CH E 572

Assignment #4

(Solution)

Course Instructor: Dr. Fraser Forbes

ANSWER TO QUESTION NO. 1

(a)

The first step would be to plot the data:

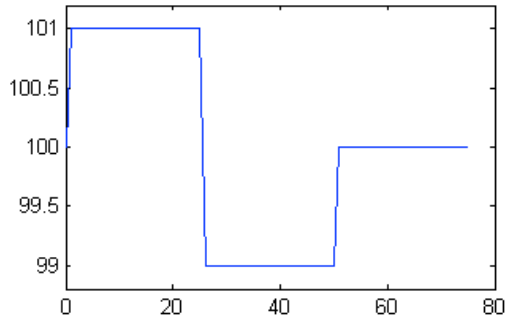

*Fig # 1 : Plot of flow rate (**f**) vs. Time (**t**)*

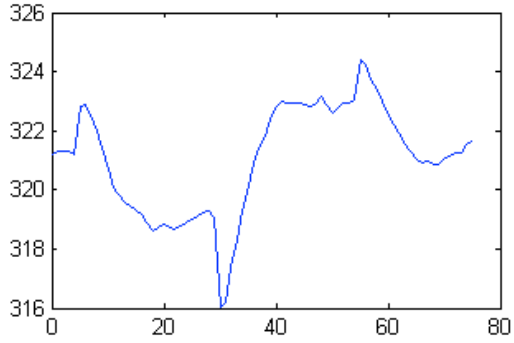

*Fig # 2 : Plot of Temperature (**f**) vs. Time (**t**)*

It can be noted from the plot that the data is quite noisy. Estimating the dead time from a noisy data can be quite difficult.

Remove the means from the input and output data and plot them on the same plot using the following commands

```
f1=f-mean(f);
T1=T-mean(T);
figure; plot (t,T-mean(T)); hold; plot (t,f-mean(f));
```

The Matlab command '*detrend*' can also be used instead of subtracting the mean this way.

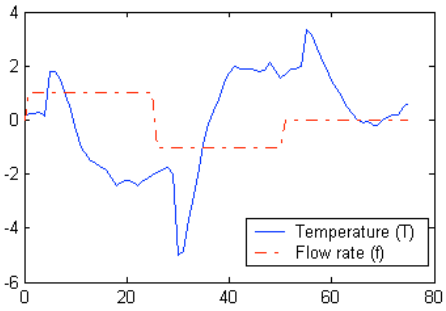

*Fig # 3 : Plot of 'detrend'-ed flow rate (**f**) and Temperature (T) vs. Time (**t**)*

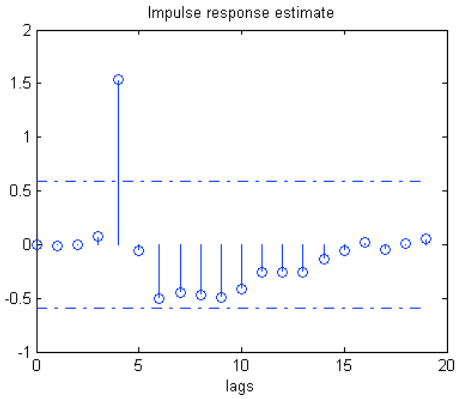

Fig # 4 : 'cra' of the given input-output data

We can zoom into Fig #3 to get an estimate of the time delay (4 unit time). But, as a confirmation, we can also look at the results from Matlab command 'cra' (see Fig #4). 'cra' clearly shows the delay to be 4 unit times.

The questions says that the process is a "second-order + dead-time" model. So, the discrete time model will be:

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + a_3 u_{k-4} \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (1.1)$$

Please note, u has a subscript of $(k-4)$.

Rewriting this Equation 1.1 in matrix form:

$$y_k = \begin{bmatrix} y_{k-1} & y_{k-2} & u_{k-4} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (1.2)$$

We know that for a linear equation of the form, $Y = X\beta$, the value of the parameters can be estimated by, $\beta = \left(X^T X\right)^{-1}\left(X^T Y\right)$

The X and Y matrix for our case will be as follows:

$$X = \begin{bmatrix} & & \\ T_0 & 0 & 0 \\ T_1 & T_0 & 0 \\ T_2 & T_1 & 0 \\ T_3 & T_2 & f_0 \\ \vdots & \vdots & \vdots \\ T_{n-1} & T_{n-2} & f_{n-4} \end{bmatrix} \text{ and, } Y = \begin{bmatrix} \\ T_1 \\ T_2 \\ T_3 \\ T_4 \\ \vdots \\ T_n \end{bmatrix}$$

