Nonlinear ARX identification

Project Assignment -part 2

System Identification

2023-2024

- Chis Dalina

- Index: 21

PROBLEM STATEMENT

The central task is to devise a black-box model for a system using a polynomial, nonlinear ARX model. The assignment involves coding a function capable of generating this ARX model, with customizable model orders *na* and *nb*, polynomial degree *m*, and a simplified delay *nk=1*.

It is imperative that the model remains versatile, operating seamlessly in both one-step-ahead prediction and simulation modes.

APPROXIMATOR STRUCTURE

The vector of delayed outputs and inputs, denoted by d(k), is defined as:

$$\mathrm{d}(k) = [y(k-1), y(k-2), \ldots, y[k-na], u(k-nk), u(k-nk-1), \ldots, u(k-nk-nb+1)]$$

where na and *nb* represents the model orders and nk represents the delay.

Next, the NARX model is denoted as:

$$\hat{y} = \mathrm{p}(\mathrm{d}(\mathrm{k}))$$

For example, if *na=nb=nk=1* and *m=2*, the Nonlinear ARX model will be:

$$\hat{y} = y(k-1) + u(k-1) + y(k-1)^2 + u(k-1)^2 + u(k-1)y(k-1) + 1$$

The parameter vector θ can be extracted using left matrix division:

$$\theta = \hat{y} \backslash y\_id$$

,where y_id is the identification output signal.

The polynomial *y(k):*

$$y(k) = \hat{y} * \theta$$

Explicitly, for the example we had before, the polynomial will have the form:

$$y(k) = ay(k-1) + bu(k-1) + cy(k-1)^2 + vu(k-1)^2 + wu(k-1)y(k-1) + z$$

To achieve the simulation mode, I will go through the previous steps again, using this time only the previous outputs of the model itself, because the real outputs are not available.
For *na=nb=nk=1* and *m=2*, the NARX simulation mode will be:

$$\tilde{y}(k) = a\tilde{y}(k-1) + bu(k-1) + c\tilde{y}(k-1)^2 + vu(k-1)^2 + wu(k-1)\tilde{y}(k-1) + z$$

After the models computation, I can calculate the MSE for prediction:

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{y}-y)^2$$

and the MSE for simulation:

$$\frac{1}{N}\sum_{i=1}^{N}(\tilde{y}-y)^2$$

Where N represents the number of parameters in the output signal y, $\hat{y}$ represents the one-step ahead prediction, $\tilde{y}$ represents the simulation.

MSE PREDICTION

mn	a	nb	MSE-IDENTIFICATION
11	12	1	0.0141713718642342
		2	1.62569440498264e-05
1	3	3	1.50981744439272e-06
1	4	4	1.20585075849691e-06
1	5	5	1.18959618586102e-06
1	6	6	1.18916103947902e-06
1	7	7	1.18277217189443e-06
2	1	1	0.0141181120399836
2	2	2	0.00722776580612515
2	3	3	0.00718275822569897
2	4	4	0.00701111305173208
2	5	5	8.90538186691477e-07
2	6	6	7.91431658142869e-07
2	7	7	7.54115465891915e-07
3	1	1	0.0140490086656949
3	2	2	0.0069022561345800
3	3	3	4
3	4	4	0.0065425243310841
3	5	5	8 3.57 115150534105e-07
3	6	6	0.00589783287141167 2.47755532295424e-07
3	7	7	8 1.64329824235700e-07

