System Identification
Project Part 2

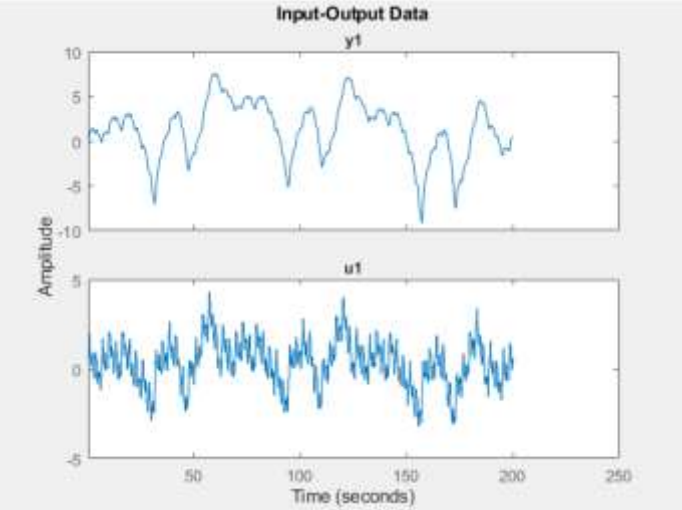
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

A brief description

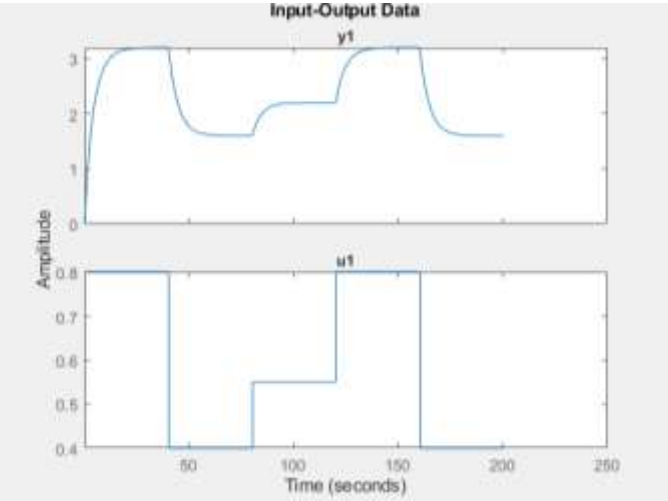
We are given:

A dataset of a dynamic system with one input and one output used for identification and a dataset measured on the same system provided for simulation. We have to develop a black-box model for this system.

Idendification set

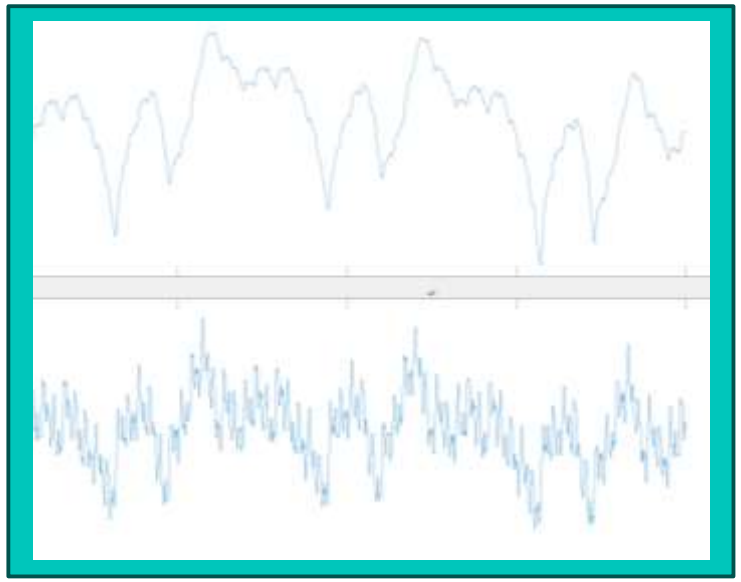


Validation set



In the next slides we will present you how we developed that black-box model.