The commands used to generate the matrices and to estimate β in Matlab are:

```
X=[T1(1:end-1) [0; T1(1:end-2)] [ 0; 0; 0;  f1(1:end-4)] ];
Y=T1(2:end);
B=inv(X'*X)*(X'*Y)
```

This gives the following estimates of the parameters:

$$\beta = \begin{bmatrix} 1.2498 \\ -0.3297 \\ -0.1407 \end{bmatrix}$$

Now, we can use the parameter to compare our estimates with the actual data. We calculate the estimated temperature, $\hat{T}$ (in Matlab it is shows as *Th*) and the residual, r using Matlab command:

```
Th=X*B;
r= T1-Th;
```

Then we plot the actual temperature with the estimated temperature (See Fig #5) and, the residual (see Fig #6) using the Matlab commands:

```
figure; plot (T)
hold; plot (Th+mean(T), 'r-.')
legend ('Temperature Actual (T)', 'Temperature
Estimated (Th)')
figure; plot (r)
```

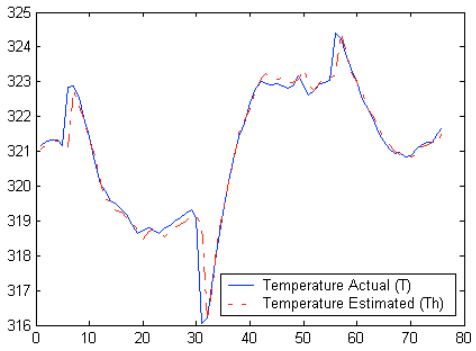

Fig # 5 : Plot of actual temperature, T and estimated temperature, $\hat{T}$ with time

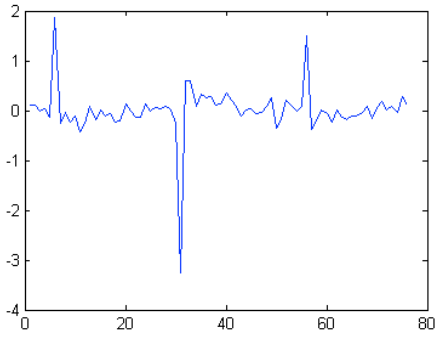

Fig # 6 : Plot of model residual, r with time

We now want to know how accurate our parameters are. We will have to calculate the variance of the residual, $\sigma_\varepsilon^2 = \text{var}(y - \hat{y}) = \dfrac{1}{N-p}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, and then the covariance matrix of the parameters, $\text{var}(\hat{\beta}) = (X^T X)^{-1}\sigma_\varepsilon^2$

We use the following Matlab commands to do this,

```
SigmaSquare = (1/(75-3))*(r'*r)
VarB = (1/(75-3))*inv(X'*X)*(r'*r)
```

Here, `SigmaSquare` stands for σ_ε^2, and `VarB` stands for $\text{var}\left(\hat{\beta}\right)$. And, the results are:

$$\sigma_\varepsilon^2 = 0.2651$$

$$\text{var}\left(\hat{\beta}\right) = \begin{bmatrix} 0.0122 & -0.0114 & 0.0018 \\ -0.0114 & 0.0118 & -0.0011 \\ 0.0018 & -0.0011 & 0.0060 \end{bmatrix}$$

Based on the two estimates (σ_ε^2 and $\text{var}\left(\hat{\beta}\right)$) given above, it appears that the parameters are all non-zero.

(b)

Based on Figure #5 it can be stated that the "second-order + time-delay" model gives quite a good fit to the given data. But, when we check the residual (Figure #6), we find that there are 'spikes' in the plots where the step changes are made.

We can go back to Figure #1 to understand the underlying reason for this. In this reactor/cooler system, when the flow rate of coolant in increased, the temperature does not go down. The temperature increases at first and then it decreases. Similarly, when the coolant flow rate is reduced, the temperature goes down first and then starts to climb.

This means the process has an inverse response. In other words, the process has a zero in its model. To get a better fit, we need to add a zero to our estimated model. We could use 'cra' to see the dependence of residuals on inputs too.

