Amirkabir University of Technology
(Tehran Polytechnic)

Electrical Engineering Faculty

Control Department

MSc Program

Assignment 04

through

System Identification

Course

by

Mohammad Azimi - 402123100

mohammadazimi2000@aut.ac.ir

Course Lecturer

Dr. Mehdi Karrari

Question 1

```
warning off; close all; clc;
% Having saved the unit step response of the system from the Simulink Run,

fprintf(">>> Having saved the unit step response of the system from the Simulink Run,\n")
```

```
>>> Having saved the unit step response of the system from the Simulink Run,
```

```
fprintf("    the data could fetched via:\n")
```

```
   the data could fetched via:
```

```
fprintf("--------------------------------------------------------------------------------")
```

```
--------------------------------------------------------------------------------
```

```
fprintf("                            stepResp = step_resp.Data;\n")
```

```
                               stepResp = step_resp.Data;
```

```
fprintf("--------------------------------------------------------------------------------")
```

```
--------------------------------------------------------------------------------
```

```
stepResp = step_resp.Data;
```

>>> 1.1 - Proper Sampling Time

```
fprintf(">>> The proper sample time for the available system could be obtained by calculating\n")
```

```
>>> The proper sample time for the available system could be obtained by calculating
```

```
fprintf("    the area between the step response of the system and the constant line with the")
```

```
   the area between the step response of the system and the constant line with the
```

```
fprintf("    value of steady state response.")
```

```
   value of steady state response.
```

```
ys = mean(stepResp(floor(length(stepResp))/10:end));

summation = 0;
for index=1:length(stepResp)
    summation = summation + (ys - stepResp(index));
end
summation = summation/ys;

proper_Ts = summation/10;

fprintf("    Hence, the proper sampling time (Ts) = 2")
```

```
   Hence, the proper sampling time (Ts) = 2
```

```
proper_Ts = 2;
```

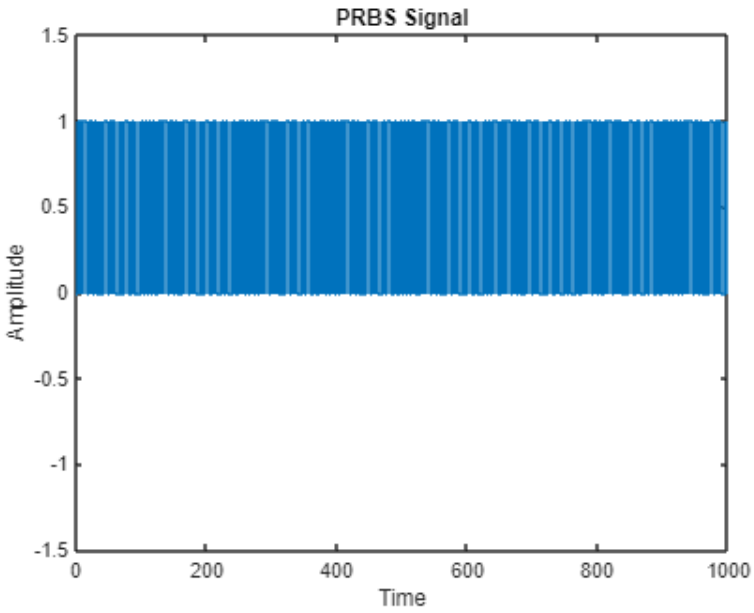
>>> 1.2 - Input/Output Generation

>>> Generate a signal with the length of 1000 in PRBS format. Acoordingly, time span vector must be generated and the `input_signal` matrix is formed to be loaded in Simulink model.

```
N = 1000;
amplitude = 1;

prbs_signal = transpose(amplitude * prbs(5, N));
tspan = transpose(0:proper_Ts:N*proper_Ts-1);
input_signal = [tspan prbs_signal];
```

The generated PRBS signal is shown in the following figure.

```
figure(1);
stairs(prbs_signal);
xlabel('Time');
ylabel('Amplitude');
title('PRBS Signal');
ylim([-1.5, 1.5]);
```



>>> Having run the Simulink model with the generated input signal, the output signal must be read in the script using:

```
y_out = y_out.Data(1:end-1);
```

```
y_out = y_out.Data(1:end-1);
```

>>> 1.3 - ARX model - degree of 1 to 10

>>> The sought information are being reported in the following except for the covariance matrix, since its size gets bigger and bigger by the degree increment. So in case you want to check it out, please run the code manually and fetch its value at each iteration.

```
fprintf("====================Degree Extraction======================\n")
```

```
======================Degree Extraction=======================
```

```
R2s  = [];
```

```
MSEs = [];
dets = [];
vars = [];
covs = [];
for degree=1:1:10
    na = degree;
    nb = degree;
    p = na + nb;

    U = arx_U_builder(na,nb,prbs_signal,y_out);

    theta_hat = inv(U'*U)*U'*y_out;
    y_hat = U*theta_hat;

    t = (step_resp.TimeInfo.Start:proper_Ts:step_resp.TimeInfo.End);

    [r2_arx, mse_arx] = rSQR(y_out, y_hat);

    error = y_out - y_hat;
    S_hat = 0;
    for i=1:length(error)
        S_hat = S_hat + error(i)^2;
    end
    variance = S_hat/(N-p);

    covariance = variance*inv(U'*U);

    detUTU = det(U'*U);
    R2s = [R2s; r2_arx];
    MSEs = [MSEs; mse_arx];
    dets = [dets; detUTU];
    vars = [vars; variance];
%     covs = [covs; covariance];
    fprintf(">>> Degree = %d : R2=%f | MSE=%f | det(UT.U)=%f | var=%f \n", degree, r2_arx, mse_arx, detUTU, variance)
    fprintf("-------------------------------------------------------------\n")
end
```

```
>>> Degree = 1 : R2=0.854735 | MSE=0.703615 | det(UT.U)=24888679.879529 | var=0.705025
-------------------------------------------------------------
>>> Degree = 2 : R2=0.864654 | MSE=0.655571 | det(UT.U)=4714768037060.724609 | var=0.658204
-------------------------------------------------------------
>>> Degree = 3 : R2=0.872951 | MSE=0.615380 | det(UT.U)=825227572548896384.000000 |
var=0.619094
-------------------------------------------------------------
>>> Degree = 4 : R2=0.930552 | MSE=0.336381 | det(UT.U)=1343992493919617742274 56.000000 |
var=0.339094
-------------------------------------------------------------
```

```
>>> Degree = 5 : R2=0.992569 | MSE=0.035991 | det(UT.U)=1176677260582184874525746790 4.000000 |
var=0.036355
------------------------------------------------------------
>>> Degree = 6 : R2=0.993220 | MSE=0.032842 |
det(UT.U)=109943778176410353379711685492736.000000 | var=0.033241
------------------------------------------------------------
>>> Degree = 7 : R2=0.994593 | MSE=0.026189 |
det(UT.U)=936125948527886606373888097756643328.000000 | var=0.026560
------------------------------------------------------------
>>> Degree = 8 : R2=0.996743 | MSE=0.015778 |
det(UT.U)=6332024679739567099688321680302946648064.000000 | var=0.016035
------------------------------------------------------------
>>> Degree = 9 : R2=0.997499 | MSE=0.012115 |
det(UT.U)=256812227590677513820863808030330038234644 48.000000 | var=0.012337
------------------------------------------------------------
>>> Degree = 10 : R2=0.998046 | MSE=0.009467 |
det(UT.U)=797808657161727729817691705397740812087223910 40.000000 | var=0.009660
------------------------------------------------------------
```

```
fprintf("================================================================\n")
```

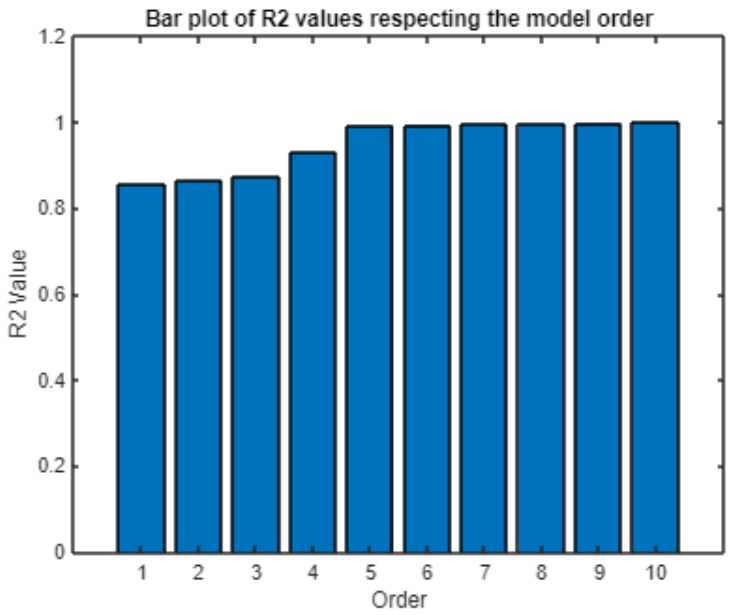
```
================================================================
```

>>> Also, note that the dterminant value of the `U'*U` matrix is increased wildly as the testing degree is being added up.

