CHE 572

Assignment #5

(Solution)

Course Instructor: Dr. Fraser Forbes

SOLUTION OF QUESTION NO. 1

There are two approaches to solve this problem. One is to assume that we don't have a priori information about the type of the model and identify the best possible model using the given data. The second approach is to assume that the order of the model is known and find the parameters of the model by using system identification methods. Here the second approach has been chosen. Therefore a model of the following form is fitted to the data:

$$y_t = \frac{\left(1 - C_1 z^{-1}\right)}{\left(1 + D_1 z^{-1} + D_2 z^{-2}\right)} \varepsilon_t$$

a)

Data Set 1:

The first set of raw data is given in Figure #1. By using the "dtrend" command of MATLAB the mean is removed (Figure #2) and using the following command an ARMA model is fitted to it.

```
model=armax(Y,[2,1])
```

The model obtained is as follows:

$$y_t = \frac{\left(1 - 0.01331 z^{-1}\right)}{\left(1 - 0.742 z^{-1} + 0.1329 z^{-2}\right)} \varepsilon_t = \frac{\left(1 - 0.01331 z^{-1}\right)}{\left(1 - 0.3021 z^{-1}\right)\left(1 - 0.4399 z^{-1}\right)} \varepsilon_t$$

The autocorrelation function plot of the residuals is given in Figure #3. The plot can be obtained by " resid(model,Y)" command.

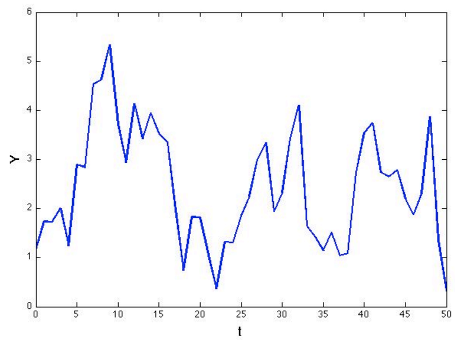


Fig # 1: Plot of given data

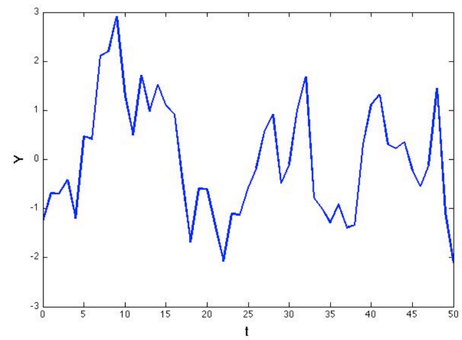


Fig # 2: Plot of given data after removing the mean

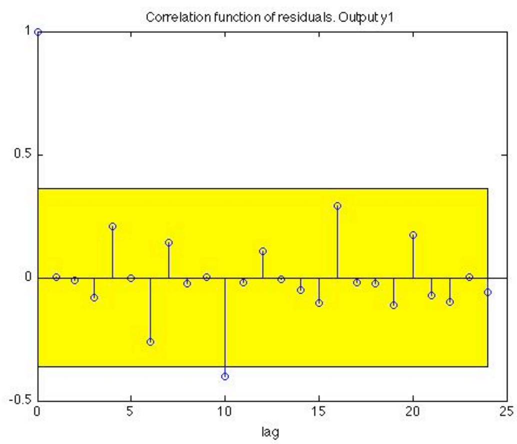


Fig # 3 : Autocorrelation (residual) plot of the model residuals

Note: No cross-correlation, since there is no input data

It can be noted here that, all the points in the autocorrelation plot are within the 95% confidence band.

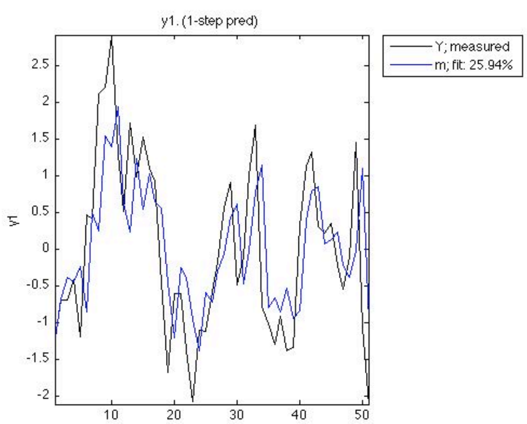


Fig # 4: Plot of the actual output and the estimated output

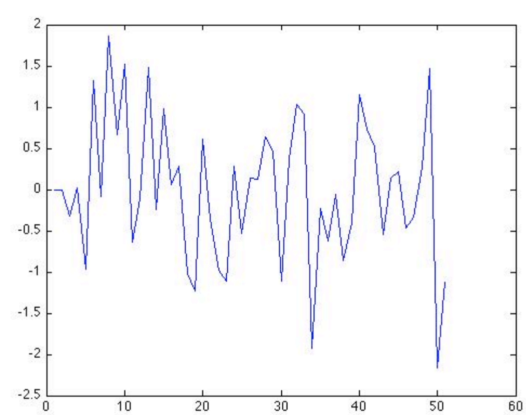


Fig # 5: Plot of model residual

Figure #4 compares the predicted output and the given data. The plot is obtained using "compare" command of MATLAB. On the other hand, Figure #3 shows the prediction errors obtained by the following commands:

```
e=pe(Y,model);
plot(e)
```

The covariance matrix of the parameters can be found by the following command:

```
model.CovarianceMatrix
```

and the results is:

$$Cov(p) = \begin{pmatrix} 1.6388 & -1.0598 & 1.6269 \\ -1.0598 & 0.6985 & -1.0523 \\ 1.6269 & -1.0523 & 1.6385 \end{pmatrix}$$

Therefore the standard deviations of the parameters are:

Table #1: Parameter Estimates for data set 2

Parameter	Value	Standard deviation
C_1	-0.01331	1.28
D_1	-0.742	1.2802
D_2	0.1329	0.8358

Note that the standard deviation can also be found by using the following command

```
Model.da  % % gives the standard deviations of the autoregressive part
Model.dc %% gives the standard deviations of the moving average part
```

Similarly, two new models are obtained from the remaining two data sets.