Understanding the project



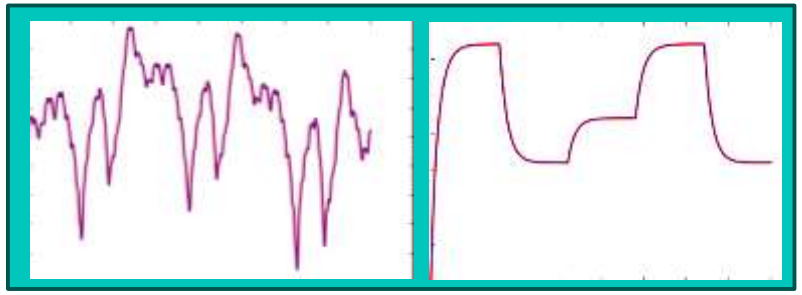
Step 1

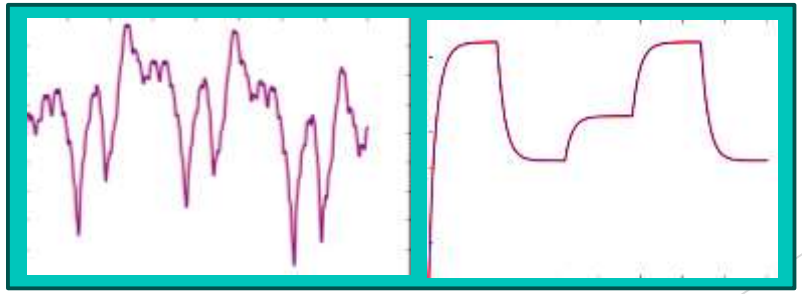
θ

θ

Prediction

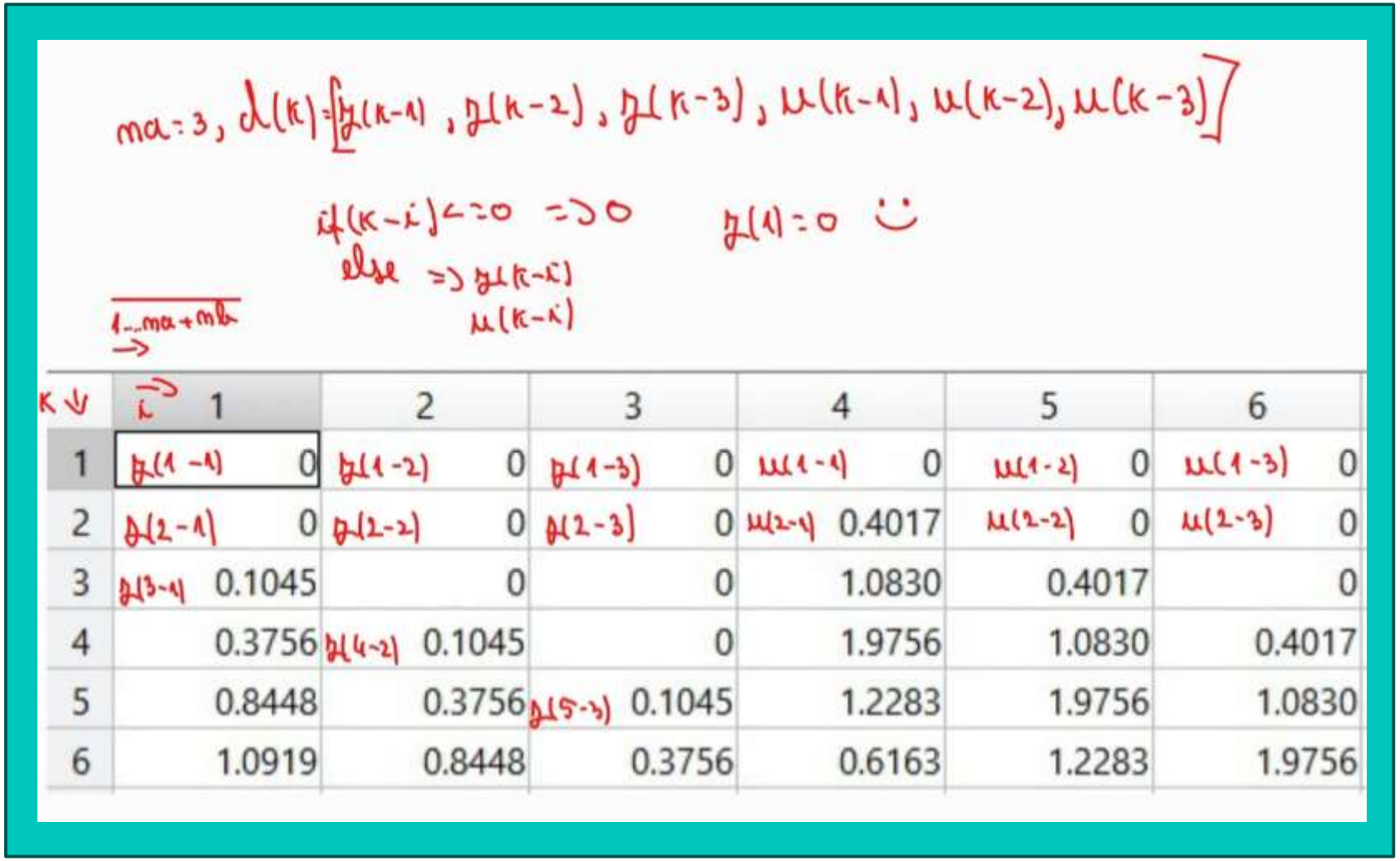