>>> 1.4 - Bar Plots

1.4.1 Based on R2

```
figure(2);
bar(1:1:10, R2s)
title("Bar plot of R2 values respecting the model order")
ylim([0,1.2])
xlabel("Order")
ylabel("R2 Value")
```



1.4.2 Based on variance

```
figure(3);
bar(1:1:10, vars)
% ylim([0,1])
title("Bar plot of variance values respecting the model order")
xlabel("Order")
ylabel("Variance Value")
```

1.4.3 Based on MSE

```
figure(4);
```

```
bar(1:1:10, MSEs)
% ylim([0,1])
title("Bar plot of MSE values respecting the model order")
xlabel("Order")
ylabel("MSE Value")
```

1.4.4 Based on determinant

>>> Since the determinant value are increasing rapidlly, plotting the values won't provide any suitable visualization of the story.

>>> 1.5 - System Order based on R2 values

>>> The minimum accepted value for thr R2 matric is reported as 0.9025.

So exploring the stored values from the previous section from order of 1 to 10 demonstrates that:

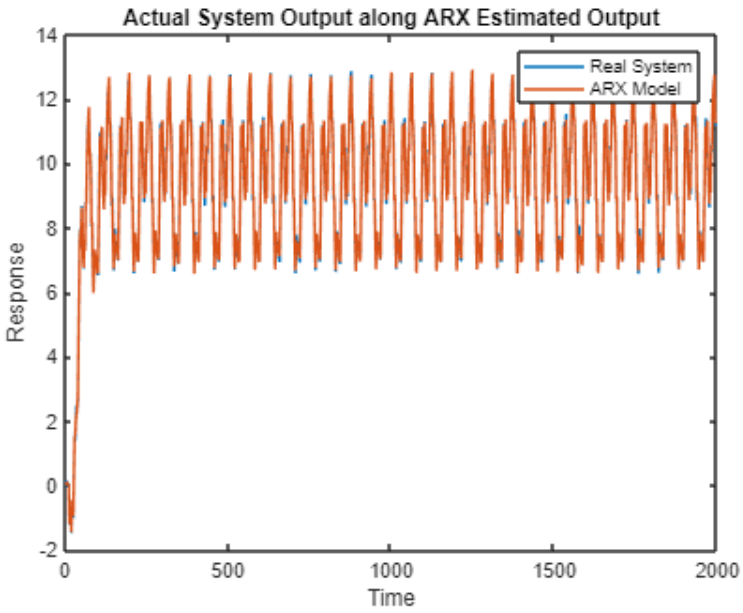
```
R2_accuracy_level = 0.9025;

[maxR2, maxR2Index] = max(R2s);

degree_based_on_R2 = maxR2Index;

first_acceptable_degree_based_on_R2 = min(find(R2s>R2_accuracy_level));

fprintf("=============================================================================================\n")
```

```
=============================================================================================
```

```
fprintf(">>> with respect to R2: Although the estimation starts to be accurate enough from the degree of %d,\n    the best results are obtained in degree of %d.\n", first_acceptable_degree_based_on_R2, degree_based_on_R2)
```

>>> with respect to R2: Although the estimation starts to be accurate enough from the degree of 4,
    the best results are obtained in degree of 10.

```
na = degree_based_on_R2;
nb = degree_based_on_R2;
p = na + nb;

U = arx_U_builder(na,nb,prbs_signal,y_out);

theta_hat = inv(U'*U)*U'*y_out;
y_hat = U*theta_hat;

t = (step_resp.TimeInfo.Start:proper_Ts:step_resp.TimeInfo.End);

figure(5)
plot(tspan,y_out,tspan,y_hat)
title("Actual System Output along ARX Estimated Output")
xlabel("Time")
ylabel("Response")
legend('Real System','ARX Model')
```



>>> 1.6 - System Order based on MSE values

>>> The maximum accepted value for the MSE matric is taken as 0.1.

So exploring the stored values from the previous section from order of 1 to 10 demonstrates that:

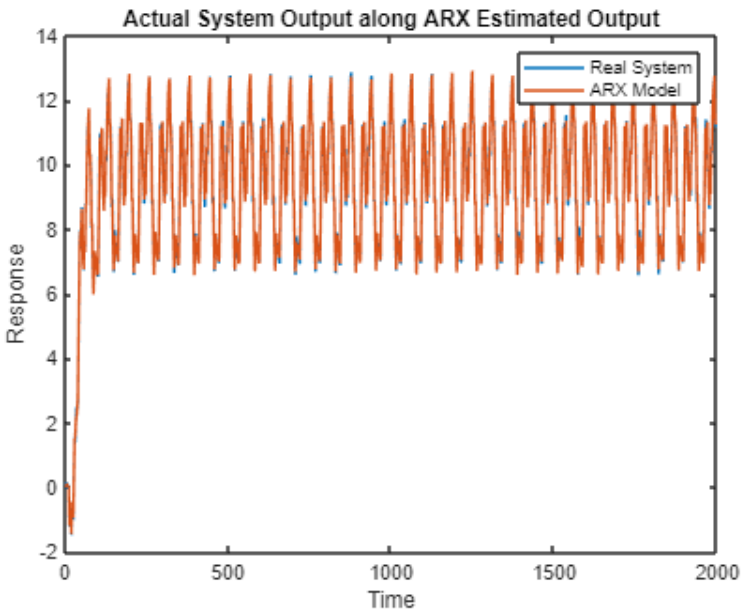
```
mse_accuracy_level = 0.1;

[minMSE, minMSEIndex] = min(MSEs);
```

```
degree_based_on_MSE = minMSEIndex;

first_acceptable_degree_based_on_MSE = min(find(MSEs<mse_accuracy_level));

fprintf("========================================================================================\n")
```

```
fprintf(">>> with respect to MSE: Although the estimation starts to be accurate enough from the degree of %d,\n    the best results are obtained in degree of %d.\n", first_acceptable_degree_based_on_MSE, degree_based_on_MSE)
```

```
>>> with respect to MSE: Although the estimation starts to be accurate enough from the degree of 5,
    the best results are obtained in degree of 10.
```

```
na = degree_based_on_MSE;
nb = degree_based_on_MSE;
p = na + nb;

U = arx_U_builder(na,nb,prbs_signal,y_out);

theta_hat = inv(U'*U)*U'*y_out;
y_hat = U*theta_hat;

t = (step_resp.TimeInfo.Start:proper_Ts:step_resp.TimeInfo.End);

figure(6)
plot(tspan,y_out,tspan,y_hat)
title("Actual System Output along ARX Estimated Output")
xlabel("Time")
ylabel("Response")
legend('Real System','ARX Model')
```



>>> 1.7 - System Order based on Variance values

>>> The maximum accepted value for the Variance matric is taken as 0.1.

So exploring the stored values from the previous section from order of 1 to 10 demonstrates that:

```
var_accuracy_level = 0.1;

[minVar, minVarIndex] = min(vars);

degree_based_on_Var = minVarIndex;

first_acceptable_degree_based_on_Var = min(find(vars<var_accuracy_level));

fprintf("==========================================================================================\n")
```

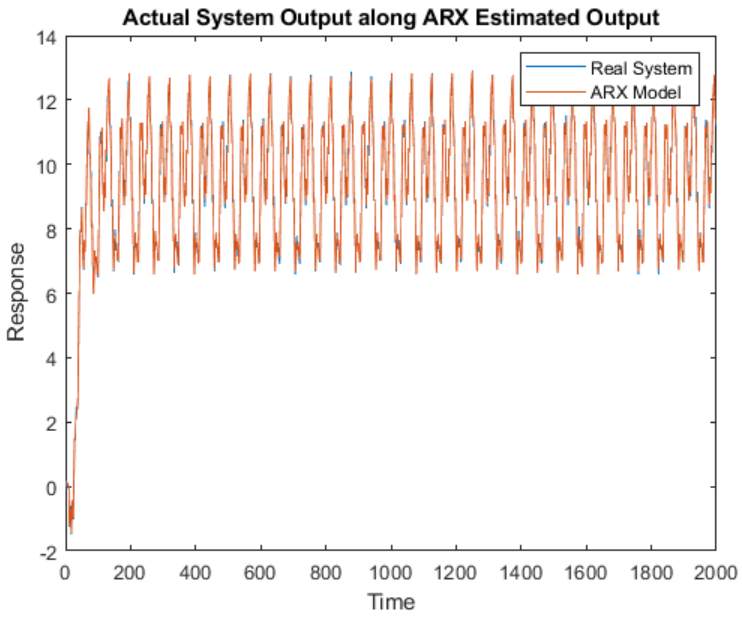
```
==========================================================================================
```

```
fprintf(">>> with respect to Variance: Although the estimation starts to be accurate enough from the degree of %d,\n
the best results are obtained in degree of %d.\n", first_acceptable_degree_based_on_Var, degree_based_on_Var)
```

```
>>> with respect to Variance: Although the estimation starts to be accurate enough from the degree of 5,
    the best results are obtained in degree of 10.
```

```
na = degree_based_on_Var;
nb = degree_based_on_Var;
p = na + nb;

U = arx_U_builder(na,nb,prbs_signal,y_out);

theta_hat = inv(U'*U)*U'*y_out;
y_hat = U*theta_hat;
```

```
t = (step_resp.TimeInfo.Start:proper_Ts:step_resp.TimeInfo.End);

figure(7)
plot(tspan,y_out,tspan,y_hat)
title("Actual System Output along ARX Estimated Output")
xlabel("Time")
ylabel("Response")
legend('Real System','ARX Model')
```