<u>Data Set 2:</u>

$$y_t = \frac{\left(1 - z^{-1}\right)}{\left(1 - 1.732z^{-1} + 0.7367z^{-2}\right)}\varepsilon_t = \frac{\left(1 - z^{-1}\right)}{\left(1 - 0.7509z^{-1}\right)\left(1 - 0.9811z^{-1}\right)}\varepsilon_t$$

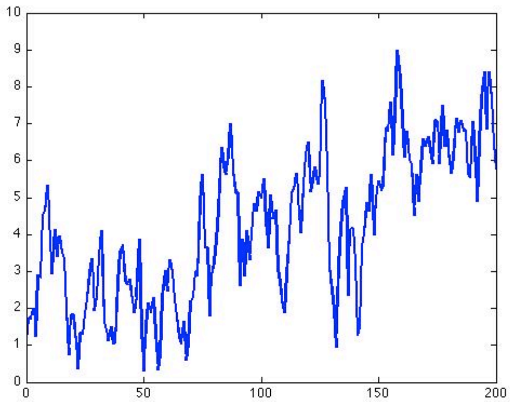


Fig # 6: Plot of given data

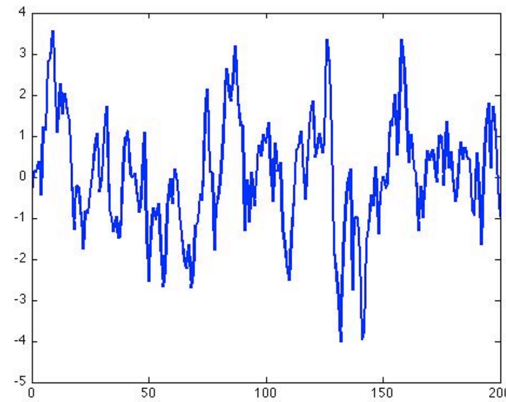


Fig # 7: Plot of given data after removing the mean

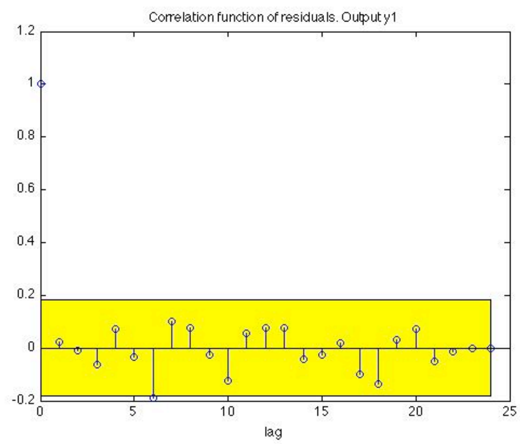


Fig # 8: Autocorrelation plot of the model residuals

Note: No cross-correlation, since there is no input data

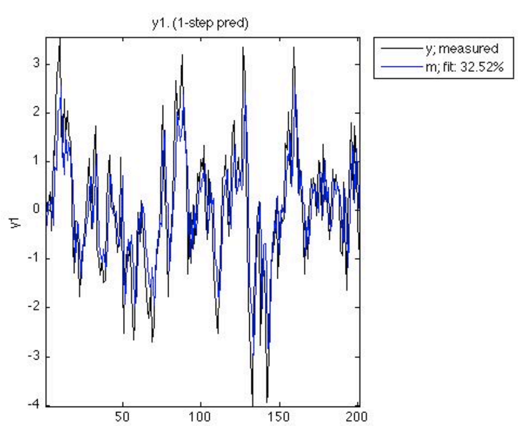


Fig # 9: Plot of the actual output and the estimated output

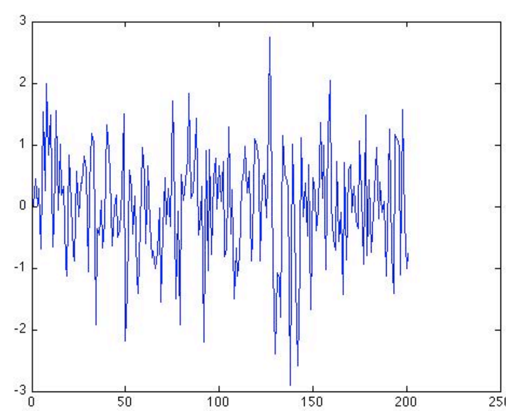


Fig # 10: Plot of model residual

The covariance matrix is:

$$Cov(p) = \begin{pmatrix} 0.0027 & -0.0025 & 0.0005 \\ -0.0025 & 0.0024 & -0.0003 \\ 0.0005 & -0.0003 & 0.0008 \end{pmatrix}$$

Table #2: Parameter Estimates for data set 2

Parameter	Value	Standard deviation
C_1	-1	0.0277
D_1	-1.732	0.0515
D_2	0.7367	0.0494

Data Set 3:

$$y_t = \frac{\left(1-0.8966z^{-1}\right)}{\left(1-1.779z^{-1}+0.7797z^{-2}\right)}\varepsilon_t = \frac{\left(1-0.8966z^{-1}\right)}{\left(1-0.7822z^{-1}\right)\left(1-0.9968z^{-1}\right)}\varepsilon_t$$