Simulation



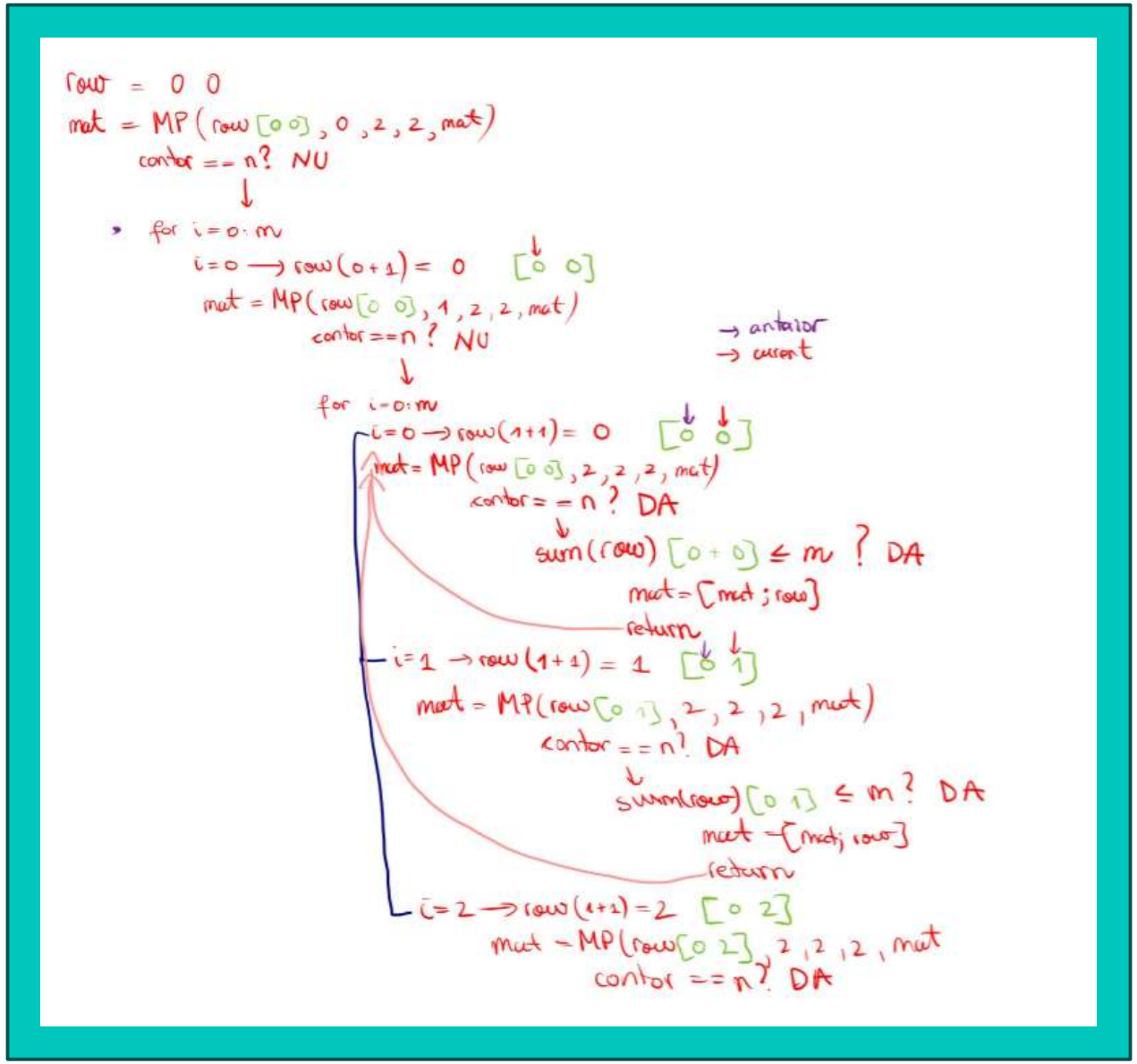


The Functions

Function to create the d(k) matrix



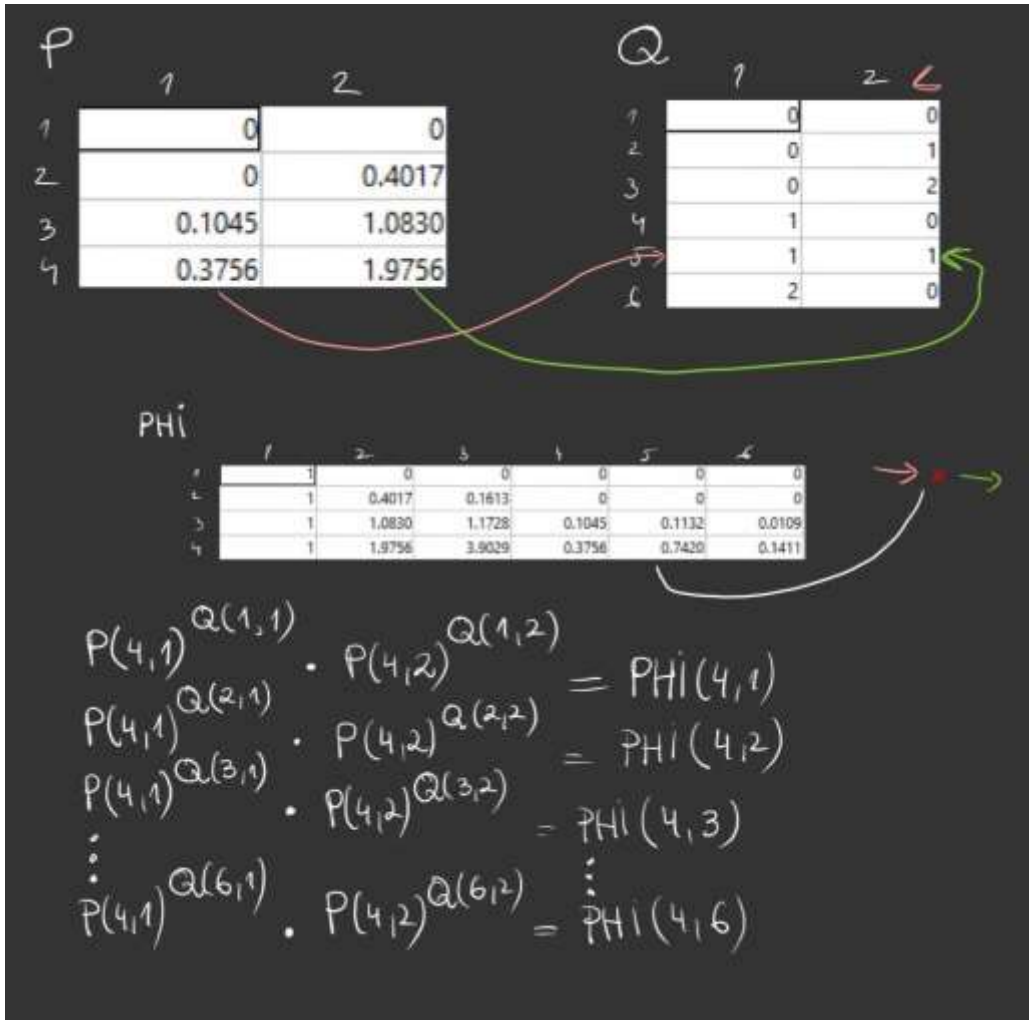
Function to create the matrix of powers