(c)

Based on the idea explained in (b), we add a zero to our model:

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + a_3 u_{k-5} \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (1.3)$$

Rewriting this Equation 1.3 in matrix form:

$$y_k = \begin{bmatrix} y_{k-1} & y_{k-2} & u_{k-4} & u_{k-5} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad \ldots \quad \ldots \quad \ldots \quad (1.4)$$

The *X* and *Y* matrix for our case will be as follows:

$$X = \begin{bmatrix} T_0 & 0 & 0 & 0 \\ T_1 & T_0 & 0 & 0 \\ T_2 & T_1 & 0 & 0 \\ T_3 & T_2 & f_0 & 0 \\ T_4 & T_3 & f_1 & f_0 \\ \vdots & \vdots & \vdots & \vdots \\ T_n & T_{n-1} & f_{n-4} & f_{n-5} \end{bmatrix} \text{ and, } Y = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ \vdots \\ T_n \end{bmatrix}$$

The estimated parameters are:

$$\beta = \begin{bmatrix} 1.15 & -0.295 & 1.50 & -1.79 \end{bmatrix}^T$$

Using the same kind of Matlab command as before we generate the following plots:

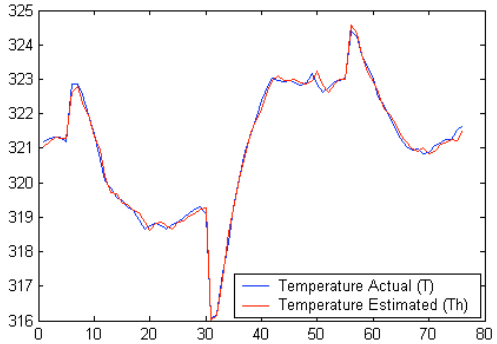

Fig # 7 : Plot of actual temperature, T and estimated temperature, $\hat{T}$ (from modified model) with time

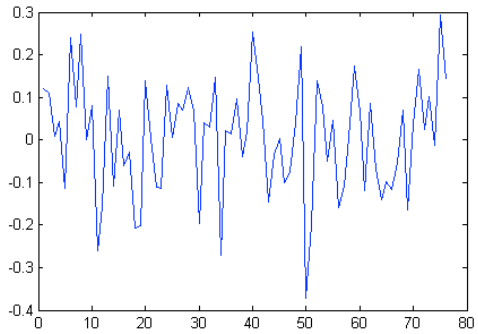

Fig # 8 : Plot of model residual, r (from modified model) with time

If we compare Figure #5 and Figure #7, it will be very difficult to see the improvement due to the modification of the model. But, comparison of Figure #6 and Figure #8 shows the true picture. In Figure #6 we had spikes in the residuals. But, in Figure #8 we see only white noise. This indicates that the new model was successful in capturing the total dynamics of the system. It can also be noted that the variance of the error has been reduced quite a bit too (see below).

$$\sigma_\varepsilon^2 = 0.0187 \quad \text{and,} \quad \text{var}\left(\hat{\beta}\right) = \begin{bmatrix} 0.86 & -0.81 & -0.038 & 0.18 \\ -0.81 & 0.84 & -0.015 & -0.066 \\ -0.038 & -0.015 & 3.24 & -3.09 \\ 0.0244 & -0.066 & -3.09 & 3.380 \end{bmatrix} \times 10^{-3}$$

We could use ESS to determine whether the improvement in fit is better or not.

ANSWER TO QUESTION NO. 2

(a)

The metal slab will be heated by a combination of convective and radiative heat transfer. The equation representing it will be:

$$q = UA\left(T_g - T\right) + \sigma\left(T_s^4 - T^4\right) \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (2.1)$$

where the temperatures measurements are given in terms of some absolute temperature scale.

(b)

As a first step of model identification, we plot all the given parameters in a single plot (see figure #9).

