mod_oe); %Displays model properties
figure(70); resid(data,mod_oe); %Compute and test the residuals associated with identified models
set(figure(70),'name','Residuals associated with identified models','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])
[y,rt2val] = compare(mod_oe, data); %Compare system response with the measured data
fpeval = fpe(mod_oe) ; %Extracts the Final Prediction Error from the model
aicval = aic(mod_oe); %Computes Akaike's Information Criterion for identified models
disp(['RT2 : ',num2str(rt2val),' | AIC : ',num2str(aicval),' | FPE : ',num2str(fpeval)]);


CT_mod_oe=d2c(mod_oe);
present(CT_mod_oe);


%% Transfer function identified

b1=10930;
b0=0;
a3=1;
a2=459.4;
a1=28930;
a0=2102000;



%% Physical parameters of the model

P=2.91
KZ=b1/(P*a1)
```

```
KY=a0/a1
T2=(a2+(a2^2-4*a1)^0.5)/(2*a1)
T1=1/(a1*T2)
```

B Lab 2 Matlab code source

```
% Alban-F lix Barreteau, M1 CORO SIP
% Mail : alban-felix.barreteau@eleves.ec-nantes.fr
% Matlab R2020b Update 5 (9.9.0.1538559), Student license
% Signal Filtering and System Identification (SISIF)
% Labwork 2 : System identification (session 1)

% Professor : said.moussaoui@ec-nantes.fr

%-----------------------------------------------------------------%

clear all
close all
clc

% Definition of the screen :
hpixels=1440;
vpixels=900;

ts=1; %Sampling period


%% Simulation of the system usinf its polynomial form

%Polynomial coefficients
A=[1];
B=[0 0.04];
C=[1 0.1];
D=[1 -1.6 0.64];
F=[1 -1.6 0.64];
sigmanoise=0.1;

mod_exact=idpoly(A,B,C,D,F,sigmanoise,ts) %Constructs a BJ polynomial model


%PRBS, lenght 1024, division factor 10, amplitude [-5 5]
Nr=1024; %Lenght in nb of samples
Nd=10; %Divison factor
Np=2;
N=[Nr-1 1 Np];
band=[]; %Default =[0 1]
a=5; levels=[-a a]; %Input levels in Volts
u=idinput(N,'PRBS',band,levels); %Pseudo random binary signal

figure(10); hold off;
plot(u,'-o'); hold on;
axis([0, length(u)-1, -1.1*a, 1.1*a]);
xlabel('k'); ylabel('u');
set(figure(10),'name','PRBS input','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])


%Simulation of the output signal
option_no_noise=simOptions('AddNoise',false);
ywithoutnoise = sim(mod_exact,u,option_no_noise);

option_noise=simOptions('AddNoise',true);
ywithnoise = sim(mod_exact,u,option_noise);

SNR=20*log10(std(ywithoutnoise)/std(ywithnoise-ywithoutnoise));
disp(['SNR signal noise ratio (dB) = ' num2str(SNR)]);


%QUESTION 6 and 7
data=iddata(ywithnoise,u,ts);
figure(20); hold off;
plot(data); hold on;
set(figure(20),'name','Data: y response to PBRS with noise','position',[hpixels/2 vpixels/2
```

```
    hpixels/2 vpixels/2])
figure(30); hold off;
plot(fft(data)); hold on;
set(figure(30),'name','Fourier transform of data','position',[hpixels/2 vpixels/2 hpixels/2
    vpixels/2])


%% IR of the system from the simulated data

%QESTION1 : identification of the FIR model
na=100;
IR=cra(data,na); %Estimate impulse response
figure(40); hold off;
plot(IR); hold on;
set(figure(40),'name','Estimation of the impulse response','position',[hpixels/2 vpixels/2 hpixels
    /2 vpixels/2])



%% Identification of an ARX polynomial model
na=2;
nb=2;
nk=delayest(data);

mod_arx=arx(data,[na na nk]); %Compute least squares estimate of ARX models
present(mod_arx); %Displays model properties
figure(50); resid(data,mod_arx); %Compute and test the residuals associated with identified models
set(figure(50),'name','Residuals associated with identified models','position',[hpixels/2 vpixels
    /2 hpixels/2 vpixels/2])
[y,rt2val] = compare(mod_arx, data); %Compare system response with the measured data
fpeval = fpe(mod_arx) ; %Extracts the Final Prediction Error from the model
aicval = aic(mod_arx); %Computes Akaike's Information Criterion for identified models
disp(['RT2 : ',num2str(rt2val),' | AIC : ',num2str(aicval),' | FPE : ',num2str(fpeval)]);