>>> 1.8 - Real Order of the System using the Determinants

```
fprintf("=============================================================================================\n")
```

```
=============================================================================================
```

```
fprintf(">>> Let's try higher orders to see wether the determinant of the system ever zero or not.\n")
```

```
>>> Let's try higher orders to see wether the determinant of the system ever zero or not.
```

```
fprintf("    For so, run the algorithm from degree of 11 to 100.\n")
```

```
    For so, run the algorithm from degree of 11 to 100.
```

```
fprintf("=============================================================================================\n")
```

```
=============================================================================================
```

```
realDegree = false;
for degree=11:1:100
    na = degree;
    nb = degree;
    p = na + nb;

    U = arx_U_builder(na,nb,prbs_signal,y_out);
```

```
    theta_hat = inv(U'*U)*U'*y_out;
    y_hat = U*theta_hat;

    t = (step_resp.TimeInfo.Start:proper_Ts:step_resp.TimeInfo.End);

    [r2_arx, mse_arx] = rSQR(y_out, y_hat);

    detUTU = det(U'*U);

    fprintf(">>> Degree = %d : R2=%f | MSE=%f | det(UT.U)=%f | var=%f \n", degree, r2_arx, mse_arx, detUTU, variance)
    fprintf("-----------------------------------------------------------\n")

    if (detUTU < 1)
        fprintf("YEP ! The system real degree is %d.\n", degree)
        realDegree = true;
        break;
    end
end
```

```
>>> Degree = 11 : R2=0.998367 | MSE=0.007911 | det(UT.U)=19241614751643890997871461887084824834024339511705 6.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 12 : R2=0.998580 | MSE=0.006879 | det(UT.U)=385318705693384610246444616061248948566320883630604288.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 13 : R2=0.998689 | MSE=0.006350 | det(UT.U)=6645616607109557950001402812138870356670210036927876300 80.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 14 : R2=0.998777 | MSE=0.005923 | det(UT.U)=1050003253957598727938880983975628539507863896002144216547328.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 15 : R2=0.998879 | MSE=0.005428 | det(UT.U)=1527729495973817806135837434844659143451495416734479486289969152.000000 |
var=0.009660
-----------------------------------------------------------
>>> Degree = 16 : R2=0.998919 | MSE=0.005238 | det(UT.U)=19731585998334374027074758883308760307975683378679297524571203174 40.000000 |
var=0.009660
-----------------------------------------------------------
>>> Degree = 17 : R2=0.998941 | MSE=0.005130 | det(UT.U)=2349139144705444967761436174168742378884641021157754276455136347815936.000000 |
var=0.009660
-----------------------------------------------------------
>>> Degree = 18 : R2=0.998949 | MSE=0.005090 | det(UT.U)=2665946747916811205753890584352735733929949098387533065560220851903660032.000000 |
var=0.009660
-----------------------------------------------------------
>>> Degree = 19 : R2=0.998964 | MSE=0.005017 | det(UT.U)=2965432929331652036825639068227110801899776141427223104602550363339005886464.000000
| var=0.009660
-----------------------------------------------------------
>>> Degree = 20 : R2=0.998972 | MSE=0.004980 |
det(UT.U)=3162302711932641209895590287095975151185897169533521953443344032867413624094720.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 21 : R2=0.998980 | MSE=0.004940 |
det(UT.U)=3316319292048510234253527764435569632290229438256749110941384344164249339317714944.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 22 : R2=0.998982 | MSE=0.004931 |
det(UT.U)=3389130870729679707992719965271460430743675746150046994629391715233494083728090071040.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 23 : R2=0.998986 | MSE=0.004911 |
det(UT.U)=3278211194305734236661142528278934695744133958639584325048318271684917490814255206760448.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 24 : R2=0.998991 | MSE=0.004888 |
det(UT.U)=2811766175977739838339198238676843182702623555355610064359967480671060716722082537191505920.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 25 : R2=0.998991 | MSE=0.004887 |
det(UT.U)=1688116413181497066988234753335369089542565003973833079637361853243590510616819526597477400576.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 26 : R2=0.998992 | MSE=0.004883 |
det(UT.U)=8132387176501214684735686128858831513682981523878724392720696203255382793480975228178377044131 84.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 27 : R2=0.998993 | MSE=0.004879 |
det(UT.U)=3361153024024841815415420651440675542585156335953676929183373427802132357178039853409996649749544 96.000000 | var=0.009660
-----------------------------------------------------------
>>> Degree = 28 : R2=0.998997 | MSE=0.004857 |
det(UT.U)=6204353531540727582306103654652320026327840718266020661677219098170873485080527062844223075567796224 0.000000 | var=0.009660
```

>>> Degree = 29 : R2=0.998998 | MSE=0.004853 |
det(UT.U)=4383478802905857300199102725355554363110409844558222210906810702897181958052113396828322817548205686784.000000 | var=0.009660

>>> Degree = 30 : R2=0.998998 | MSE=0.004851 |
det(UT.U)=3006937359139690830817872100556718926029120238878467019326423747204033265841685373148401479547630534000064.000000 | var=0.009660

>>> Degree = 31 : R2=0.999005 | MSE=0.004820 |
det(UT.U)=160310300826690609099886657717283896263482811332040192347895497732008153502886566767717630964589652827176960.000000 | var=0.009660

>>> Degree = 32 : R2=0.999005 | MSE=0.004819 |
det(UT.U)=6566936940346016023623108107385646630406243637160663327404516940205032485104578926771615776455506584257167360.000000 | var=0.009660

>>> Degree = 33 : R2=0.999005 | MSE=0.004818 |
det(UT.U)=267817788815314551864676100976954220686020748317614987949605059513621441300667378502761855806882464190232002560.000000 |
var=0.009660

>>> Degree = 34 : R2=0.999011 | MSE=0.004789 |
det(UT.U)=10803242219609890895018196715380287913208118904816584944640510894530530433342046144520845364831703595045921751040.000000 |
var=0.009660

>>> Degree = 35 : R2=0.999017 | MSE=0.004764 |
det(UT.U)=432281275574804422754918959909895141612039875923107198861198542339315456338336208853845379038052633525013079654400.000000 |
var=0.009660

>>> Degree = 36 : R2=0.999017 | MSE=0.004761 |
det(UT.U)=17056854114206687196302669382372881455987341719806597521277228431982084239021298266306967681093904921147755861114880.000000 |
var=0.009660

>>> Degree = 37 : R2=0.999018 | MSE=0.004756 |
det(UT.U)=665711218259464926770099371676830906650355286825092630765518626969921010970116823181324073828430111093540748106137600.000000 |
var=0.009660

>>> Degree = 38 : R2=0.999019 | MSE=0.004752 |
det(UT.U)=25811949759857122641330723727702724957597939920874468651325604434142779143763371177379098865043997744168440255270092800.000000 |
var=0.009660

>>> Degree = 39 : R2=0.999020 | MSE=0.004745 |
det(UT.U)=9913577814299901846225798711881071470908547640449790941655728416553380822373651055073798723082252480329687080284612198400.000000 |
var=0.009660

>>> Degree = 40 : R2=0.999021 | MSE=0.004744 |
det(UT.U)=37882345386391055911275835537261578837586471185335125778702720781549078545766827346793684689768541249263007036144300851200.000000 |
var=0.009660

>>> Degree = 41 : R2=0.999021 | MSE=0.004741 |
det(UT.U)=1421180225283723763184202631384147251478887372048407297655130177810435081955372733424376406935679878656932242361443313254400.000000
| var=0.009660

>>> Degree = 42 : R2=0.999021 | MSE=0.004741 |
det(UT.U)=52705519933560960281228906954907723074547744662775021699349988513214963840476660128689754809949569309433049345165438327193600.000000
| var=0.009660

>>> Degree = 43 : R2=0.999022 | MSE=0.004739 |
det(UT.U)=19520107570200891577826899683344857179838816286852241423660925047645243266503696228432123407259854929900747323585223876149248000.0000
00 | var=0.009660

>>> Degree = 44 : R2=0.999023 | MSE=0.004734 |
det(UT.U)=7135815459909006837146789656418212427070056721097387042048373274615319506300107025795669559394992527808541237836754276654776320.000
000 | var=0.009660

>>> Degree = 45 : R2=0.999029 | MSE=0.004702 |
det(UT.U)=259998177787989233565839773225454225970311095163714522441883129890148243362661458198366495083313078413385249520100330107641528320.0
00000 | var=0.009660

>>> Degree = 46 : R2=0.999029 | MSE=0.004701 |
det(UT.U)=934148154657430166733769322206023851079182351246862438405172882901037791217652141778967888905733667458625348814942578484499709952 0.
000000 | var=0.009660

>>> Degree = 47 : R2=0.999031 | MSE=0.004694 |
det(UT.U)=3325717977417292445067082008692769803934340071739796833171821495441995815591020632481806403163841904027878898760335412500426632724 8.000000 | var=0.009660

>>> Degree = 48 : R2=0.999039 | MSE=0.004657 |
det(UT.U)=11770755996634137999875041084835267872616114989006941921056968167395568315998568295169341058105344695404141005462611824745994675814
400.000000 | var=0.009660

>>> Degree = 49 : R2=0.999039 | MSE=0.004654 |
det(UT.U)=4102608353689924778114704536860939484744021759940701863737947398385354832905836917075857679647717385439712951186783541056635854756 24960.000000 | var=0.009660