	1	2
1	0	0
2	0	1
3	0	2
4	1	0
5	1	1
6	2	0

Function to create the PHI matrix

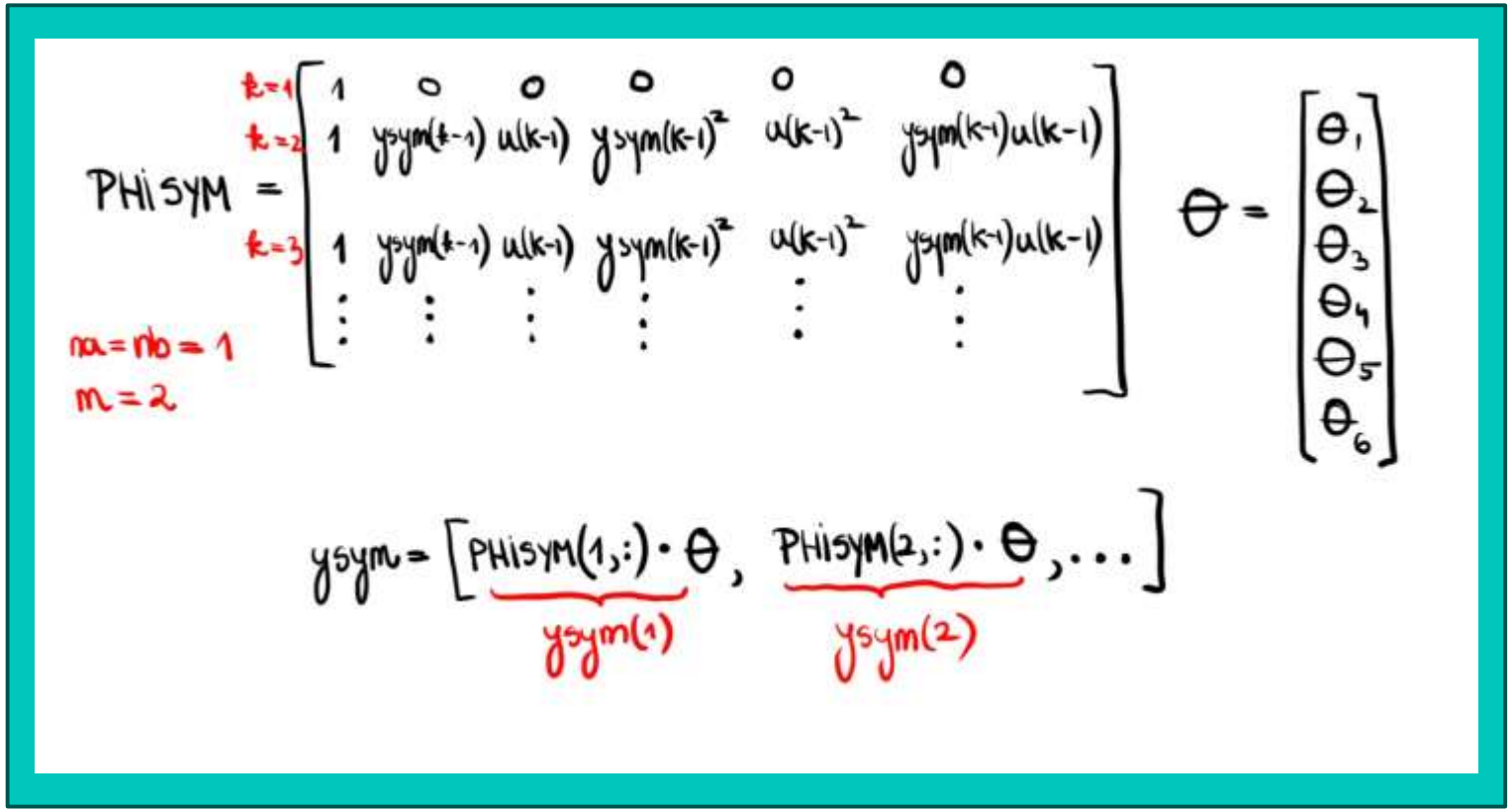
We use the functions that we presented previously