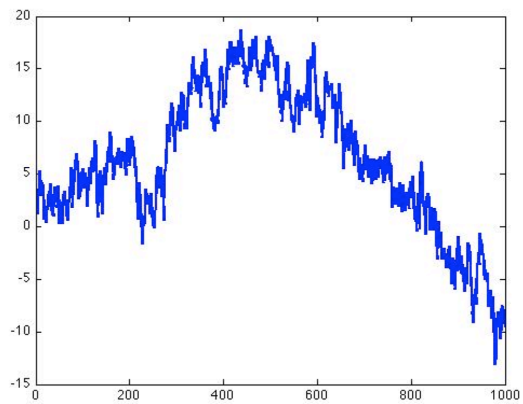


Fig # 11: Plot of given data

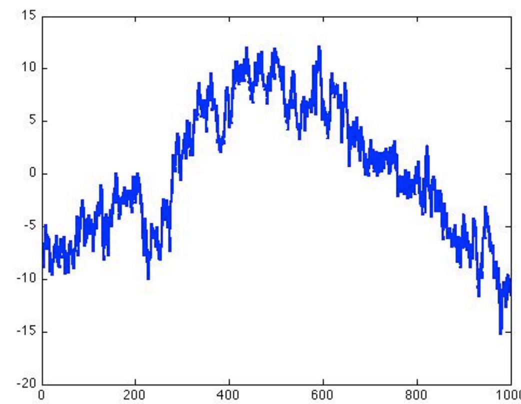


Fig # 12: Plot of given data after removing the mean

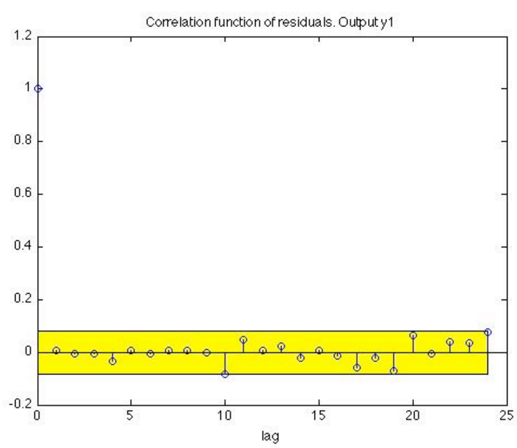
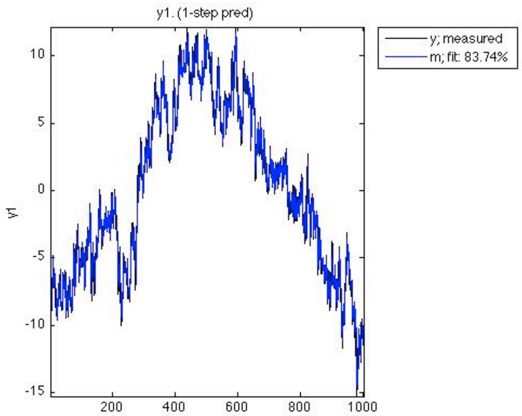


Fig # 13: Autocorrelation plot of the model residuals



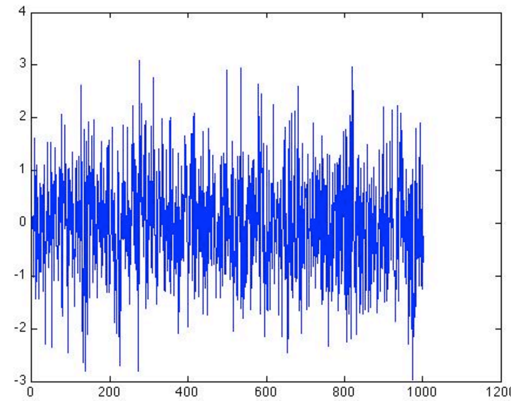


Fig # 14 : Plot of the actual output and the estimated output

Fig # 15 : Plot of model residual

The covariance matrix is:

$$Cov(p) = \begin{pmatrix} 0.0029 & -0.0029 & 0.0019 \\ -0.0029 & 0.0029 & -0.0019 \\ 0.0019 & -0.0019 & 0.0015 \end{pmatrix}$$

Table #3: Parameter Estimates for data set 3

Parameter	Value	Standard deviation
C_1	-0.8966	0.0388
D_1	-1.779	0.0540
D_2	0.77797	0.0537

Table #4: Parameter Estimates for all three non-differenced data sets

Parameter	a		b		c	
	Value	SD	Value	SD	Value	SD
C_1	-0.01331	1.28	-1	0.0277	-0.8966	0.0388
D_1	-0.742	1.2802	-1.732	0.0515	-1.779	0.0540
D_2	0.1329	0.8358	0.7367	0.0494	0.77797	0.0537

(b)

In this part we are differencing the data (i.e. *y*). So, the new predicted model will be:

$$y_t = \frac{\left(1 - C_1 z^{-1}\right)}{\left(1 + D_1 z^{-1}\right)} \varepsilon_t$$

We will be getting similar kind of plots for this part as before. The resulting model parameters and their standard deviations are given in Table 5.

Table #5 : Parameter Estimates for all three differenced data sets

		51 Data	**201 Data**	**1001 Data**
C_1	*Parameter*	-0.9752	-0.9425	-0.9088
	Standard Deviation	0.0445	0.0362	0.0461
D_1	*Parameter*	-0.7798	-0.7336	-0.7898
	Standard Deviation	0.1266	0.0728	0.0313

(c, d)

From Tables #4 and #5, it is evident that as the number of data points increase; we get better estimates of the model parameters. This improvement is both in terms of the bias (*i.e.*, the difference between the parameter estimates and their true values) and the variance of these estimates. The covariance matrix for the parameter estimates is proportional to the variance of the prediction errors, which is estimated as:

$$\sigma_e^2 = \frac{1}{N-p}\varepsilon^{\mathrm{T}}\varepsilon$$

Thus, the variance of the parameter estimates is inversely proportional to the number of data points available for estimating them. This is not quite borne out by the standard deviations given by Matlab, but you must remember that these standard deviations do not really incorporate the effects of the covariance between the individual estimates.