>>> Degree = 50 : R2=0.999040 | MSE=0.004652 |
det(UT.U)=1412817276912411192563698984893294666237355152387765300348088597086804874797589762588425283774641495889631491466147505432142655246 4605184.000000 | var=0.009660

```
>>> Degree = 51 : R2=0.999042 | MSE=0.004641 |
det(UT.U)=4839722642806197194392408161585171843267232728528509215219240943632991925946392604079842043000685433773408139530585373068005218197
1340544.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 52 : R2=0.999043 | MSE=0.004635 |
det(UT.U)=1645457990295678147121041276122267413547339555168105256110243093498082607623461435192353635507153966966248846934694787521283545891
904051200.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 53 : R2=0.999044 | MSE=0.004631 |
det(UT.U)=5559675003069307990545888481036240549043434230909748641122739921054998175850793818129520671433034702471942321910453147706342862526
1807520768.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 54 : R2=0.999045 | MSE=0.004626 |
det(UT.U)=1866576231139319529919765258580391502947795508362860418575443317704511120980546821131822710007814445347990302977856606133267275518
064262815744.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 55 : R2=0.999048 | MSE=0.004611 |
det(UT.U)=6242209203281222857184837040183635231149030232352770119783452067313844117672571255920869400899140967971056080740104265045767298306
9254886342656.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 56 : R2=0.999052 | MSE=0.004593 |
det(UT.U)=2071142045161640290202165636503044045198787553150142184107430528212825564794718050012013871350001117475250217977063776714095739821
9520159942443008.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 57 : R2=0.999054 | MSE=0.004581 |
det(UT.U)=6813504651679207955302928537788817181239923640760089515899544145668572156527288101309927313048022031139645043503079425420487526591
9390314206035968.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 58 : R2=0.999055 | MSE=0.004578 |
det(UT.U)=2223486532967685296521863237937601358146393453805767414762445649320991994876069894575566355220716680895231069883363230476099332453
87144515405506150 4.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 59 : R2=0.999055 | MSE=0.004578 |
det(UT.U)=7207707790622037200710801315345376217577272510296741523233995373759147177802902766683030498053375781623897385657423292464429851954 8
329387529491133 2352.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 60 : R2=0.999061 | MSE=0.004546 |
det(UT.U)=2332720472262036382789708691001352613472338405836889168691418541898971842512792868574390155438666917397849117298301353769632739786 1
57233572405758132224.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 61 : R2=0.999061 | MSE=0.004546 |
det(UT.U)=7463232868141329967590666839336821129663090770815228175147702147085514579743851817533897714854136650524898310482355811937962448233 4
119895201958297652 4288.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 62 : R2=0.999067 | MSE=0.004521 |
det(UT.U)=2370429897244537241115811568303963780671357482220969713463784211753565187177445783486221205339795759967241690339641785259908452948 8
020283120581341052141568.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 63 : R2=0.999071 | MSE=0.004501 |
det(UT.U)=7422846517658163658296432521353203537578910735321069029839962372848218548764793943547653493398071980066810488062363845897531607198 0
0075093280634379666718720.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 64 : R2=0.999072 | MSE=0.004495 |
det(UT.U)=2311831910692868420208669712250382347750020920522666889336837602735107512761056892794923150365826124114483683139821905625123084558 4
5167120778774606866798346 24.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 65 : R2=0.999073 | MSE=0.004491 |
det(UT.U)=7181974947123639216920984123475773324076806723981373524265005125737402149119899870820421059251064865782914468457127729634670774724 2
196222973202027013544817 4592.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 66 : R2=0.999073 | MSE=0.004489 |
det(UT.U)=2218873907211576181555478701482468165349572432826347633833614201749939522172473156915906717565966895218722191088009493776413352231 8
627427799835659664353082212352.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 67 : R2=0.999088 | MSE=0.004415 |
det(UT.U)=6843837554225099242034943008547213127429395272036074916530403832510595075256202730355478274133374874678712549583629121633904725706 7
5040968325185555124182088417280.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 68 : R2=0.999089 | MSE=0.004412 |
det(UT.U)=2073031595677131751846583094271358465904209690701100599493455936680737693217411974073355373165766679000356550189367303614771907520 5
6136320073570506108655725665320 96.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 69 : R2=0.999090 | MSE=0.004409 |
det(UT.U)=6243786698502151600037913598137974629550459530984937775542643007092124900621800571908806058561698890911844700809535520341450956481 5
836270049055233410592052723947929 6.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 70 : R2=0.999095 | MSE=0.004384 |
det(UT.U)=1873745801242164475399311956957873398175411262312178003302093193636129451572272230389129002351746909503056178107628776769121977395 5
5844986421201655125342244860759900 16.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 71 : R2=0.999099 | MSE=0.004364 |
det(UT.U)=5576460212018903638549796545000260431563867917682953291624051881955114576753613855554044121207603159286624349812941274312980495233 6
8481897047134465833300082414992752640.000000 | var=0.009660
------------------------------------------------------------
```

```
>>> Degree = 72 : R2=0.999100 | MSE=0.004362 |
det(UT.U)=164898783640234229137251143725734460910305279484822995185674288514396736932730825803173276663556983167305449008850696895398658521374852230074877618256743777252975658926 08.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 73 : R2=0.999101 | MSE=0.004357 |
det(UT.U)=485032020638191103146074740914505936235342135616591834373643099764683973547308314088328052682842374125438048729664061170551039059983444283409638767725428748202230258073600.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 74 : R2=0.999102 | MSE=0.004347 |
det(UT.U)=142288558367955541869381669862627881859214830655515875744963548423160162973537445529700174998123258194660887403905833895371949469471316210719506399736611435270202934920478 72.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 75 : R2=0.999107 | MSE=0.004324 |
det(UT.U)=41601736147581789578961865251406389573190604143191506333940503387090006115009987594728897293775167086039995982420352744334374333948870042396474027819373276225365068334550220 8.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 76 : R2=0.999109 | MSE=0.004316 |
det(UT.U)=120928186383180841344179198491909107053715693653071608738033246196248876241702943266098000681832677414137562508707798797657620654169243726553881047331141553804981843795996835 84.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 77 : R2=0.999110 | MSE=0.004311 |
det(UT.U)=350578287955420726286354879198413247522629528613910499918837447536831476789732725292151123662148096152412146800979091067487848450301257264111359893022050485919405632339492995072.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 78 : R2=0.999112 | MSE=0.004302 |
det(UT.U)=101479216869959627300507651763807866251189300412548817456799529472193540638506029667121924678124521790448918765763649128028772726327926841623374099133850481429174851552132391239 68.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 79 : R2=0.999114 | MSE=0.004293 |
det(UT.U)=292974599735831082065625831197589526805128082096856980330069692519793115493435081928373655422494453026720360669793104302898531822783554084278736891453187096524945040870927316811776.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 80 : R2=0.999114 | MSE=0.004290 |
det(UT.U)=840324897393815408393631142336817869671655938890200914272607461688921320215490323294736507540899139566214361381640611274398363798835626882160368473348411580671300447018285266894848 0.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 81 : R2=0.999119 | MSE=0.004266 |
det(UT.U)=240034612698685988128788609501126082969046325998702502888962240604438013122650231890607954830361324657013650482327068029261451682790401839281194690705993151039993030239592385030914048.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 82 : R2=0.999120 | MSE=0.004260 |
det(UT.U)=681379652653147939167347222171029532107956428133221214337406692345644261066802314134634128187787645284548330593803122952363697072196762791408770328462338741106682946433170931774075699 2.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 83 : R2=0.999121 | MSE=0.004257 |
det(UT.U)=191330542942140371893278525050209814606307215472340250927255015694743853550765188867241836160674562348748803962141153682903520739017012062659241231344046173893151275173955667268468736000.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 84 : R2=0.999122 | MSE=0.004254 |
det(UT.U)=534611529425972111053255342755828828193933011095422355427158653728921595683041724590946718956227660969971392407148652972321819220835358795648433998586474054783470862101720243827134116659 2.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 85 : R2=0.999126 | MSE=0.004233 |
det(UT.U)=149096275544730619557384010076678674799491091886976013578553387331395973519516710053766590047978873388552161573151449291357161732804960459031841522756790410329764826326920768090746273136640.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 86 : R2=0.999127 | MSE=0.004230 |
det(UT.U)=410463068478146010210772560986795339557433882544199606591161576377227740080132010671073207477600931897580712261480576738405694982137356817806255658978607306462615261481777310739487415022387 2.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 87 : R2=0.999131 | MSE=0.004210 |
det(UT.U)=112541422455042001833876875654220851619001836452280896901254425989166269390959716574229573536158797492276479129247237781851301287585407876567340146217841622194091843858667099402440793160417280.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 88 : R2=0.999131 | MSE=0.004207 |
det(UT.U)=306593360129687518182169950305470102041446131890704726011960770909092572546517791728291983096229405478741487728582548867019189523052619040314696004643450954635465674306006402938240464359050444 8.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 89 : R2=0.999133 | MSE=0.004200 |
det(UT.U)=830236361963269959967369690855779747555189372762300362413541164613736864574366338295196920543975299981970221244355938182945204237789274658695536715533117784988644902409205774634770041827219210 24.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 90 : R2=0.999134 | MSE=0.004194 |
det(UT.U)=224160244277261516342985511256169887471570121818157212397984078231096673811706532272561623666784091574897397503234519390248071401126399185730303406187169600068319212107567533037717667508510733107 2.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 91 : R2=0.999143 | MSE=0.004152 |
det(UT.U)=603458213540513611271665716050748989468034106060355181136292553429775289625214279917280553342215424477768911519514360248942969405467248936017763588403055660287301828945408773357482161462012228403 20.000000 | var=0.009660
------------------------------------------------------------
>>> Degree = 92 : R2=0.999145 | MSE=0.004140 |
det(UT.U)=160107162336431469624423053277850796094743167139110257306273758323544165868444625914789786487851165021747918430303987060018794449150631756117068389921948561266941761666142726376903459495359156951449 6.000000 | var=0.009660
------------------------------------------------------------
```

```
>>> Degree = 93 : R2=0.999147 | MSE=0.004133 |
det(UT.U)=42297315933242693398606661262235432119219611895584655217125745919419578694810126727616594586541269534874511589768758618934272144549
17979348112489885956197147133582852625020783575741261715794283410227 2.000000 | var=0.009660
------------------------------------------------------------------
>>> Degree = 94 : R2=0.999147 | MSE=0.004131 |
det(UT.U)=11142391686712127532170350858097468270265491519505259745075959318736422721004432327360329001487043494862731592627375100452449386359
38791227703506980969547513171090935708250322070104907964153685442822144.000000 | var=0.009660
------------------------------------------------------------------
>>> Degree = 95 : R2=0.999149 | MSE=0.004123 |
det(UT.U)=29299222928355992839739730936570297886521042772608958831981503719121040590300664879624680935996862407856272753228676776214065353908
93762811204232646099074165367066326665734236781530775217255781297133977 6.000000 | var=0.009660
------------------------------------------------------------------
>>> Degree = 96 : R2=0.999150 | MSE=0.004118 |
det(UT.U)=76582528470107130507493931280461545214429476124486472677705170140972241728580939759774570672009669796697317884950272146434788300112
85530418167373780622858694318708629130213582924349801684726666778745241 60.000000 | var=0.009660
------------------------------------------------------------------
>>> Degree = 97 : R2=0.999151 | MSE=0.004113 |
det(UT.U)=19981623596207235360162577716440251416582039332531075964717542327142162978625026181895442182315974704428783841008696482457057235573
88333324289544857647324820044365275005189155883653141231576774402142476697 6.000000 | var=0.009660
------------------------------------------------------------------
>>> Degree = 98 : R2=0.999153 | MSE=0.004101 |
det(UT.U)=51966181264758586269282574280854805019639268019617490755427268713749713056645873135404861167424766616898375451321949856060942815402
94469897412971967290288200108202299451887068862264096682601467443976035368 96.000000 | var=0.009660
------------------------------------------------------------------
>>> Degree = 99 : R2=0.999157 | MSE=0.004085 |
det(UT.U)=13458030272772235828539185431772418169845327195840357663648638890736909152519608813215540876677006434083106073168484601656606240892
61457338708314356644859150247010057431824491736890254523356167398404652217139 2.000000 | var=0.009660
------------------------------------------------------------------
>>> Degree = 100 : R2=0.999163 | MSE=0.004056 |
det(UT.U)=34702800150504918295612919584316112197248998367413819959424439401192688047536296416565980437769666423341381934866118760671843893423
63368621215339752714294748250900042142423680535110279855453988626438984522792 96.000000 | var=0.009660
------------------------------------------------------------------
```

```
if ~realDegree
    fprintf(">>> No! Not until degree of 100. \n")
end
```

```
>>> No! Not until degree of 100.
```

```
fprintf("====================================================================\n")
```

```
====================================================================
```