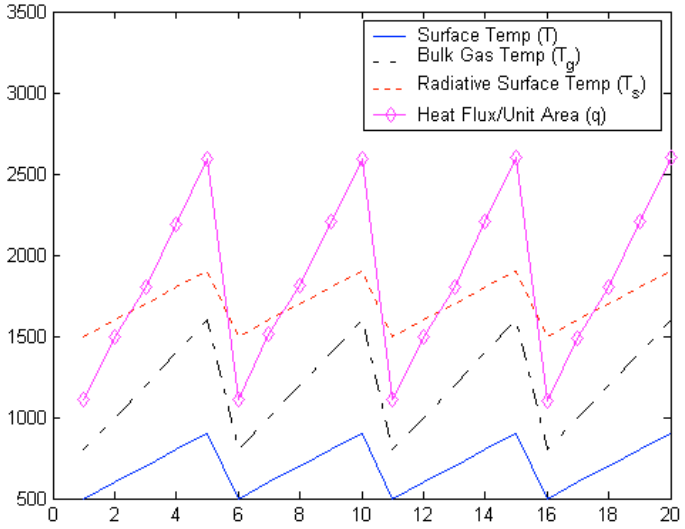

Fig # 9 : Plot of the given input-output parameters vs. time

The Matlab commands used to generate this plot are:

```
figure; plot (T, 'b-');
hold; plot (Tg, 'k-.')
plot (Ts, 'r:'); plot (q, 'm-d')
legend ('Surface Temp (T)', 'Bulk Gas Temp (T_g)',
'Radiative Surface Temp (T_s)','Heat Flux/Unit Area (q)')
```

We can write equation 2.1 in the following form:

$$q = \begin{bmatrix} \left(T_g - T\right) & \left(T_s^4 - T^4\right) \end{bmatrix} \begin{bmatrix} U \\ \sigma \end{bmatrix}$$

Now, that the equation has taken the form of $Y = X\beta$, we can use our parameter estimation formula, $\beta = \left(X^T X\right)^{-1}\left(X^T Y\right)$ to estimate U and σ.

The Matlab commands for this calculation like before are:

```
X = [(Tg-T)   (Ts.^4 - T.^4)*10^-10]
Y = q;
B=inv(X'*X)*(X'*Y)
```

Note that the second term has been multiplied by 10^{-10} to make the parameters of the same scale.

$$\beta = \begin{bmatrix} 3.2598 \\ 0.2504 \end{bmatrix}$$

The estimated U and σ are,

$$\hat{U} = 3.2598 \text{ and } \hat{\sigma} = 2.504 \times 10^{-9}$$

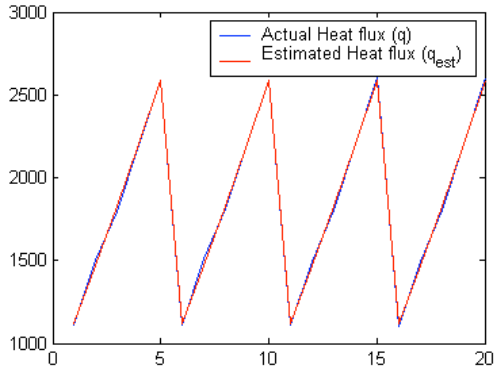

Fig # 10 : Plot of actual heat flux, q and estimated heat flux, $\hat{q}$ with time

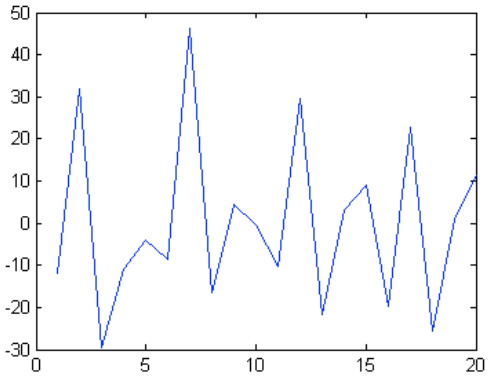

Fig # 11 : Plot of residual with time

(c)

Using the estimated parameter, we can get the estimated heat transfer rate, $\hat{q}$ from the following equation:

$$\hat{q} = \left[\left(T_g - T\right) \quad \left(T_s^4 - T^4\right)\right]\begin{bmatrix} \hat{U} \\ \hat{\sigma} \end{bmatrix}$$

Then, we calculate the covariance matrix of the parameters, $\text{var}\left(\hat{\beta}\right)$ by:

$$\text{var}\left(\hat{\beta}\right) = \left(X^T X\right)^{-1}\sigma_\varepsilon^2$$

where, σ_ε^2 = variance of the residual $= \text{var}\left(y - \hat{y}\right) = \dfrac{1}{N-p}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

In Matlab we use the following commands for it:

```
qh=X*B;
```

```
r= Y-qh;
SigmaSquare = (1/(20-2))*(r'*r)
VarB = (1/(20-2))*inv(X'*X)*(r'*r)
```

$\sigma_\varepsilon^2 = 473.1241$ and, $\text{var}\left(\hat{\beta}\right) = \begin{bmatrix} 0.0676 & -0.0399 \\ -0.0399 & 0.0236 \end{bmatrix}$