Also by comparing Table 4 and Table 5, one can observe that differencing the data gives more accurate results, especially for cases that the number of data point is limited. For example in the case of 51 data points, the uncertainty in the identified model using non-differenced data is significant. The values of standard deviations are bigger than the values of parameters and the 95% confidence interval includes 0 for all parameters.

SOLUTION OF QUESTION NO. 2

(a)

In this question we have been given a large set of input-output data of a Single-Input-Single-Output (SISO) process. Our target will be to estimate a model.

The first step in identification is to build an iddata object of the given data by the following command:

```
data= iddata(y,u,1)
```

Then we should look at the raw data (See Figure #1). From it, we notice that there is some random walk in the output data.

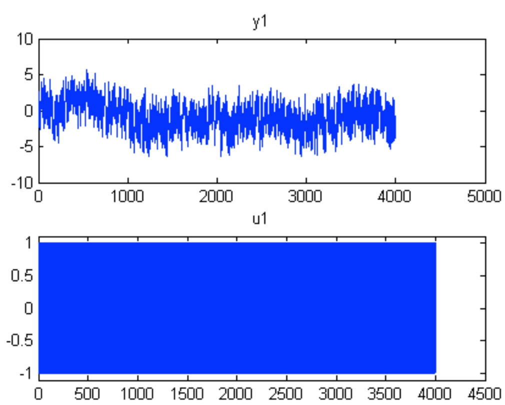


Fig # 1: The given input-output raw data

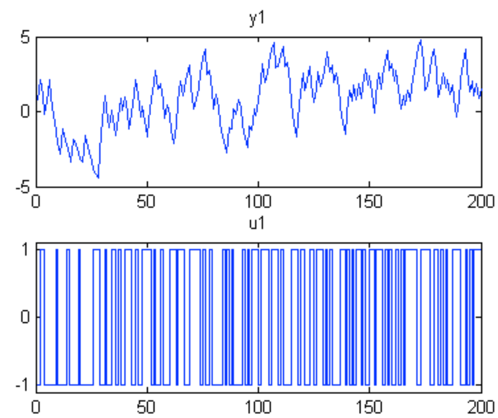


Fig # 2: A section of the given data

To further study the data, we zoom into the data (see Figure #2). From it, we notice that there is some correlation between the input data and the output data. To know the exact correlation, we use the Matlab command '*cra*' on the data (see Figure #3). But, before that we do have to remove the mean from the data(using dtrend command).

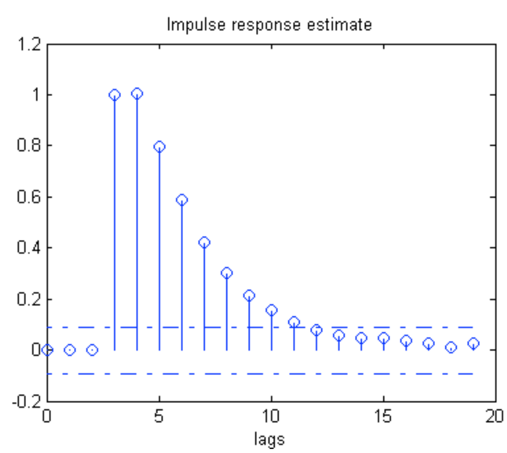


Fig # 3 : The impulse response of the process from 'cra'

From Figure #3, we notice that the process has 3 unit time-delay. Furthermore, it looks like a first order process or an over-damped second order process.
The autocorrelation function (see Figure #4) and partial autocorrelation function (see Figure #5) are given below. To obtain these graphs the following commands have been use:

```
autocf(data.outputdata,20,0)
pautocf(data.outputdata,20)
```

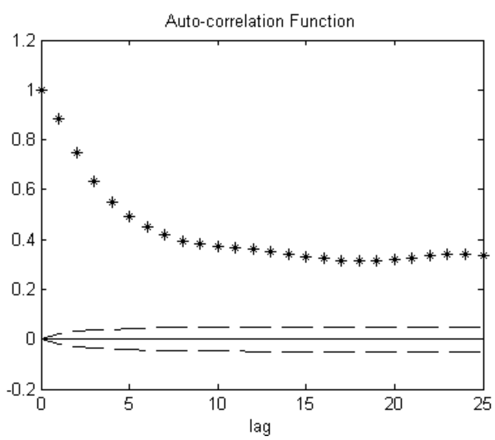


Fig # 4: Autocorrelation of the output data

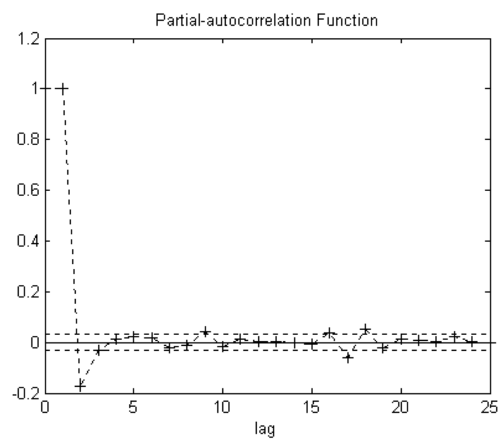


Fig # 5: Partial-Autocorrelation of the output data

All the points in the autocorrelation (see Figure #4) are outside the confidence band, which indicates that the data probably should be differenced. On the other hand, there are two points (at 1 and 2) outside the confidence band. One is at one and the other one is below the confidence band. So, we can say that, the AR part of the model might be of second order.