>>> 1.9 - Comparing Approaches

>>> Both Variance-based and MSE-based degree extraction approaches suggested a proper order of 5 for the given system while the R2-based stated that the model would be good enough with the order of 4.

Question 2

```
clear;
clc;

load('402123100-q2.mat')
```

>>> 2.0 - Initial Assignments

```
na = 3*n;
nb = 3*n;
p = na+nb;
N = length(id.u);
```

>>> 2.1 - RLS Function

>>> Write the following piece of code in a separate `.m` file and save it in the same directory as the current file.

```
% function [theta, U] = RLS(na, nb, th0, p0, u, y)
%
%     p = na + nb;
%     N = length(y);
%     theta = th0;
%     P = p0 * eye(na + nb);
%
%     theta_estimate=th0';
%
%     U=[];
%     Y=[];
%
%     for i=1:N
%         Y = [Y ; y(i)];
%
%         Ut=zeros(1,p);
%         Y1=y(i);
%         for j=1:na
%             if (i-j>0)
%                 Ut(1,j) = -y(i-j);
%             end
%         end
%         for j=na+1:na+nb
%             k=j-na;
%             if (i-k>0)
%                 Ut(1,j) = u(i-k);
%             end
%         end
%
%         U=[U;Ut];
```

```
%         Kt=(P*Ut')/(1+Ut*P*Ut');
%         theta = theta + Kt*(Y1 - Ut*theta);
%         theta_estimate = [theta_estimate ; theta'];
%         P = (eye(na+nb) - Kt*Ut) * P;
%     end
%
% end
```

>>> 2.2 - ARX Modeling

2.2.0 Data Split

```
data10 = id(1:floor(length(id.u)*0.1));
data100 = id;
```

2.2.1 ARX Modeling - 100% of Data

```
U100 = arx_U_builder(na, nb, data100.u, data100.y);
theta_hat100 = inv(U100'*U100)*U100'*data100.y;

[theta100, U100] = RLS(na, nb, theta_hat100, 0.01*eye(p), data100.u, data100.y);
y100 = U100*theta100;
```

>>> theta with 100% of data:

```
theta100'
```

```
ans = 1×18
    0.0265   -0.3508   -0.3159   -0.2247   -0.2263   -0.0957   -0.0737 ⋯
```

2.2.2 ARX Modeling - 10% of Data

```
U10 = arx_U_builder(na, nb, data10.u, data10.y);
theta_hat10 = inv(U10'*U10)*U10'*data10.y;

[theta10, U10] = RLS(na, nb, theta_hat10, 0.01*eye(p), data10.u, data10.y);
y10 = U10*theta10;
```

>>> theta with 10% of data:

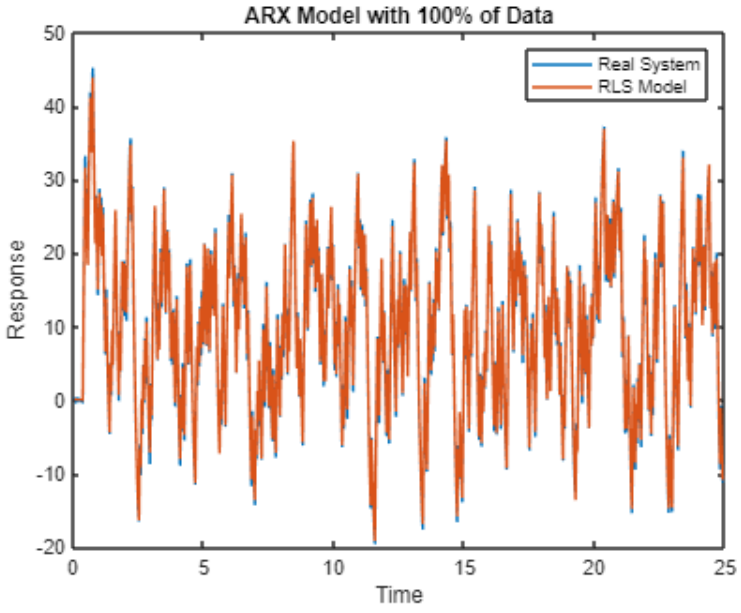
```
theta10'
```

```
ans = 1×18
    0.0337   -0.3867   -0.1876   -0.4060   -0.2768    0.0985   -0.0081 ⋯
```

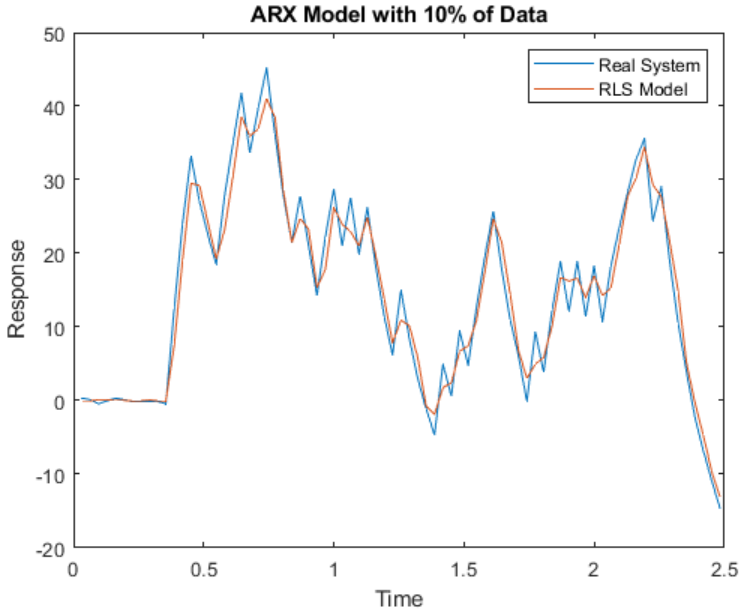
>>> 2.2.3 - Plot both 100% and 10%

```
t100 = (data100.Tstart:data100.Ts:N*data100.Tstart);
t10 = (data10.Tstart:data10.Ts:floor(N*0.1)*data10.Tstart);

figure(1)
plot(t100,data100.y,t100,y100)
```

```
title("ARX Model with 100% of Data")
xlabel("Time")
ylabel("Response")
legend('Real System','RLS Model')
```




```
figure(2)
plot(t10,data10.y,t10,y10)
title("ARX Model with 10% of Data")
xlabel("Time")
ylabel("Response")
legend('Real System','RLS Model')
```



>>> R2 values for both approaches:

100% of data:

```
r2_rls_100 = rSQR(data100.y, y100)
```

```
r2_rls_100 = 0.9979
```

10% of data:

```
r2_rls_10 = rSQR(data10.y, y10)
```

```
r2_rls_10 = 0.9619
```

Clearly, the model with 100% of data works better than the one working with 10% of data.