Creating the Simulation

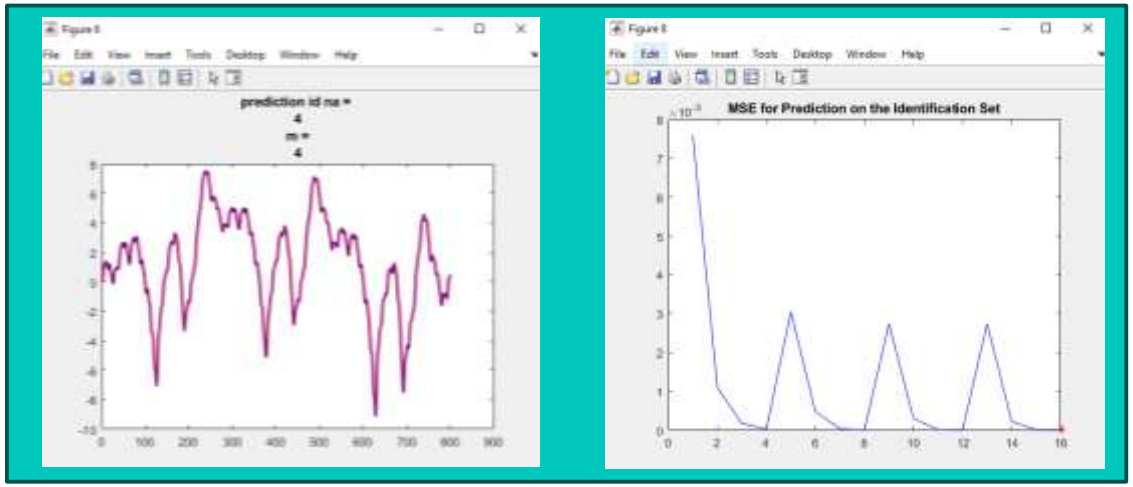
Simulation

We use the same concepts but slightly changed, for the PHI matrix we use the next formula