%% Identification of an BJ polynomial model

nb=1;
nc=1;
nd=2;
nf=2;
nk=delayest(data);

mod_bj=bj(data,[nb nc nd nf nk]);
present(mod_bj); %Displays model properties
figure(60); resid(data,mod_bj); %Compute and test the residuals associated with identified models
set(figure(60),'name','Residuals associated with identified models','position',[hpixels/2 vpixels
    /2 hpixels/2 vpixels/2])
[y,rt2val] = compare(mod_bj, data); %Compare system response with the measured data
fpeval = fpe(mod_bj) ; %Extracts the Final Prediction Error from the model
aicval = aic(mod_bj); %Computes Akaike's Information Criterion for identified models
disp(['RT2 : ',num2str(rt2val),' | AIC : ',num2str(aicval),' | FPE : ',num2str(fpeval)]);



%% Identification of an OE polynomial model

nb=2;
nf=2;
nk=delayest(data);

mod_oe=oe(data, [nb nf nk]);
present(mod_oe); %Displays model properties
figure(70); resid(data,mod_oe); %Compute and test the residuals associated with identified models
set(figure(70),'name','Residuals associated with identified models','position',[hpixels/2 vpixels
    /2 hpixels/2 vpixels/2])
```

```
[y,rt2val] = compare(mod_oe, data); %Compare system response with the measured data
fpeval = fpe(mod_oe) ; %Extracts the Final Prediction Error from the model
aicval = aic(mod_oe); %Computes Akaike's Information Criterion for identified models
disp(['RT2 : ',num2str(rt2val),' | AIC : ',num2str(aicval),' | FPE : ',num2str(fpeval)]);
```

$$\frac{Z(s)}{Y_c(s)} = P \times \frac{\frac{Z(s)}{U(s)}}{1+\frac{Y(s)}{Z(s)}} = P \times \frac{\frac{k_z . k}{(1+T_1 s)(1+T_2 s)}}{1+\frac{k_y\, k}{s(1+sT_1)(1+sT_2)}}$$

$$= P \times \frac{\frac{K_z}{s^2 T_1 T_2 + s(T_1+T_2)+1}}{1+\frac{K_y}{s\left[s^2 T_1 T_2 + s(T_1+T_2)+1\right]}} \times \frac{s\left[s^2 T_1 T_2 + s(T_1+T_2)+1\right]}{s\left[s^2 T_1 T_2 + s(T_1+T_2)+1\right]}$$

$$= P \times \frac{s K_z}{s\left[s^2 T_1 T_2 + s(T_1+T_2)+1\right] + K_y}$$

$$\frac{Z(s)}{Y_c(s)} = \frac{P K_z s}{s^3 (T_1 T_2) + s^2 (T_1+T_2) + s + K_y} \quad \text{and} \quad \frac{Z(s)}{Y_c(s)} = \frac{b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0}$$

But the transfer function is normalized on s^3 in Matlab.

$$\boxed{\frac{Z(s)}{Y_c(s)} = \frac{\left(\frac{P K_z}{T_1 T_2}\right) s}{s^3 + s^2\left(\frac{T_1+T_2}{T_1 T_2}\right) + s\left(\frac{1}{T_1 T_2}\right) + \frac{K_y}{T_1 T_2}}} \quad \text{and} \quad \frac{Z(s)}{Y_c(s)} = \frac{b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

We identify : $b_1 = \frac{P K_z}{T_1 T_2} \Leftrightarrow \boxed{K_z = \frac{b_1 T_1 T_2}{P} = \frac{b_1}{P a_1}}$

$$a_2 = \frac{T_1+T_2}{T_1 T_2} \qquad a_1 = \frac{1}{T_1 T_2} \qquad a_0 = \frac{K_y}{T_1 T_2} \Leftrightarrow \boxed{K_y = \frac{a_0}{a_1}}$$

$$\begin{cases} a_1 = \frac{1}{T_1 T_2} \\ a_2 = \frac{T_1+T_2}{T_1 T_2} \end{cases} \Leftrightarrow \begin{cases} \boxed{T_1 = \frac{1}{a_1 T_2}} \\ a_2 = \frac{\frac{1}{a_1 T_2} + T_2}{\frac{1}{a_1 T_2} \times T_2} \end{cases} \Leftrightarrow a_2 = \left(\frac{1}{a_1 T_2} + T_2\right) \times a_1 \Leftrightarrow a_2 = \frac{1}{T_2} + a_1 T_2$$

$$\Leftrightarrow a_1 T_2^2 - a_2 T_2 + 1 = 0$$

$$\boxed{T_2 = \frac{a_2 \pm \sqrt{a_2^2 - 4a_1}}{2a_1}}$$

$$\frac{Z(s)}{Y_c(s)} = \frac{b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \quad \text{Continuous time transfer function}$$

$$\frac{a z^2 + b z + c}{d z^3 + e z^2 + f z + g} \times \frac{z^{-3}}{z^{-3}} = \frac{a z^{-1} + b z^{-2} + c z^{-3}}{d + e z^{-1} + f z^{-2} + g z^{-3}} \quad \text{Discrete time transfer function}$$

The discrete numerator and denominator of the transfer function is 3 and 3.

Figure 1: Transfer function of the system with the velocity as output.