Question 3

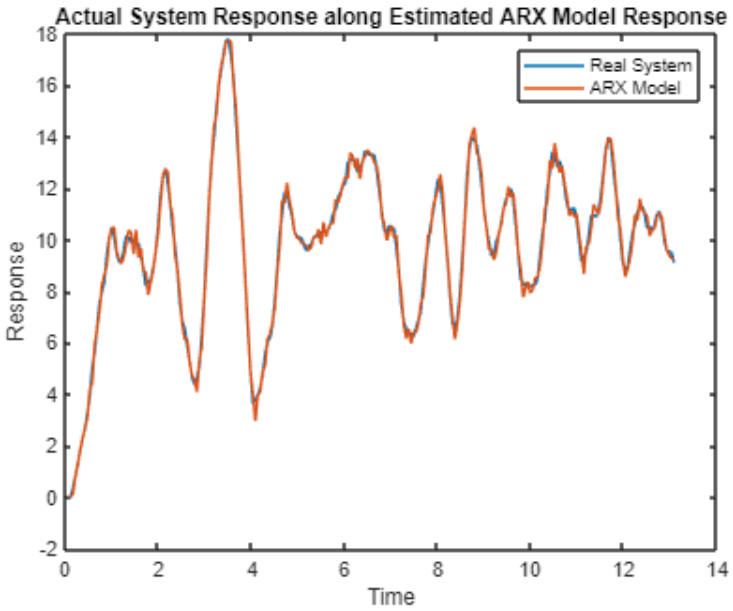
```
warning off; close all; clear; clc
load 402123100-q3

u = id.u;
y = id.y;
```

>>> 3.0 - Initial Assignments

```
N = length(y);
na = n;
nb = n;
p = na + nb;
```

>>> 3.1 - ARX Modeling

```
U = arx_U_builder(na,nb,u,y);

theta_hat = inv(U'*U)*U'*y;
y_hat = U*theta_hat;
t = (id.Tstart:id.Ts:N*id.Tstart);

figure(1)
plot(t,y,t,y_hat)
title("Actual System Response along Estimated ARX Model Response")
xlabel("Time")
ylabel("Response")
legend('Real System','ARX Model')
```



>>> Evaluate the Estimated Model

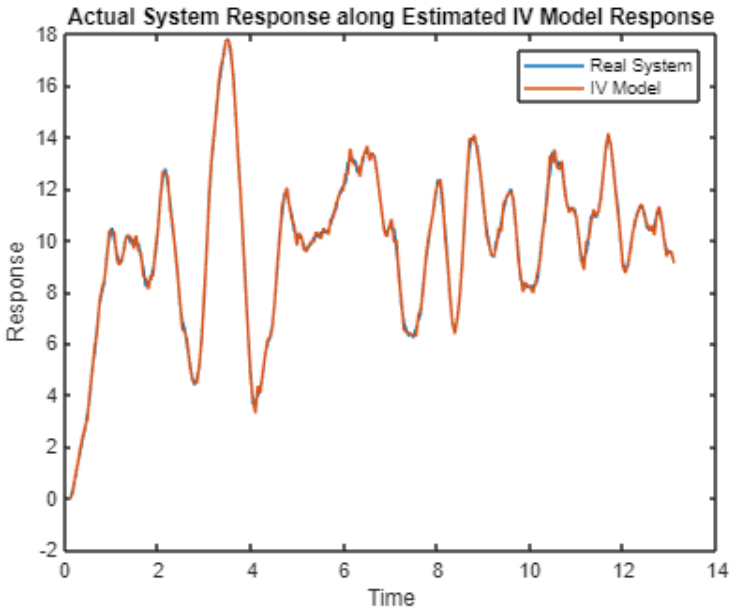
R2 parameter:

```
r2_arx = rSQR(y, y_hat)
```

```
r2_arx = 0.9910
```

>>> 3.2 - IIV Modeling

```
first_estimation = y_hat;
% first_estimation = randn(250,1);

Z = zeros(N,p);
U = zeros(N,p);

for index_N=1:N
    for index_y=1:na
        if index_N-index_y>0
            Z(index_N,index_y) = -first_estimation(index_N-index_y);
            U(index_N,index_y) = -y(index_N-index_y);
        end
    end
    for index_x=na+1:p
        if index_N-index_x+na>0
            Z(index_N,index_x) = u(index_N-index_x+na);
            U(index_N,index_x) = u(index_N-index_x+na);
        end
    end
end

theta_hat_IV = inv(transpose(Z)*U)*transpose(Z)*y;

y_IV = U*theta_hat_IV;
figure(2)
plot(t,y,t,y_IV)
title("Actual System Response along Estimated IV Model Response")
xlabel("Time")
ylabel("Response")
legend('Real System','IV Model')
```



>>> Evaluate the Estimated Model

R2 parameter:

```
r2_iv = rSQR(y, y_IV)
```

```
r2_iv = 0.9959
```

>>> 3.3 - ARX vs. IIV

>>> It is expected for the IIV method to leave a better performance than the ARX model and it is doing so. See the following reported R2 values for a better grasp of the results.

```
disp("================Model Evaaluation Report==================")
```

```
================Model Evaaluation Report==================
```

```
fprintf('------> R2 ARX  : %.7f \n', r2_arx);
```

```
------> R2 ARX  : 0.9909943
```

```
fprintf('------> R2 IIV  : %.7f \n', r2_iv);
```

```
------> R2 IIV  : 0.9958984
```

```
disp("============================================================")
```

```
============================================================
```

Question 4

```
warning off; close all; clc
```

>>> 4.0 - Data Read and Split

```
load HW4_Q4.mat

sys_1_data = Z;
sys_2_data = Z2;

data_length = length(sys_1_data.u);

sys_1_iden_data = sys_1_data(1:data_length/2);
sys_1_val_data = sys_1_data(data_length/2+1:end);

sys_2_iden_data = sys_2_data(1:data_length/2);
sys_2_val_data = sys_2_data(data_length/2+1:end);



sys_1_iden_u = sys_1_iden_data.u;
sys_1_iden_y = sys_1_iden_data.y;

sys_1_val_u = sys_1_val_data.u;
sys_1_val_y = sys_1_val_data.y;


sys_2_iden_u = sys_2_iden_data.u;
sys_2_iden_y = sys_2_iden_data.y;

sys_2_val_u = sys_2_val_data.u;
sys_2_val_y = sys_2_val_data.y;
```

>>> 4.1 - Reducing the Average to Zero

```
threshold = 10e-4; % = 0.001

if mean(sys_1_iden_u) > threshold
    sys_1_iden_u = detrend(sys_1_iden_u);
    sys_1_iden_y = detrend(sys_1_iden_y);
end

if mean(sys_2_iden_u) > threshold
    sys_2_iden_u = detrend(sys_2_iden_u);
    sys_2_iden_y = detrend(sys_2_iden_y);
end
```

>>> 4.2 - ARX for system 1 & 2

4.2.1 ARX - system 1

```
N = length(sys_1_iden_u);
fprintf("====================Degree Extraction System 1=======================\n")
```

```
====================Degree Extraction System 1=======================
```

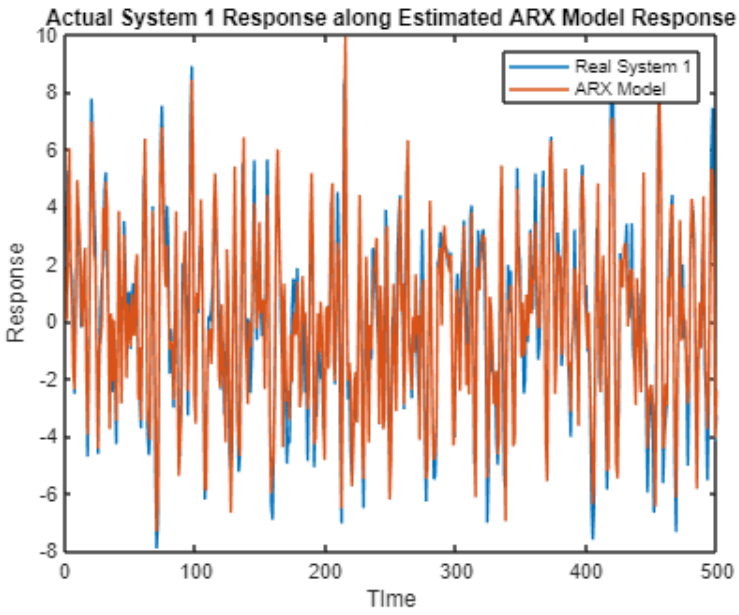
```
progress = 0.001;
prev_r2 = 0;
for degree=1:1:15
    na = degree;
    nb = degree;
    p = na + nb;

    U = arx_U_builder(na,nb,sys_1_iden_u,sys_1_iden_y);

    sys_1_theta_hat = inv(U'*U)*U'*sys_1_iden_y;
    y_hat = U*sys_1_theta_hat;

    t = (sys_1_iden_data.Tstart:sys_1_iden_data.Ts:N*sys_1_iden_data.Tstart);

    r2_arx = rSQR(sys_1_iden_y, y_hat);

    fprintf(">>> Degree = %d : R2=%f\n", degree, r2_arx)

    if (r2_arx-prev_r2)>progress
        prev_r2 = r2_arx;
    else
        fprintf("-------------------------------------------------------------\n")
        fprintf("Since the improvement of R2 metric value is less than %.4f,\nthe proper degree of the system is obtained as %d .\n", progress, degree)
        break;
    end
    fprintf("-------------------------------------------------------------\n")
end
```

```
>>> Degree = 1 : R2=0.377767
-------------------------------------------------------------
>>> Degree = 2 : R2=0.891014
-------------------------------------------------------------
>>> Degree = 3 : R2=0.908740
-------------------------------------------------------------
>>> Degree = 4 : R2=0.910575
-------------------------------------------------------------
>>> Degree = 5 : R2=0.911021
-------------------------------------------------------------
Since the improvement of R2 metric value is less than 0.0010,
the proper degree of the system is obtained as 5 .
```

```
fprintf("=================================================================\n")
```

```
=================================================================
```

```
sys_1_degree = degree;
sys_1_na = sys_1_degree;
sys_1_nb = sys_1_degree;
U = arx_U_builder(sys_1_na,sys_1_nb,sys_1_val_u,sys_1_val_y);

sys_1_y_hat = U*sys_1_theta_hat;

t = (sys_1_iden_data.Tstart:sys_1_iden_data.Ts:N*sys_1_iden_data.Tstart);

sys_1_r2_arx = rSQR(sys_1_val_y, sys_1_y_hat);

t = (sys_1_iden_data.Tstart:sys_1_iden_data.Ts:N*sys_1_iden_data.Tstart);
figure(1)
plot(t,sys_1_val_y,t,sys_1_y_hat)
title("Actual System 1 Response along Estimated ARX Model Response")
xlabel("TIme")
ylabel("Response")
legend("Real System 1", "ARX Model")
```