The general procedure, which I follow to fit these models, is to: 1) fit the plant model; 2) identify the noise model using the residuals produced from the subtracting the predicted outputs from the measure outputs (using "pe" command); 3) use the original data to re-estimate all of the parameters in the full plant+noise model; and 4) validate the full model by checking a variety of plots. This procedure is repeated until I am satisfied with the fit.

We will start with a 1st order model for the plant. A model can be identified by the following commands

```
data=diff(data) %% for differencing the data (it will difference both input and output)
nb=1; nc=0; nd=0; nf=1; nk=3;%% nc and nd are zero, because we are interested in plant model only
model=bj(data,[nb nc nd nf nk])
reside(data,mode)
```

The residual plot is given in Figure 6.

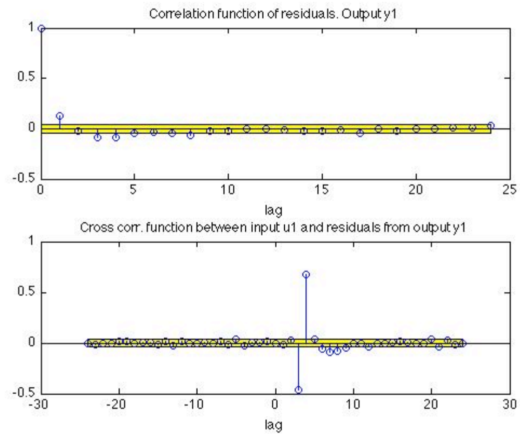


Figure #6, Residual plot of BJ model [1 0 0 1 3]

Some points are outside the 95% confidence interval. Then we try to increase nf to 2.

The residual is given in Figure 7.

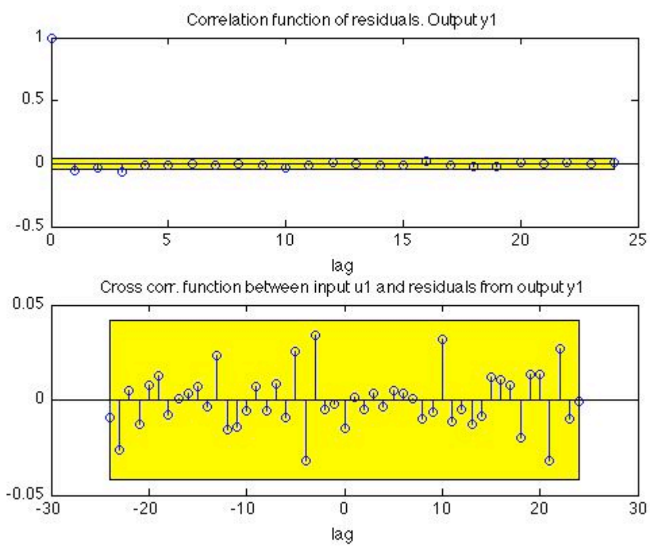


Figure #7, Residual plot of BJ model [1 0 0 2 3]

Now residuals look fine. Using pe command we can calculate the prediction error.

e=pe(data,model).

Now we can identify a noise model for the prediction errors. The autocorrelation and partial auto correlation plot of the error are given in Figures 8 and 9. We can observe that both of them die off but none of them truncates. We attempt to fit an ARMA model with AR order of 1 and MA order of 1.

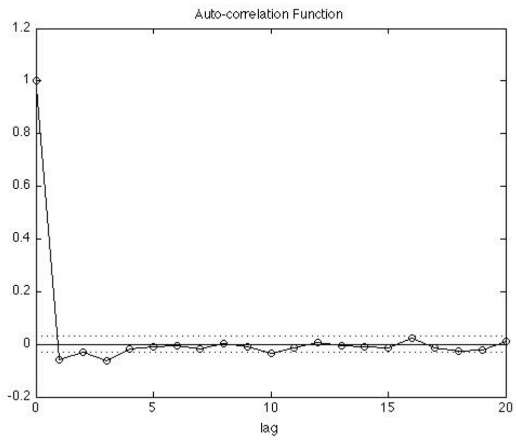


Figure #8: Autocorrelation function of residuals

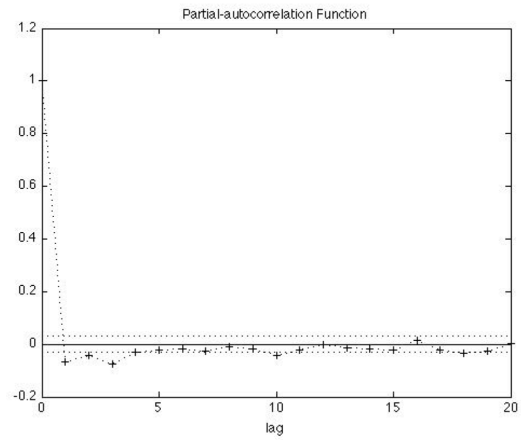


Figure #9: Partial autocorrelation function of residuals

The noise model can be identified by the following command:

```
noise=armax(e,[1 1])
```

In order to validate the noise model, we should calculate the residuals of this noise model and plot autocorrelation and partial autocorrelation graphs (Figures 10 & 11)

```
e2=pe(noise,e)
autocf(e2,20,0)
pautocf(e2,20)
```