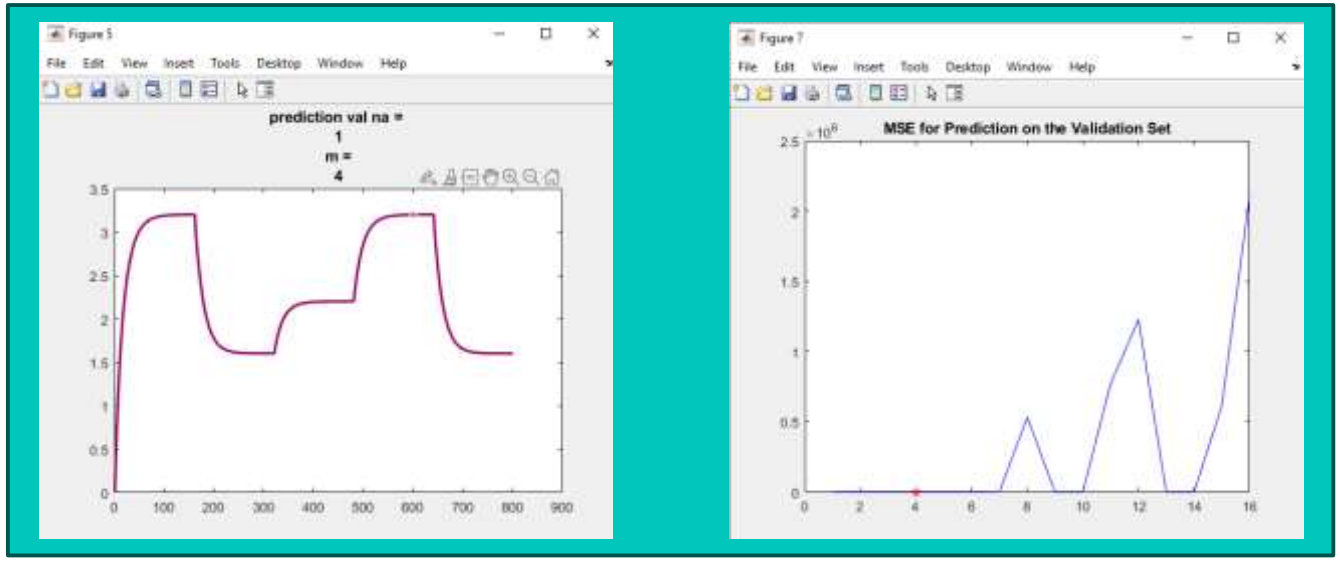


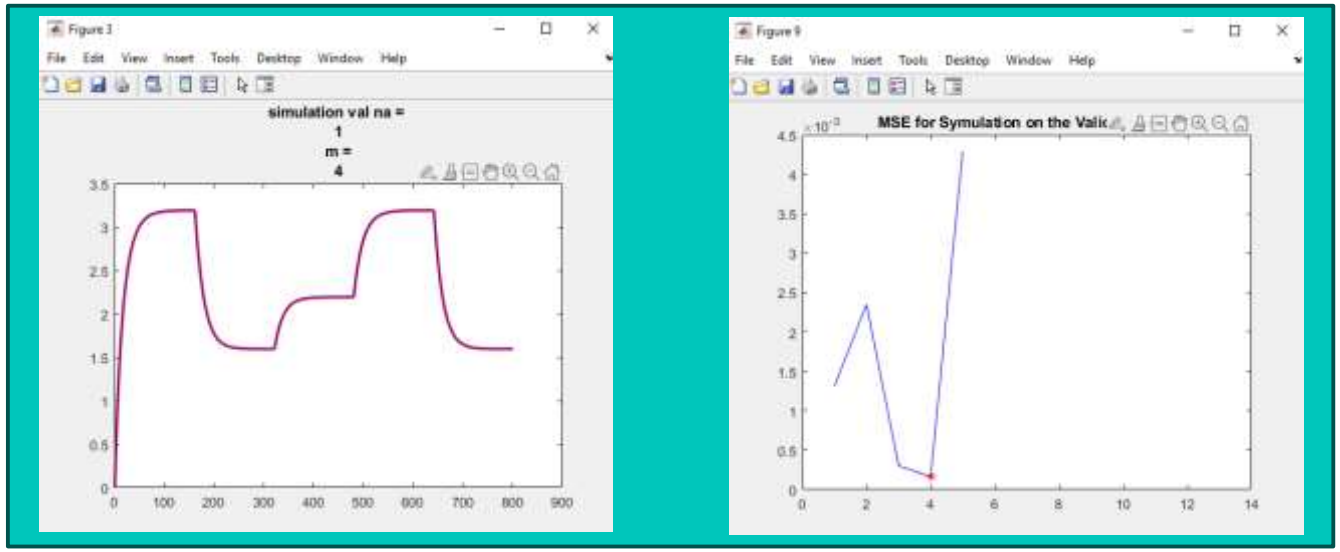
We chose the range for na = nb = 1:4 and degree m = 1:4
The best fits are:

For Identification set:



For Validation set:





Conclusion

- In conclusion, the implementation of the nonlinear ARX model in matlab offers a powerful approach for modeling and predicting systems that have nonlinear behaviours.
- My goal was to write a code that can be understood by everyone and we hope that I did a good job presenting my idea to you.