(d)

Based on the given data, the estimated values of the parameters will be:

$\hat{U} = 3.2598 \pm \sqrt{0.068} = 3.2598 \pm 0.26$ and

$\hat{\sigma} = \left(0.2504 \pm \sqrt{0.024}\right) \times 10^{-10} = \left(2.504 \pm 1.54\right) \times 10^{-9}$

These values of the parameter estimates are quite far off of the design values. The convective heat transfer coefficient is approximately 5 standard deviations from the design values; whereas, the radiative coefficient is certainly within one standard deviation of the design value. (Note that it is only approximately 1.4 standard deviations from zero, which raises the question of its statistical significance). Finally, the covariance terms for the parameter estimates cannot be ignored in this case, since they are large in comparison to the variance terms (and the actual parameter estimate for the radiative term).

ANSWER TO QUESTION NO. 3

(a)

Following the technique shown in Module 3, Question 6 on page 462 of "Process Dynamics Modeling, Analysis and Simulation", by B. Wayne Bequette we can rearrange the given equation $\left(r = \frac{kx_1x_2}{1+\alpha x_1^2}\right)$ in the following way:

$$\frac{1}{r} = \frac{1}{kx_1x_2} + \frac{\alpha x_1^2}{kx_1x_2} = \frac{1}{k}\frac{1}{x_1x_2} + \frac{\alpha}{k}\frac{x_1}{x_2}$$

or, $$\frac{1}{r} = \begin{bmatrix} \frac{1}{x_1x_2} & \frac{x_1}{x_2} \end{bmatrix} \begin{bmatrix} \frac{1}{k} \\ \frac{\alpha}{k} \end{bmatrix} \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad (3.1)$$

Before going into the regression, we plot the data to get a physical understanding of its nature.

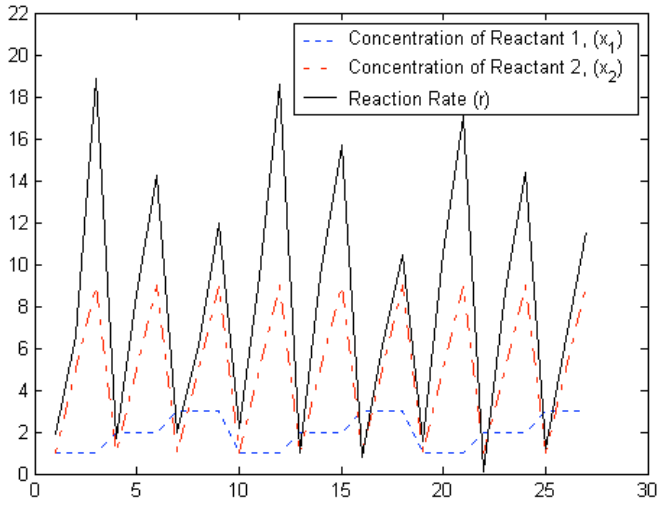

Fig # 12 : Plot of the given input-output parameters

Based on equation 3.1, create our X and Y matrix now:

$$X = \begin{bmatrix} \frac{1}{x_{10}x_{20}} & \frac{x_{10}}{x_{20}} \\ \frac{1}{x_{11}x_{21}} & \frac{x_{11}}{x_{21}} \\ \vdots & \vdots \\ \frac{1}{x_{1n}x_{2n}} & \frac{x_{1n}}{x_{2n}} \end{bmatrix} \text{ and, } \quad Y = \begin{bmatrix} \frac{1}{r_0} \\ \frac{1}{r_1} \\ \vdots \\ \frac{1}{r_n} \end{bmatrix}$$

We can generate the matrices with the following Matlab command and also estimate the parameters:

```
Y = 1./r;
U = [1./(X(:,1).*X(:,2)) X(:,1)./X(:,2)];
B1=inv(U'*U)*(U'*Y)
```

The estimated parameters will be:

$\beta = \begin{bmatrix} 0.7214 \\ 0.7949 \end{bmatrix}$ or $k = 1.3862$ and $\alpha = 1.1019$. Note how poor the estimate is for k.

To know the precision of our estimates, we will have to calculate the variance of the residual, $\sigma_\varepsilon^2 = \text{var}(y - \hat{y}) = \frac{1}{N-p}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, and also the covariance matrix of the parameters, $\text{var}(\hat{\beta}) = (X^T X)^{-1}\sigma_\varepsilon^2$.