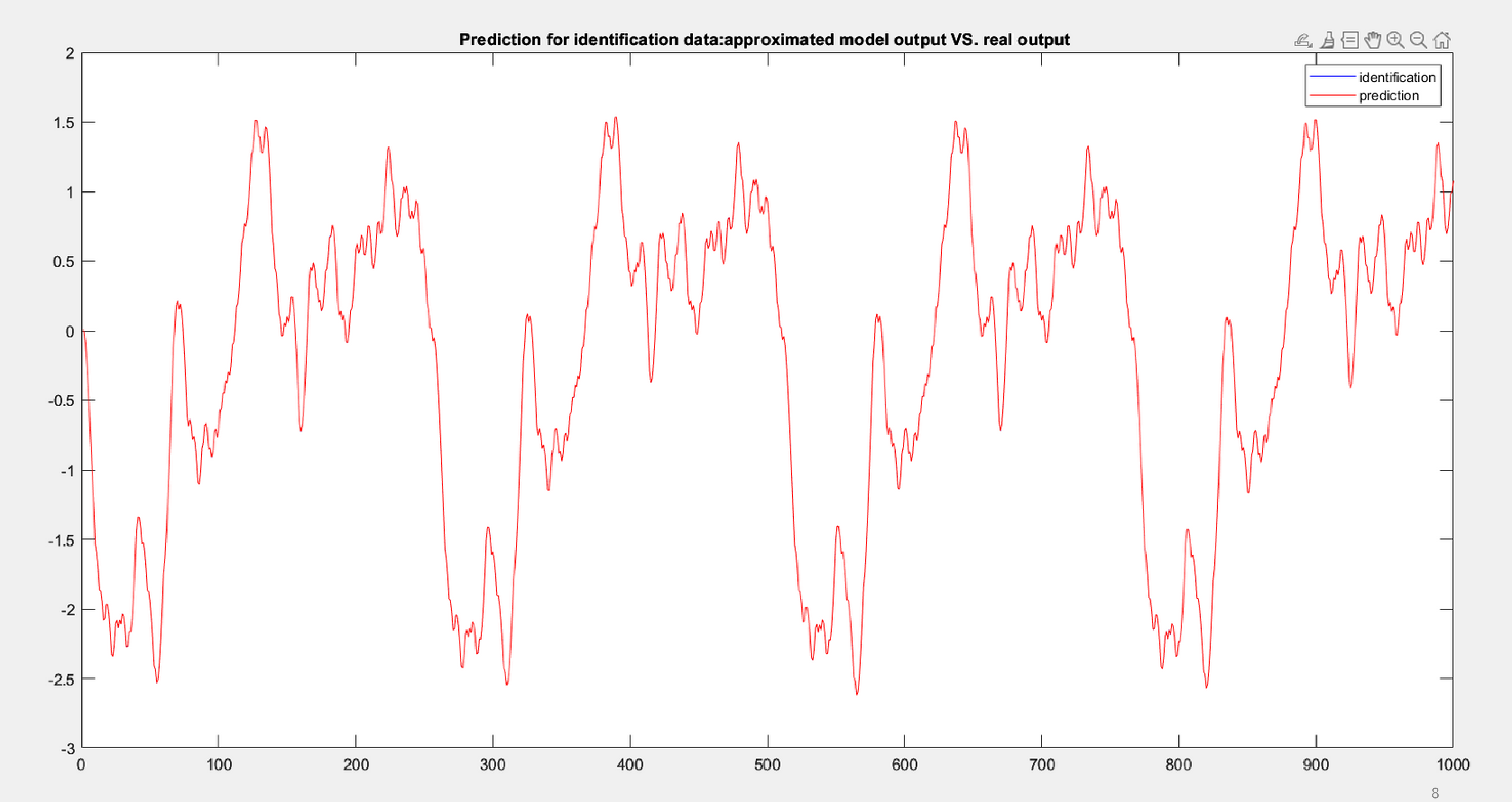
mn	a	nb	MSE-VALIDATION
11	12	1	0.0143781841015675
		2	5.98344842904510e-05
1	3	3	2.02443163647362e-06
1	4	4	2.05410477555564e-06
1	5	5	2.06078038902150e-06
1	6	6	2.05674588031488e-06
1	7	7	2.03974669479565e-06
2	1	1	0.0144815216102294
2	2	2	0.00741761240387187
2	3	3	0.00853599405122706
2	4	4	0.0265864103990573
2	5	5	2.62062652199178e-06
2	6	6	3.11456373864799e-06
2	7	7	8.95593515057902e-06
3	1	1	0.017605979383271
3	2	2	8
3	3	3	0.012745626369155
3	4	4	4
3	5	5	0.0274892506472031 5047.99266072246
3	6	6	48150.2541202675 29 84784704.72063187
3	7	7	1649515956.62714