APPEDIX

```
clear all;
clc;
tic;
% citim datele
load("iddata-08.mat");
uid = id.u;
yid = id.y;
uval = val.u;
yval = val.y;
contor = 1; % contorul pentru MSE
plot(id)
figure
plot(val)
%%
for na = 1:4
for m = 1:4
nb = na;
poliMat = createPoli(uid,yid,na); % creare d(k) id
row = zeros(1, 2*na);
powMat = [];
powMat = createPow(row, 0, 2*na, m, powMat); % creare matrice puter
PHI = createPHI(uid,poliMat,powMat,na,nb); % creare PHI id
theta = PHI \ yid;
poliMat2 = createPoli(uval,yval,na); % creare d(k) val
row2 = zeros(1, 2*na);
powMat2 = [];
powMat2 = createPow(row2, 0, 2*na, m, powMat2); % creare matrice puteri
PHIval = createPHI(uval,poliMat2,powMat2,na,nb); % creare PHI val
yhat = PHIval * theta; % am creat predictia
yhatID = PHI * theta;
N = length(uid);
mse=0;
% calculm mse pentru fiecare na si m
for c=1:N
mse = mse+(yval(c)-yhat(c)).^2;
end
MSEpredict(contor)=1/N * mse;
mse = 0;
for c=1:N
mse = mse+(yid(c)-yhatID(c)).^2;
end
MSEpredictID(contor)=1/N * mse;
```

```
% VAL
if contor == 1
% dam prima valoare lui MinPredict primul MSE
MinPredict = MSEpredict(1);
YhatPredict = yhat;
naPredict = na;
mPredict = m;
% cautam fiecare MSE daca este mai mic decat cele anterioare
% si o salvam
elseif MSEpredict(contor)<MinPredict
MinPredict = MSEpredict(contor);
YhatPredict = yhat;
naPredict = na;
mPredict = m;
end
% ID
if contor == 1
% dam prima valoare lui MinPredict primul MSE
MinPredictID = MSEpredictID(1);
YhatPredictID = yhatID;
naPredictID = na;
mPredictID = m;
% cautam fiecare MSE daca este mai mic decat cele anterioare
% si o salvam
elseif MSEpredictID(contor)<MinPredictID
MinPredictID = MSEpredictID(contor);
YhatPredictID = yhatID;
naPredictID = na;
mPredictID = m;
end
% SIMULARE
% Val
PHISYM = [];
row = zeros(1, 2*na);
powMatSYM = [];
powMatSYM = createPow(row, 0, 2*na, m, powMatSYM); % cream matricea de puteri
term = [];
```

```
for k =1:length(uval)
for i = 1:na
j = i+na;
% verificam daca k-i este mai mic sau egal decat 0 si in
% functie de asta term va lua valoari 0 sau ysim / uval
if k-i <= 0
term(i) = 0;
term(j) = 0;
else
term(i) = ysim(k-i);
term(j) = uval(k-i);
end
end
% cream PHI SYM inmultind fiecare valoare din term cu
for j = 1:length(powMatSYM)
phi = 1;
for i = 1:na+nb
termen = term(i)^powMatSYM(j,i); % ridicam prima valoare din
% term la prima valoare din powmat si o punem in phi apoi urmatoarea
% valoare din term o ridicam la urmatoarea valoare din
% powmat si o inmultim cu valoarea anteriora astfel
% obtinem prima valoare din PHI
phi = phi*termen;
end
PHISYM(k,j) = phi;
end
% ysim devine linia anterioara din PHI inmultita cu Theta care
% ne va ajuta in linia 72
ysim(k) = PHISYM(k,:) * theta;
end
% ID
PHISYMID = [];
row = zeros(1, 2*na);
powMatSYMID = [];
powMatSYMID = createPow(row, 0, 2*na, m, powMatSYMID); % cream matricea de puteri
term = [];
```

```
for k =1:length(uid)
for i = 1:na
j = i+na;
% verificam daca k-i este mai mic sau egal decat 0 si in
% functie de asta term va lua valoari 0 sau ysim / uval
if k-i <= 0
term(i) = 0;
term(j) = 0;
else
term(i) = ysimID(k-i);
term(j) = uid(k-i);
end
end
% cream PHI SYM inmultind fiecare valoare din term cu
for j = 1:length(powMatSYMID)
phi = 1;
for i = 1:na+nb
termen = term(i)^powMatSYMID(j,i); % ridicam prima valoare din
% term la prima valoare din powmat si o punem in phi apoi urmatoarea
% valoare din term o ridicam la urmatoarea valoare din
% powmat si o inmultim cu valoarea anteriora astfel