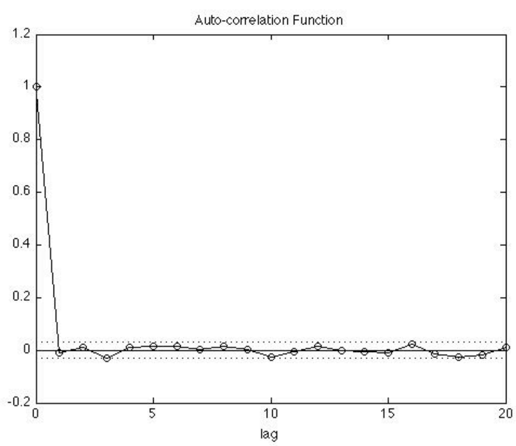


Figure #10: Autocorrelation function of residuals of the noise model

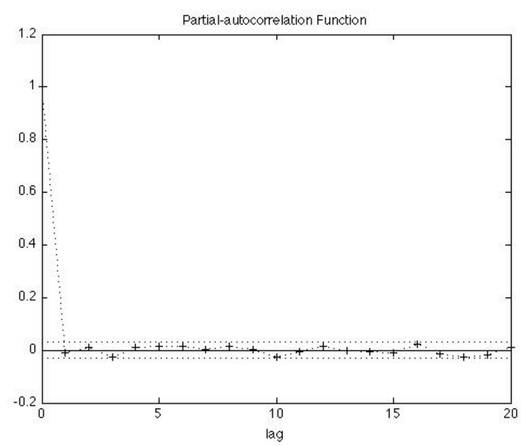


Figure #11: Partial autocorrelation function of residuals of the noise model

Now we can combine the noise and plant models using the bj command of form [1 1 1 2 3]. The resulting model will be

$$y_t = \frac{z^{-3}}{\left(1-1.001z^{-1}+0.2097z^{-2}\right)}u_t + \frac{\left(1+0.9672z^{-1}\right)}{\left(1+0.9723z^{-1}\right)\left(1-z^{-1}\right)}\varepsilon_t \qquad (2.1)$$

or

$$y_t = \frac{z^{-3}}{\left(1-0.7025z^{-1}\right)\left(1-0.2985z^{-1}\right)} u_t + \frac{\left(1+0.9672z^{-1}\right)}{\left(1+0.9723z^{-1}\right)\left(1-z^{-1}\right)} \varepsilon_t$$

Table 2.1: Parameters and their Standard deviation

B_1	*Parameter*	1
	Standard Deviation	0.0016
C_1	*Parameter*	0.9672
	Standard Deviation	0.0469
D_2	*Parameter*	0.9723
	Standard Deviation	0.0432
F_1	*Parameter*	-1.001
	Standard Deviation	0.0015
F_2	*Parameter*	0.2097
	Standard Deviation	0.0015

It should be noted from Table #1 that none of the parameters have the zero in their $\pm 2\sigma$ range. So, we can say that, none of the above parameters is zero. Note that the actual model used to produce this data was:

$$y_t = \frac{z^{-3}}{\left(1-0.7z^{-1}\right)\left(1-0.3z^{-1}\right)} u_t + \frac{\left(1-0.9z^{-1}\right)}{\left(1-\ 0.85z^{-1}\right)\left(1-z^{-1}\right)} \varepsilon_t$$

Figure 12 compares the predicted output and the actual data.

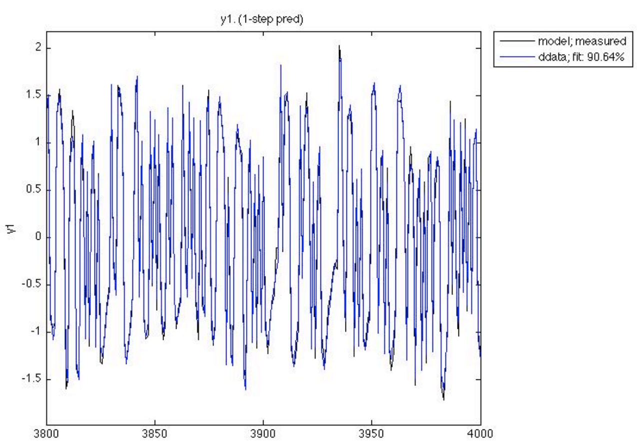
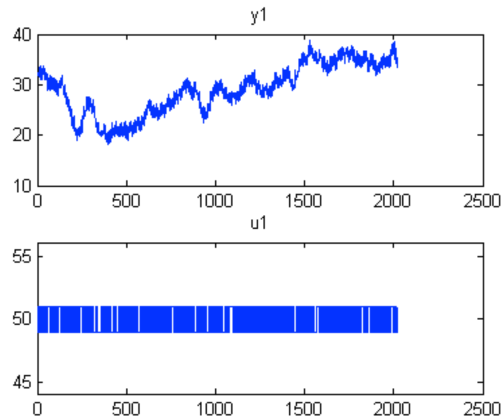


Fig # 12: Actual data and the data generated by the simulation

(b)

As before, the first step in identification should be to look at the raw data (See Figure #13). From it, we notice that there is random walk in the output data.



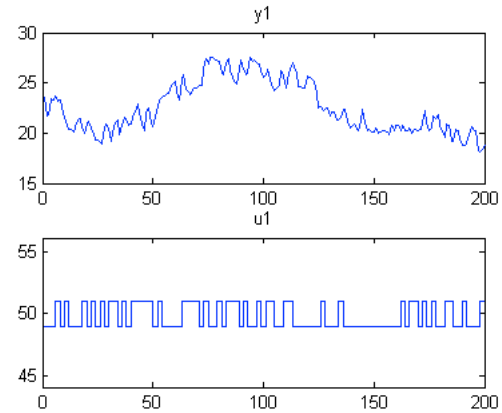


Fig # 13 : The given input-output raw data   *Fig # 14 : A section of the given data*

After detrending the data, we can use “cra” command to identify the time delay of the process. The result is given in Figure 15.