4.2.2 ARX - system 2

```
N = length(sys_2_iden_u);
fprintf("====================Degree Extraction System 2=======================\n")
```

```
====================Degree Extraction System 2=======================
```

```
progress = 0.001;
prev_r2 = 0;
```

```
 for degree=1:1:15
     na = degree;
     nb = degree;
     p = na + nb;

     U = arx_U_builder(na,nb,sys_2_iden_u,sys_2_iden_y);

     sys_2_theta_hat = inv(U'*U)*U'*sys_2_iden_y;
     y_hat = U*sys_2_theta_hat;

     t = (sys_1_iden_data.Tstart:sys_1_iden_data.Ts:N*sys_1_iden_data.Tstart);

     r2_arx = rSQR(sys_2_iden_y, y_hat);

     fprintf(">>> Degree = %d : R2=%f\n", degree, r2_arx)

     if (r2_arx-prev_r2)>progress
         prev_r2 = r2_arx;
     else
         fprintf("-----------------------------------------------------------\n")
         fprintf("Since the improvement of R2 metric value is less than %.4f,\nthe proper degree of the
system is obtained as %d .\n", progress, degree)
         break;
     end
     fprintf("-----------------------------------------------------------\n")
 end
```

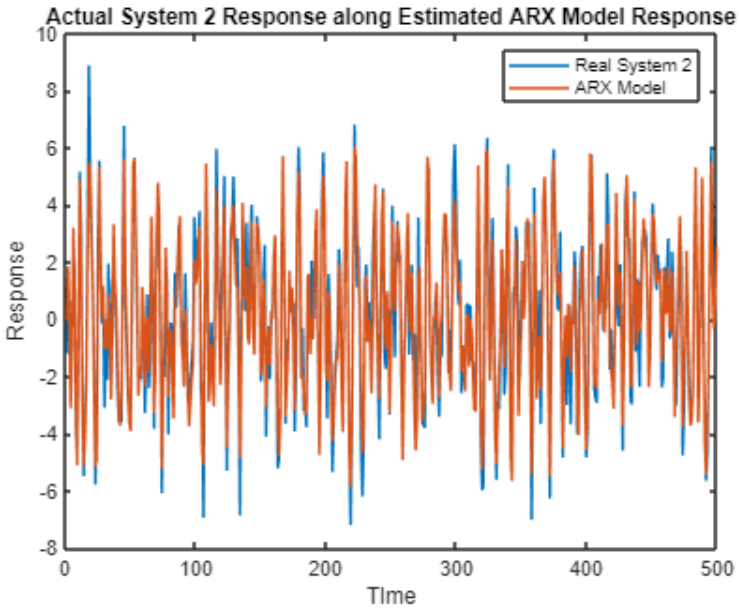
```
>>> Degree = 1 : R2=0.176378
-------------------------------------------------------------
>>> Degree = 2 : R2=0.748271
-------------------------------------------------------------
>>> Degree = 3 : R2=0.832194
-------------------------------------------------------------
>>> Degree = 4 : R2=0.839945
-------------------------------------------------------------
>>> Degree = 5 : R2=0.853096
-------------------------------------------------------------
>>> Degree = 6 : R2=0.864011
-------------------------------------------------------------
>>> Degree = 7 : R2=0.869402
-------------------------------------------------------------
>>> Degree = 8 : R2=0.869908
-------------------------------------------------------------
Since the improvement of R2 metric value is less than 0.0010,
the proper degree of the system is obtained as 8 .
```

```
 fprintf("=================================================================\n")
```

```
=================================================================
```

```
 sys_2_degree = degree;
 sys_2_na = sys_2_degree;
```

```
sys_2_nb = sys_2_degree;
U = arx_U_builder(sys_2_na,sys_2_nb,sys_2_val_u,sys_2_val_y);

sys_2_y_hat = U*sys_2_theta_hat;

t = (sys_2_iden_data.Tstart:sys_2_iden_data.Ts:N*sys_2_iden_data.Tstart);

sys_2_r2_arx = rSQR(sys_2_val_y, sys_2_y_hat);

t = (sys_2_iden_data.Tstart:sys_2_iden_data.Ts:N*sys_2_iden_data.Tstart);
figure(2)
plot(t,sys_2_val_y,t,sys_2_y_hat)
title("Actual System 2 Response along Estimated ARX Model Response")
xlabel("TIme")
ylabel("Response")
legend("Real System 2", "ARX Model")
```



>>> 4.3 - ARMAX for system 1 & 2

4.3.1 ARMAX - system 1

```
fprintf("=====================================================================\n")
```

```
=====================================================================
```

```
fprintf("=====================================================================\n")
```

```
=====================================================================
```

```
sys_1_na = sys_1_degree;
sys_1_nb = sys_1_degree;
sys_1_nk = 1;
sys_1_nc = 0;
```

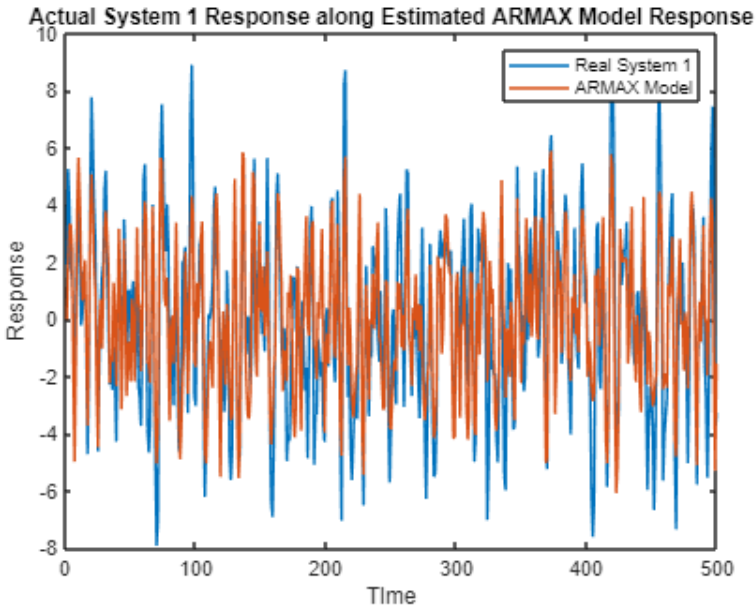
```
data = iddata(sys_1_iden_y, sys_1_iden_u, 1);
sys_1_armaxModel = armax(data, [sys_1_na, sys_1_nb, sys_1_nc, sys_1_nk]);

sys_1_y_hat = sim(sys_1_armaxModel, sys_1_val_u);
fprintf("R2 metric value:\n")
```

```
R2 metric value:
```

```
sys_1_r2_armax = rSQR(sys_1_val_y, sys_1_y_hat)
```

```
sys_1_r2_armax = 0.6646
```

```
t = (sys_1_iden_data.Tstart:sys_1_iden_data.Ts:N*sys_1_iden_data.Tstart);
figure(3)
plot(t,sys_1_val_y,t,sys_1_y_hat)
title("Actual System 1 Response along Estimated ARMAX Model Response")
xlabel("TIme")
ylabel("Response")
legend("Real System 1", "ARMAX Model")
```



4.3.2 ARMAX - system 2

```
fprintf("=================================================================\n")
```

```
=================================================================
```

```
data = iddata(sys_2_iden_y, sys_2_iden_u, 1);
sys_2_na = sys_2_degree;
sys_2_nb = sys_2_degree;
sys_2_nk = 1;
sys_2_nc = 0;
```