We use the following Matlab commands to do this,

```
Yh=U*B;
resid= Y-Yh;
SigmaSquare = (1/(27-2))*(resid'*resid)
VarB = (1/(27-2))*inv(X'*X)*(resid'*resid)
```

And, the results are:

$$\sigma_\varepsilon^2 = 6.4987$$

$$\text{var}\left(\hat{\beta}\right) = \begin{bmatrix} 0.1292 & -0.0362 \\ -0.0362 & 0.0169 \end{bmatrix}$$

Looking at the covariance matrix for the parameters, it appears that parameter estimates are quite inaccurate (*i.e.*, compare the values in the matrix to the values of the parameter estimates). The accuracy of the reaction rate predictions from this model is also very poor. (Note the size of the prediction error variance compared to the size of $1/r$).

We plot the actual rate with the estimated rate in Figure #13 and the residual of the model in Figure #14. The explanation of the plots is given in (c).

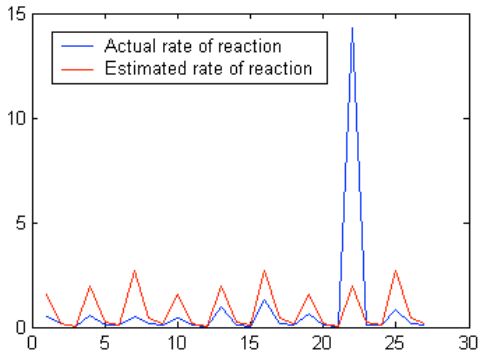

Fig # 13 : Plot of actual rate of reaction (r) and estimated rate of reaction ($\hat{r}$) with time

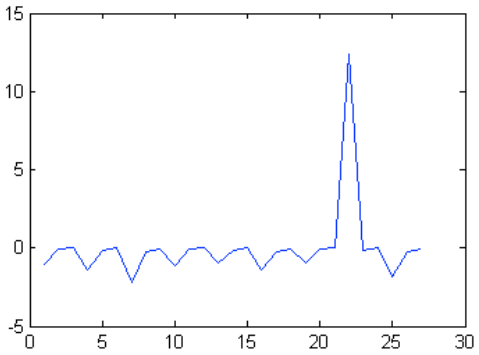

Fig # 14 : Plot of residual with time

(b)

In this section we have a nonlinear equation for the given model. We will have to use "lsqcurvefit" command from Matlab to get the parameter estimates and then we will use that to estimate reaction rate ($\hat{r}$) and the residual of the reaction rate (*resid*)

The Matlab command used to for this is:

```
B2 = lsqcurvefit('kinetics',[ 0 0],X,r)
rh=B2(1).*X(:,1).*X(:,2)./(1+B2(2)*X(:,1).^2);
resid2 = r-rh;
```

The estimated parameters are:

$$\beta = \begin{bmatrix} 3.6005 \\ 0.8363 \end{bmatrix}$$

The variance of the residual and the covariance matrix of the parameters were found to be as follows:

$$\sigma_\varepsilon^2 = 0.8289$$

This means that the standard deviation on our reaction rate predictions is ± 0.9104 ($\sqrt{\sigma_\varepsilon} = \sqrt{0.8289} = 0.9104$). Given that the magnitude of the reaction rates, this seems reasonably accurate.

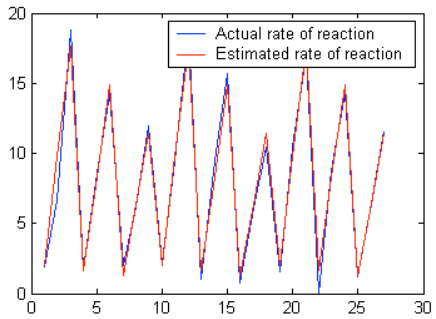

Fig # 15 : Plot of actual rate of reaction (r) and estimated rate of reaction ($\hat{r}$) from nonlinear regression with time

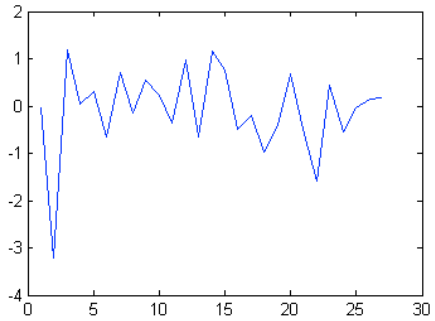

Fig # 16 : Plot of residual from nonlinear regression with time

(c)

In the residuals of the linear model (see Figure #14) we notice a very large spike for sample number 22. The reason for this is that the reaction rate r for this sample is very small compared to the other reaction rates in the data and inverting a small number results in a very large number. The transformed model must then take this large spike into account, which "biases" the parameter estimates. On the other hand, the nonlinear model is showing very good fit. This is evident from the almost perfect white noise residual (see Figure #16).