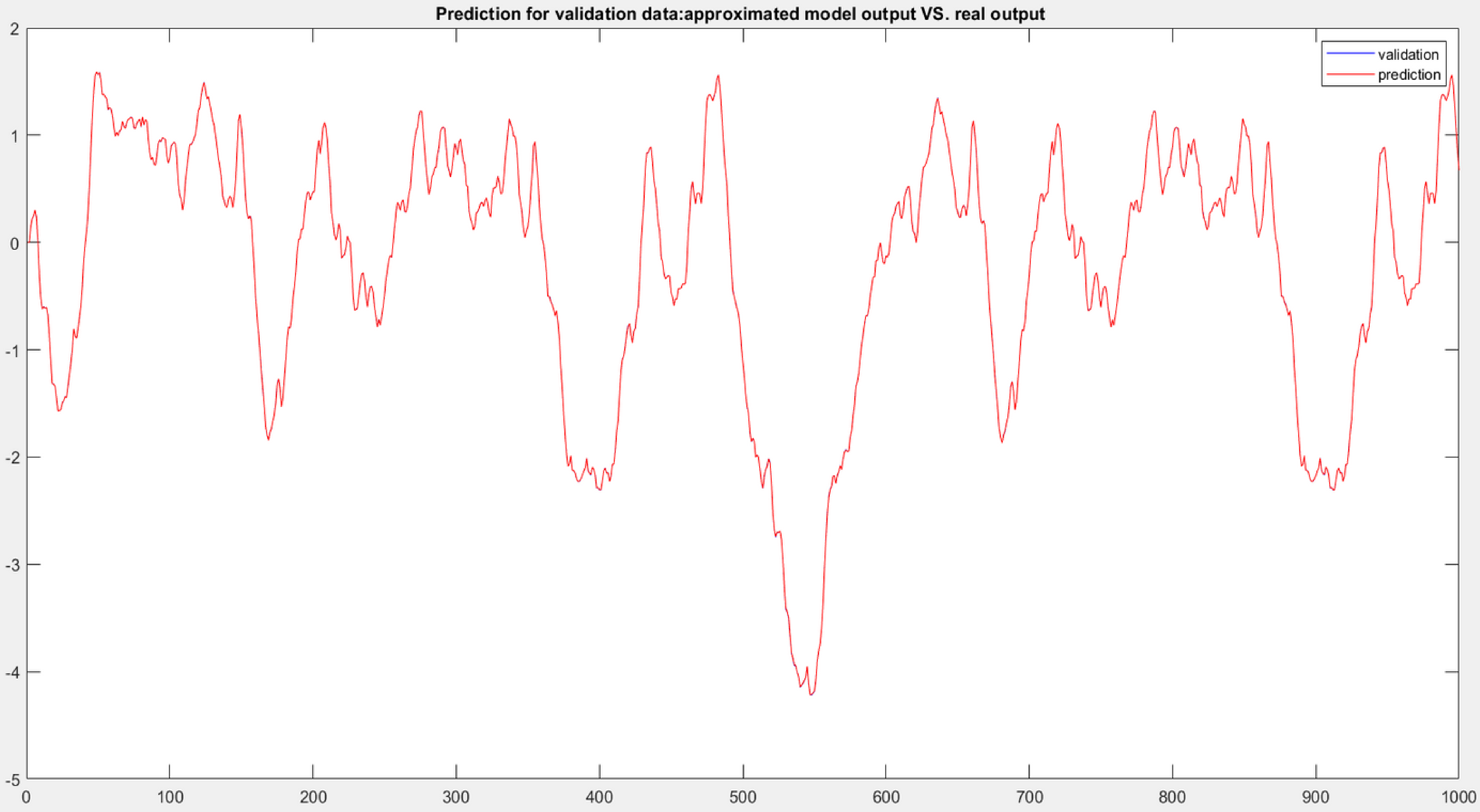
MSE SIMULATION

m	na	nb	MSE-IDENTIFICATION
1	1	1	1.29635596300024
1	2	2	0.0481754947052744
1	3	3	0.0465180717460251
1	4	4	0.0473487142077035
1	5	5	0.0488070624426243
1	6	6	0.0489728189741069
1	7	7	0.0497982618027379
2	1	1	1.47286812528404
2	2	2	52.6925653902392
2	3	3	NaN
2	4	4	NaN
2	5	5	1.11352177443568
2	6	6	0.142496854563857
2	7	7	0.0271134774232392
3	1	1	2.00198038967236
3	2	2	NaN
3	3	3	NaN
3	4	4	NaN
3	5	5	NaN
3	6	6	NaN
3	7	7	NaN