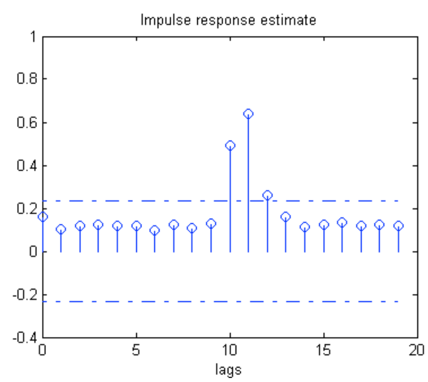


Fig # 15 : The impulse response of the process from ‘cra’

From Figure #15, we notice that the process has a 10 unit time-delay. Furthermore, it looks like a first order process or a second order over-damped process.

The autocorrelation function (see Figure #16) and partial autocorrelation function (see Figure #17) are given below.

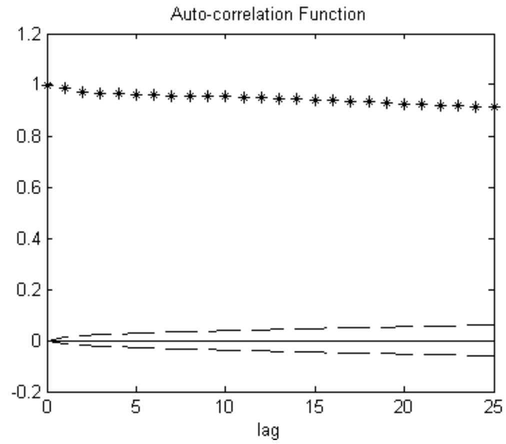


Fig # 16 : Autocorrelation of the output data

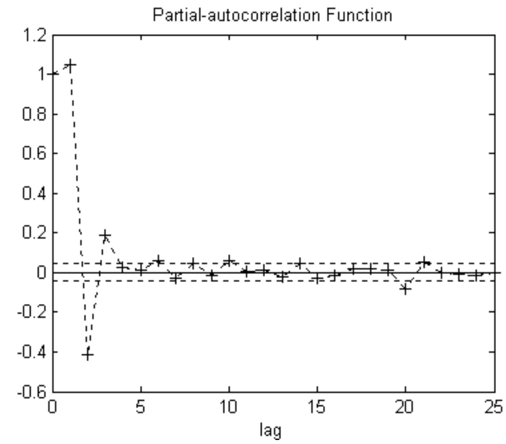


Fig # 17 : Partial-Autocorrelation of the output data

All the points in the autocorrelation (see Figure #16) are outside the confidence band, which indicates that the data may need to be differenced. Figure 18 is the plot of the differenced data.

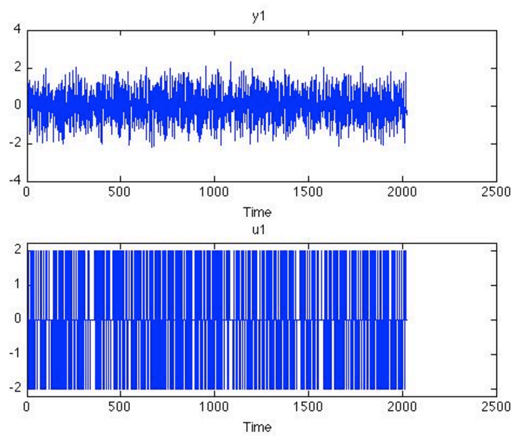


Fig #18: plot of the differenced data

Following the same procedure as outlined in part 2(a) we start from a plant model of order 1 (bj of form [1 0 0 1 10]). The resulting residual plot is given in Figure 19.

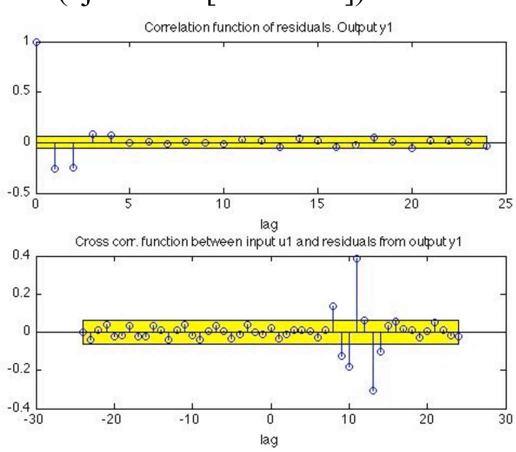


Fig. #19: Residual of BJ model of form [1 0 0 1 10]

Since there are some points outside the 95% confidence interval, we try to increase the orders. We tried: [2 0 0 1 3]. The resulting residuals are:

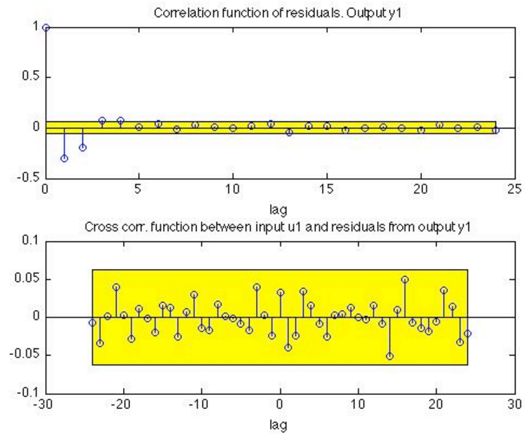


Fig. #20: residuals of BJ of form [2 0 0 1 10]

By increasing the orders of denominator and numerator in plant model we will so no improvement. So we try to fit a noise model. First we should plot the autocorrelation and partial autocorrelation plots of the prediction error (Figures 21 & 22).