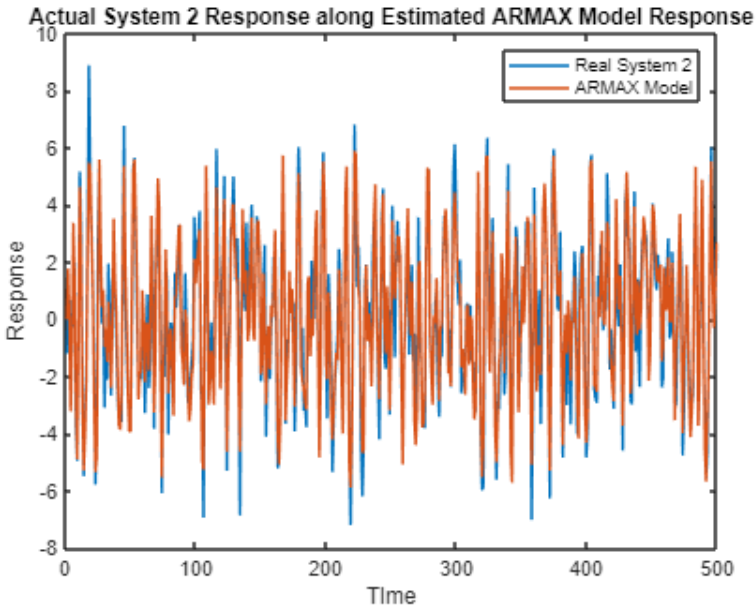
```
data = iddata(sys_2_iden_y, sys_2_iden_u, 1);
sys_2_armaxModel = armax(data, [sys_2_na, sys_2_nb, sys_2_nc, sys_2_nk]);

sys_2_y_hat = sim(sys_2_armaxModel, sys_2_val_u);
fprintf("R2 metric value:\n")
```

```
R2 metric value:
```

```
sys_2_r2_armax = rSQR(sys_2_val_y, sys_2_y_hat)
```

```
sys_2_r2_armax = 0.8928
```

```
t = (sys_2_iden_data.Tstart:sys_2_iden_data.Ts:N*sys_2_iden_data.Tstart);
figure(4)
plot(t,sys_2_val_y,t,sys_2_y_hat)
title("Actual System 2 Response along Estimated ARMAX Model Response")
xlabel("TIme")
ylabel("Response")
legend("Real System 2", "ARMAX Model")
```



>>> 4.4 - Box-Jenkins for system 1 & 2

4.4.1 Box-Jenkins - system 1

```
sys_1_na = sys_1_degree;
sys_1_nb = sys_1_degree;
sys_1_nk = 1;
sys_1_nc = 0;
fprintf("================================================================\n")
```

```
================================================================
```

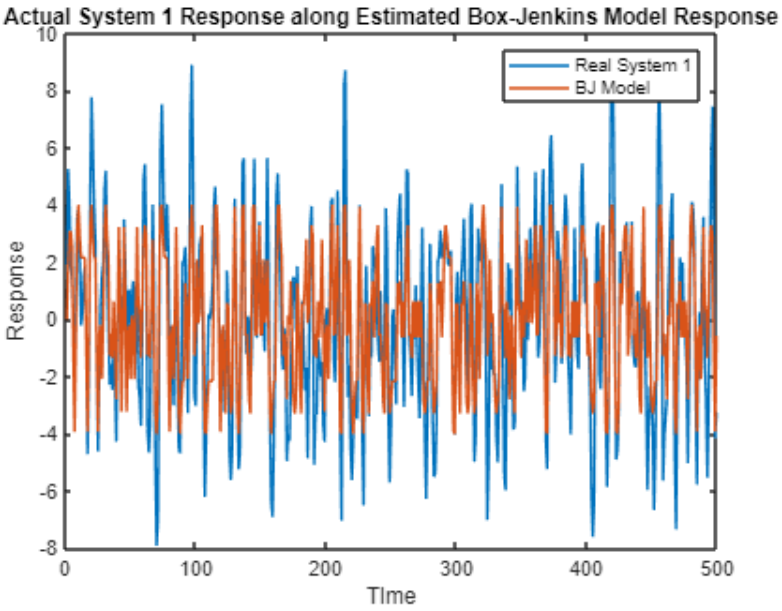
```
fprintf("=================================================================\n")
data = iddata(sys_1_iden_y, sys_1_iden_u, 1);
sys_1_BJModel = bj(data, [sys_1_na, sys_1_nb, sys_1_nc, 0, sys_1_nk]);



sys_1_y_hat = sim(sys_1_BJModel, sys_1_val_u);
fprintf("R2 metric value:\n")
```

```
R2 metric value:
```

```
sys_1_r2_bj = rSQR(sys_1_val_y, sys_1_y_hat)
```

```
sys_1_r2_bj = 0.6144
```

```


t = (sys_1_iden_data.Tstart:sys_1_iden_data.Ts:N*sys_1_iden_data.Tstart);
figure(5)
plot(t,sys_1_val_y,t,sys_1_y_hat)
title("Actual System 1 Response along Estimated Box-Jenkins Model Response")
xlabel("TIme")
ylabel("Response")
legend("Real System 1", "BJ Model")
```




```

```

4.4.2 Box-Jenkins - system 2

```
sys_2_na = sys_2_degree;
sys_2_nb = sys_2_degree;
sys_2_nk = 1;
sys_2_nc = 0;
fprintf("=================================================================\n")
```

```
=================================================================
```

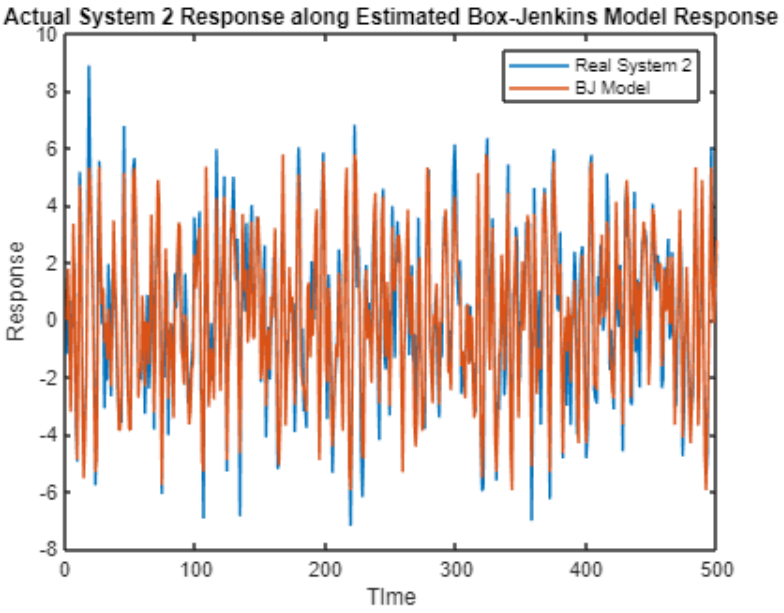
```
data = iddata(sys_2_iden_y, sys_2_iden_u, 1);
sys_2_BJModel = bj(data, [sys_2_na, sys_2_nb, sys_2_nc, 0, sys_2_nk]);

sys_2_y_hat = sim(sys_2_BJModel, sys_2_val_u);

fprintf("R2 metric value:\n")
```

```
R2 metric value:
```

```
sys_2_r2_bj = rSQR(sys_2_val_y, sys_2_y_hat)
```

```
sys_2_r2_bj = 0.8872
```

```
t = (sys_2_iden_data.Tstart:sys_2_iden_data.Ts:N*sys_2_iden_data.Tstart);
figure(6)
plot(t,sys_2_val_y,t,sys_2_y_hat)
title("Actual System 2 Response along Estimated Box-Jenkins Model Response")
xlabel("TIme")
ylabel("Response")
legend("Real System 2", "BJ Model")
```



>>> 4.5 - Comparison

>>> System 1:

ARX R2 = 0.91

ARMAX R2 = 0.66

Box-Jenkins R2 = 0.61

>>> System 2:

ARX R2 = 0.86

ARMAX R2 = 0.89

Box-Jenkins R2 = 0.88

Helper Functions - Appendix

>>> arx_U_builder

```
function U = arx_U_builder(n,m,u,y)

    N = length(y);
    p = n+m;
    U = zeros(N,p);

    for i=1:N
        for j=1:n
            if (i-j)>0
                U(i,j) = -y(i-j);
            else
                continue
            end
        end
        for j=n+1:p
            k=j-(n+1);
            if (i-k)>0
                U(i,j) = u(i-k);
            else
                continue
            end
        end
    end
end
```

>>> rSQR

```
function [r2, mse] = rSQR(y, y_hat)
    error = y - y_hat;

    sse = sum(error.^2);
    mse = mean(error.^2);

    sst = sum((y - mean(y)).^2);
    r2 = 1 - sse / sst;
end
```

>>> RLS

```
function [theta, U] = RLS(na, nb, th0, p0, u, y)

    p = na + nb;
    N = length(y);
    theta = th0;
    P = p0 * eye(na + nb);

    theta_estimate=th0';

    U=[];
    Y=[];

    for i=1:N
        Y = [Y ; y(i)];

        Ut=zeros(1,p);
        Y1=y(i);
        for j=1:na
            if (i-j>0)
                Ut(1,j) = -y(i-j);
            end
        end
        for j=na+1:na+nb
            k=j-na;
            if (i-k>0)
                Ut(1,j) = u(i-k);
            end
        end

        U=[U;Ut];
        Kt=(P*Ut')/(1+Ut*P*Ut');
        theta = theta + Kt*(Y1 - Ut*theta);
        theta_estimate = [theta_estimate ; theta'];
        P = (eye(na+nb) - Kt*Ut) * P;
    end

end
```