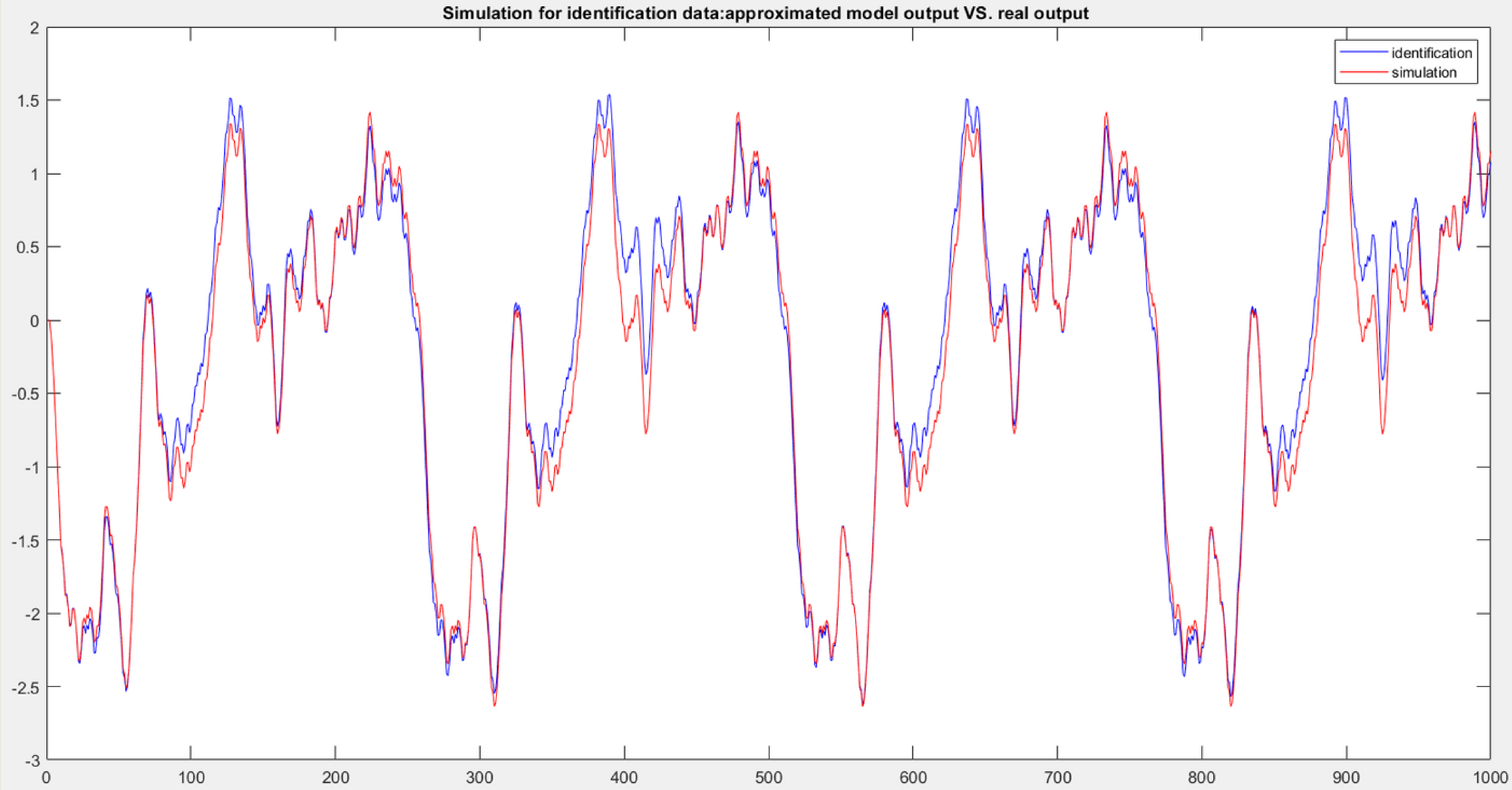
m	na	nb	MSE-VALIDATION
1	1	1	1.54396403409430
1	2	2	0.307122279979304
1	3	3	0.303187944833722
1	4	4	0.306422602897989
1	5	5	0.305089187160046
1	6	6	0.304845228114266
1	7	7	0.303782683174513
2	1	1	1.79488833348989
2	2	2	52.7143572788075
2	3	3	NaN
2	4	4	NaN
2	5	5	1.69228527915042
2	6	6	0.117668766316217
2	7	7	NaN
3	1	1	2.41705777278304
3	2	2	NaN
3	3	3	NaN
3	4	4	NaN
3	5	5	NaN
3	6	6	NaN
3	7	7	NaN