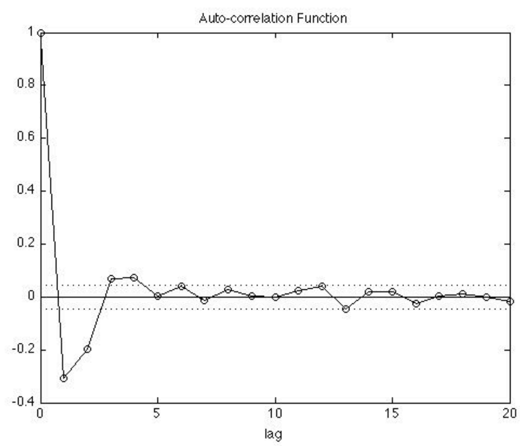


Fig. # 21: Autocorrelation function of residuals

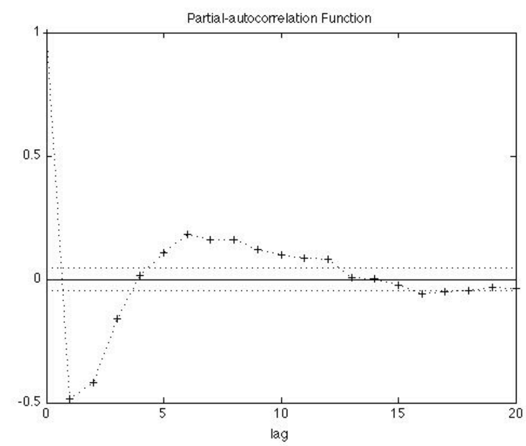


Fig. # 22: Partial Autocorrelation function of residuals

Based on Figures 21 & 22 we can see that the Autocorrelation function truncates at lag 4. So choose a moving average model of order 4. The residual plot of the noise model is given in Figure 23. We can observe that all of the points are inside the 95% confidence interval.

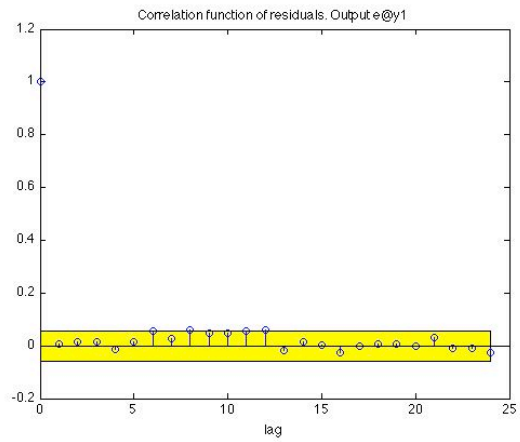


Fig. # 23: Residual of the noise model

Finally we can combine the identified plant model and noise model by a BJ function of order [2 4 0 1 10]. The resulting model is:

$$y_t = \frac{0.385z^{-10} + 0.4136z^{-11}}{\left(1 - 0.2723z^{-1}\right)} u_t + \frac{\left(1 - .4712z^{-1} - 0.1888z^{-2} + 0.1324z^{-3} + 0.1258z^{-4}\right)}{\left(1 - z^{-1}\right)} \varepsilon_t \tag{2.2}$$

Note that the zero in the plant transfer function probably comes from a partial period of delay.

Table 2.2: Parameters and their Standard deviation

B_1	*Parameter*	0.385
	Standard Deviation	0.011
B_2	*Parameter*	0.4136
	Standard Deviation	0.0162
C_1	*Parameter*	-0.4712
	Standard Deviation	0.0221
C_2	*Parameter*	-0.1888
	Standard Deviation	0.0243
C_3	*Parameter*	0.1324
	Standard Deviation	0.0243
C_4	*Parameter*	0.1258
	Standard Deviation	0.0221
F_1	*Parameter*	-0.2723
	Standard Deviation	0.0173

It should be noted that none of the parameters have the zero in their $\pm 2\sigma$ range. So, we can say that, none of the above parameters is likely to be zero. Note that the standard deviations of variable can be found by the following commands:

model.df %% standard deviation of the parameters of polynomial F
model.db %% standard deviation of the parameters of polynomial B
model.dc %% standard deviation of the parameters of polynomial C
model.dd %% standard deviation of the parameters of polynomial D

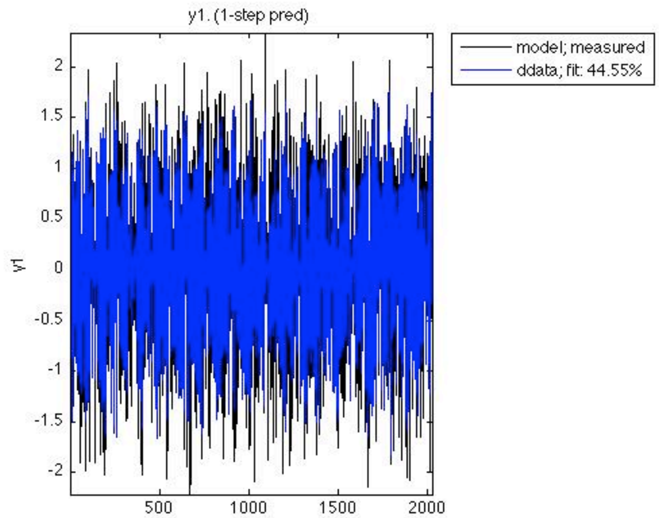


Fig # 19 : A section of the actual data and the estimated data plotted together

By looking at the inputs for cases a and b, we notice that the switching frequency of the input is equal to the sampling frequency, which is a little fast to get good estimates of slow dynamics. On the other hand the switching frequency for case b is half of the sampling frequency and is more appropriate than case a.