% obtinem prima valoare din PHI
phi = phi*termen;
end
PHISYMID(k,j) = phi;
end
% ysim devine linia anterioara din PHI inmultita cu Theta care
% ne va ajuta in linia 72
ysimID(k) = PHISYMID(k,:) * theta;
end
% MSE val
N = length(uval);
mse=0;
for c=1:N
mse = mse+(yval(c)-ysim(c)).^2;
end
MSEsym(contor)=1/N * mse;
```

```
if contor == 1
MinSym = MSEsym(1);
YhatSym = ysim;
naSym = na;
mSym = m;
elseif MSEsym(contor)<MinSym
MinSym = MSEsym(contor);
YhatSym = ysim;
naSym = na;
mSym = m;
end
% MSE ID
N = length(uid);
mse=0;
for c=1:N
mse = mse+(yid(c)-ysimID(c)).^2;
end
MSEsymID(contor)=1/N * mse;
if contor == 1
MinSymID = MSEsymID(1);
YhatSymID = ysimID;
naSymID = na;
mSymID = m;
elseif MSEsymID(contor)<MinSymID
MinSymID = MSEsymID(contor);
YhatSymID = ysimID;
naSymID = na;
mSymID = m;
end
contor = contor+1;
end
end
%Simualre Validare
figure
plot(YhatSym,"r", 'LineWidth', 2);
hold on
plot(yval,"b")
title(["simulation val na = " num2str(naSym), " m = " num2str(mSym)])
% Simulare Identificare
figure
plot(YhatSymID,"r", 'LineWidth', 2);
hold on
plot(yid,"b")
title(["simulation id na = " num2str(naSymID), " m = " num2str(mSymID)])
% Predictie validare
```

```
figure
plot(YhatPredict,"r", 'LineWidth', 2);
hold on
plot(yval,"b")
title(["prediction val na = " num2str(naPredict), " m = " num2str(mPredict)])
% Predictie identificare
figure
plot(YhatPredictID,"r", 'LineWidth', 2);
hold on
plot(yid,"b")
title(["prediction id na = " num2str(naPredictID), " m = " num2str(mPredictID)])
% MSE plots
% MSE predict val
figure
plot(MSEpredict,"b");
hold on;
plot(mPredict*naPredict, MinPredict, "r*");
title("MSE for Prediction on the Validation Set");
% MSE predict ID
figure
plot(MSEpredictID,"b");
hold on;
plot(mPredictID*naPredictID, MinPredict, "r*");
title("MSE for Prediction on the Identification Set");
% MSE sym val
figure
plot(MSEsym,"b");
hold on;
plot(mSym*naSym, MinSym, "r*");
title("MSE for Symulation on the Validation Set");
% MSE sym ID
figure
plot(MSEsymID,"b");
hold on;
plot(mSymID*naSymID, MinSymID, "r*");
title("MSE for Symulation on the Identification Set");
toc;
```

```
%% FUNCTII
% scoatem polinomu in functie de na si nb
function poliMat = createPoli(u,y,na)
poliMat = [];
for k =1:length(u)
for i = 1:na
j = i+na;
% verificam daca k-i este mai mic sau egal decat 0 si in
% functie de asta term va lua valoari 0 sau ysim / uval
if (k-i)<=0
term(i) = 0;
term(j) = 0;
else
term(i) = y(k-i);
term(j) = u(k-i);
end
end
% in final vom avea o matrice cu linile D(k) in functie de na si nb
poliMat = [poliMat;term];
end
end
function powMat = createPow(row, contor, n, m, powMat)
% verifiva daca sunt n elemente pe linia de matrice de putere (n=nb+na)
if contor == n
% verificam daca suma valorilor de pe linie <= decat gradul la care
% ridicam
if sum(row) <= m
powMat = [powMat; row];
end
return;
end
% daca contorul nu e egal cu n, de face recursivitatea
for i = 0:m
row(contor + 1) = i;
powMat = createPow(row, contor + 1, n, m, powMat);
end
end
function PHI = createPHI(uid,poliMat,powMat,na,nb)
for k = 1:length(uid)
for j = 1:length(powMat)
phi = 1;
for i = 1:na+nb
% termen este prima valoare din poli mat ridicata la prima
% valoare din powMat apoi urmatoarea valoare din polimat
% ridicata la urmatoarea valoare din powmat apoi aceste
% valori sunt inmultite sunt prima valoare din PHI
termen = poliMat(k,i)^powMat(j,i);
phi = phi*termen;
end
PHI(k,j) = phi;
end
end
end
```