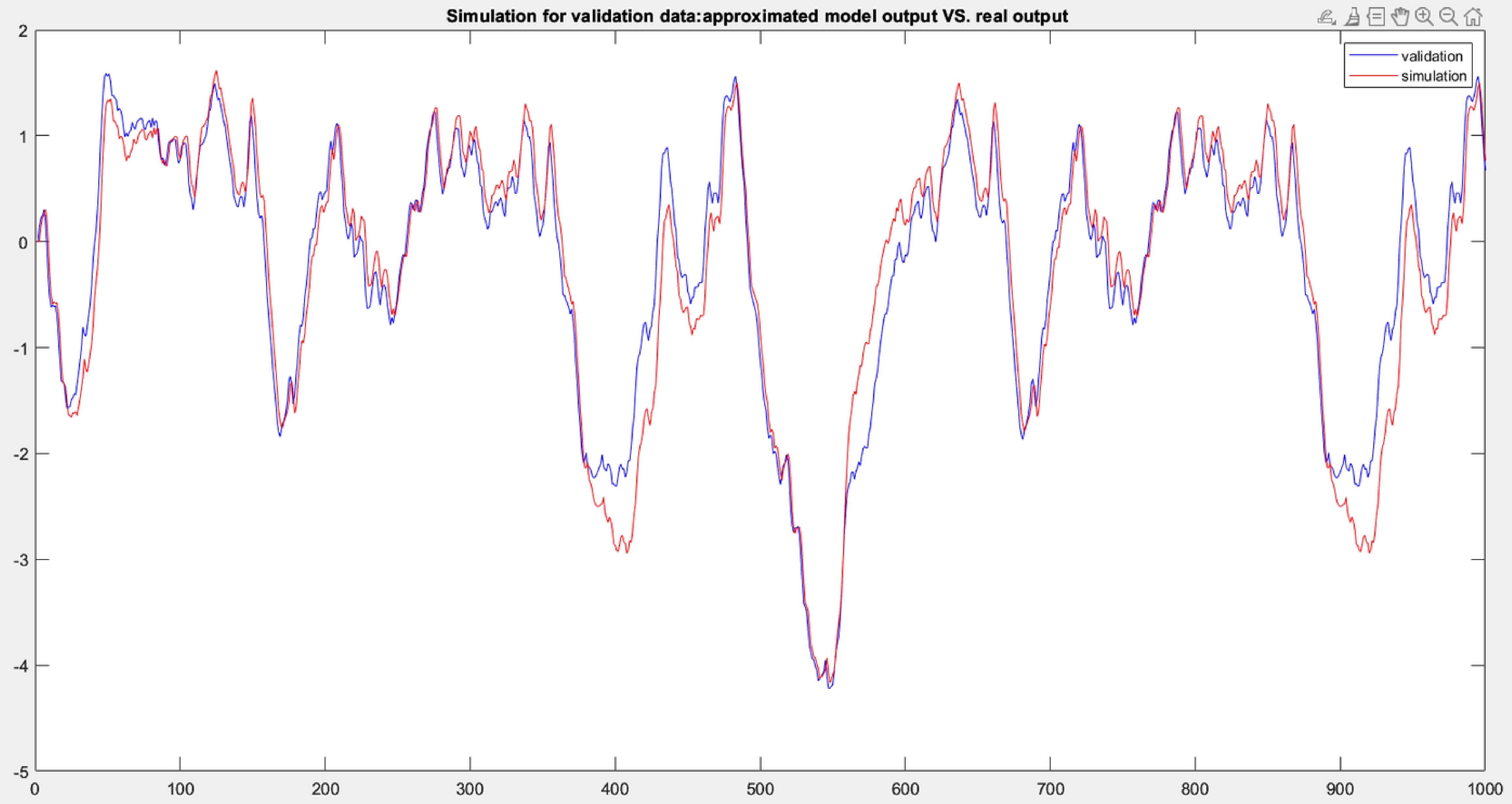
TUNING RESULTS





Prediction for validation data:approximated model output VS. real output
validation
prediction
2
1
0
-1
-2
-3
-4
-5
0
100
200
300
400
500
600
700
800
900
1000


Simulation for identification data:approximated model output VS. real output
identification
simulation
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
100
200
300
400
500
600
700
800
900
1000


Simulation for validation data:approximated model output VS. real output
validation
simulation
2
1
0
-1
-2
-3
-4
-5
0
100
200
300
400
500
600
700
800
900
1000

THANK YOU FOR YOUR ATTENTION

Code

```
clear load("iddata-
21.mat");

% Variable initialization
y_id=id.y;
u_id=id.u;
Ts_id=id.Ts;

y_val= val.y;
u_val= val.u;
Ts_val= val.Ts;

len_y_id=length(y_id);
len_u_id=length(u_id);
len_y_val=length(y_val)
;
len_u_val=length(u_val);

m_final_pred_id= 0;
m_final_pred_val= 0;
m_final_sim_id= 0;
m_final_sim_val= 0;
```

```
min_mse_pred_id= 5000;
min_mse_pred_val= 5000;
min_mse_sim_id= 5000;
min_mse_sim_val= 5000;


y_pred_id_final= zeros(len_y_id, 1);
y_pred_val_final= zeros(len_y_val, 1);
y_sim_id_final= zeros(len_y_id, 1);
y_sim_val_final= zeros(len_y_id, 1);


na_final_pred_id= 0;
nb_final_pred_id= 0;
na_final_pred_val= 0;
nb_final_pred_val= 0;
na_final_sim_id= 0;
nb_final_sim_id= 0;
na_final_sim_val= 0;
nb_final_sim_val= 0;


aux = 1;
nk=1;
```

```
for m = 1:3
for na= 1:7
nb= na;

% powers matrix
nr_rows= 0;
powers = zeros(1000, na+nb);
i= 1;
active = 1;
while (active)
powers(i+1,:) = powers(i,:);
if sum(powers(i,:)) + 1 <= m
powers(i+1, na+nb) = powers(i+1, na+nb) + 1;
else
for k = length(powers(i,:)):-1:1
ifpowers(i,k) > 0
powers(i+1,:) = powers(i,:);
powers(i+1, k) = 0;
if(k -1) > 0
                    powers(i+1, k -1) = powers(i+1, k -1) + 1;
else
active = 0;
end
break
end
end
end
i= i+ 1;
nr_rows= nr_rows+ 1;
end

    powers = powers(1:nr_rows, :);
```

```
% PHI_id
for k=1:len_y_id
for j=1:na
if k-j<=0
phi_y(k,j)=y_id(1);
else
phi_ y(k,j)=y_id(k-j);
end
end
end
for k=1:len_u_id
for j=1:nb
if (k-nk-j+1)<=0
phi_u(k,j)=0;
else
        phi_u(k,j)=u_id(k-nk-j+1);
end
end
end
    PHI_id=cat(2, phi_y, phi_u);
```

```
% PHI_val
for k=1:len_y_val
for j=1:na
if k-j<=0
phi_ y(k,j)=y_val(1);
else
                    phi_ y(k,j)=y_val(k-j);
end
end
end
for k=1:len_u_val
for j=1:nb
             if (k-nk-j+1)<=0
phi_u(k,j)=0;
else
                 phi_u(k,j)=u_val(k-nk-j+1);
end
end
end
         PHI_val=cat(2, phi_y, phi_u);
```

```
% y_model_id
y_model_id= zeros(len_y_id, nr_rows);
for k= 1:len_y_id
fori= 1:nr_rows
element = 1;
for j = 1:na+nb
                 element = element*(PHI_id(k,j) ^ powers(i,j));
end
y_model_id(k,i) = element;
end
end
% theta
theta = y_model_id\y_id;
% y_model_val
y_model_val= zeros(len_y_val, nr_rows);
for k = 1:len_y_val
fori= 1:nr_rows
element = 1;
for j = 1:na+nb
                element = element*(PHI_val(k,j) ^ powers(i,j));
end
y_model_val(k,i) = element;
end
end
```

```
% Prediction on identification data
y_pred_id= zeros(len_y_id, 1);
for k = 1:len_y_id
        y_pred_id(k) = y_model_id(k,:)*theta;
    end

    % MSE Prediction id
    mse_pred_id=       1/len_y_id.*sum((y_pred_id-y_id).^2);
    mse_matrix_pred_id(aux, :) = [m nanbmse_pred_id];

    ifmse_pred_id< min_mse_pred_id
        min_mse_pred_id= mse_pred_id;
    y_pred_id_final= y_pred_id;
    m_final_pred_id= m;
    na_final_pred_id= na;
    nb_final_pred_id= nb;
    end
```

```
% Prediction on validation data
y_pred_val= zeros(len_y_val, 1);
for k = 1:len_y_val
        y_pred_val(k) = y_model_val(k,:)*theta;
    end

    % MSE Prediction val
    mse_pred_val=       1/len_y_val.*sum((y_pred_val-y_val).^2);
    mse_matrix_pred_val(aux, :) = [m nanbmse_pred_val];

    ifmse_pred_val< min_mse_pred_val
        min_mse_pred_val= mse_pred_val;
    y_pred_val_final= y_pred_val;
    m_final_pred_val= m;
    na_final_pred_val= na;
    nb_final_pred_val= nb;
    end
```

```
% Simulation: identification
PHI_sim_id= zeros(len_y_id, na+nb);
	y_model_sim_id= zeros(len_y_id, nr_rows);
ysim_id= zeros(len_y_id, 1);

for k=1:len_y_id
forj = 1:na+nb
if j <= na
if k-j<=0
PHI_sim_id(k,j) = y_id(1);
else
PHI_sim_id(k,j) = ysim_id(k-j);
end
elseif j > na
if k-(j-na-nk+1)<=0
PHI_sim_id(k,j) = 0;
else
				PHI_sim_id(k,j) = u_id(k-(j-na-nk+1));
end
end
end
```

```
for i= 1:nr_rows
element = 1;
forj = 1:na+nb
                element = element*(PHI_sim_id(k,j) ^ powers(i,j));
end
y_model_sim_id(k,i) = element;
end
ysim_id(k) = y_model_sim_id(k,:)*theta;
end

% MSE

mse_sim_id= 1/len_y_id.*sum((ysim_id-y_id).^2);
       mse_matrix_sim_id(aux, :) = [m nanbmse_sim_id];

      if mse_sim_id< min_mse_sim_id
         min_mse_sim_id= mse_sim_id;
         y_sim_id_final= ysim_id;
         m_final_sim_id= m;
         na_final_sim_id= na;
         nb_final_sim_id= nb;
      end
```

```
% Simulation: validation
y_model_sim_val= zeros(len_y_val, nr_rows);
ysim_val= zeros(len_y_val, 1);
PHI_sim_val= zeros(len_y_val, na+nb);

for k=1:len_y_val
for j = 1:na+nb
ifj <= na
ifk-j > 0
PHI_sim_val(k,j) = ysim_val(k-j);
else
PHI_sim_val(k,j) = y_val(1);
end
elseif j > na
ifk-(j-na-nk+1) > 0
                    PHI_sim_val(k,j) = u_val(k-(j-na-nk+1));
else
PHI_sim_val(k,j) = 0;
end
end
end
```

```
fori= 1:nr_rows
element = 1;
forj = 1:na+nb
                    element = element*(PHI_sim_val(k,j) ^ powers(i,j));
end
y_model_sim_val(k,i) = element;
end
ysim_val(k) = y_model_sim_val(k,:)*theta;
end
% MSE
         mse_sim_val= 1/len_y_id.*sum((ysim_val-y_val).^2);
        mse_matrix_sim_val(aux, :) = [m nanbmse_sim_val];
ifmse_sim_val< min_mse_sim_val
min_mse_sim_val= mse_sim_val;
y_sim_val_final= ysim_val;
m_final_sim_val= m;
na_final_sim_val= na;
nb_final_sim_val= nb;
end
aux = aux + 1;
end
end
```

```
% Plots
figure;
plot(y_id, 'b'); hold on;
plot(y_pred_id_final,'r' );
title('Prediction for identification data:approximatedmodel output VS. real output
'); legend('identification', 'prediction')

figure;
plot(y_val, 'b'); hold on;
plot(y_pred_val_final, 'r');
title('Prediction for validation data:approximatedmodel output VS. real output ');
legend('validation', 'prediction')

figure;
plot(y_id,'b'); hold on;
plot(y_sim_id_final,'r');
title('Simulation for identification data:approximatedmodel output VS. real output
');
legend('identification', 'simulation')
```

```
figure;
plot(y_val,'b'); hold on;
plot(y_sim_val_final,'r');
title('Simulation for validation data:approximatedmodel output VS. real output
'); legend('validation', 'simulation')

figure;
plot(mse_matrix_pred_id(:,4),'r');
title('Identification Prediction Errors’)

figure;
plot(mse_matrix_pred_val(:,4),'r');
title('Validation Prediction Errors’)

figure;
plot(mse_matrix_sim_id(:,4),'b');
title('Identification Simulation Errors’)

figure;
plot(mse_matrix_sim_val(:,4),'b');
title('Validation Simulation
Errors')
```