The key things to note in the results are that: 1) the parameter estimates from the nonlinear least squares procedure are quite close to the real values (*i.e*, there is very little bias in the estimates); 2) the prediction error variance from the nonlinear least squares estimates is quite reasonable in comparison to the experimental values of the reaction rates; 3) the parameter estimates that resulted from transforming the model and using linear least squares were quite biased and particularly so for k; 4) the parameter

covariance matrix for the transformed model showed how poor the accuracy was for the parameter estimates; and 4) the prediction error variance for the transformed model was quite large in comparison to the values of $1/r$. In short, the transformation and linear least squares approach failed miserably for this data.

(d)

To explain why the experiments were done in the way they were done, we need to plot the input variables one against the other (see Figure #17).

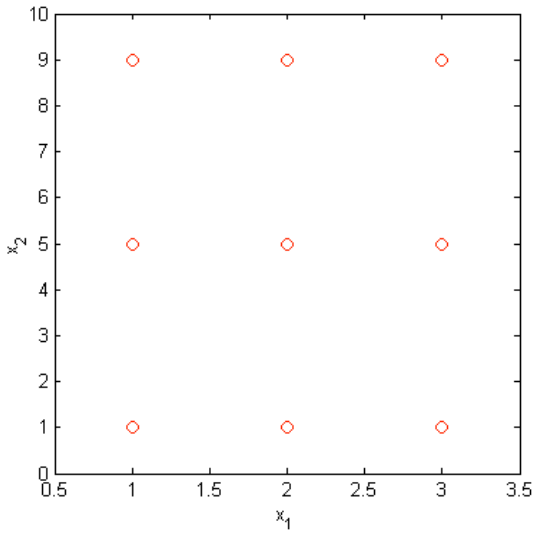

Fig # 17 : Plot of x_1 verses x_2

From the plot we can easily see that the experiments were done in a grid like fashion, with three different values chosen for each independent variable. In fact this is a full, three level experimental design, with repeated experiments. Three or higher level experimental designs like this are required to capture nonlinearity in the models. The more conventional two-level designs are best suited to fitting linear relationships.

APPENDIX A

The Matlab code used in this assignment are as follows:

Question # 1

```
clear all; close all
load che562_assn5q1_2002
f1=f-mean(f);
T1=T-mean(T);
figure; plot (t,T-mean(T)); hold; plot (t,f-mean(f), 'r-.')
legend ('Temperature (T)', 'Flow rate (f)')
figure; cra ([T1 f1]);

X=[ T1(1:end-1) [0; T1(1:end-2)] [ 0; 0; 0;  f1(1:end-4)] ];
Y=T1(2:76);
B=inv(X'*X)*(X'*Y)

Th=X*B;
r= T1-Th;

figure; plot (T)
hold; plot (Th+mean(T), 'r-.')
legend ('Temperature Actual (T)', 'Temperature Estimated (Th)')
figure; plot (r)

SigmaSquare = (1/(75-3))*(r'*r)
VarB = (1/(75-3))*inv(X'*X)*(r'*r)

X2=[ T1(1:end-1) [ 0; T1(1:end-2)] [ 0; 0; 0; f1(1:end-4)] [ 0; 0; 0; 0; f1(1:end-5)] ];
B2=inv(X2'*X2)*(X2'*Y)

Th2=X2*B2;
r2= T1-Th2;

figure; plot (T)
hold; plot (Th2+mean(T), 'r')
figure; plot (r2)

SigmaSquare2 = (1/(75-4))*(r2'*r2)
VarB = (1/(75-4))*inv(X2'*X2)*(r2'*r2)
```

Question # 2

```
clear all; close all
load che562_assn5q2_2002

figure; plot (T, 'b-');
hold; plot (Tg, 'k-.')
plot (Ts, 'r:'); plot (q, 'm-d')
legend ('Surface Temp (T)', 'Bulk Gas Temp (T_g)', 'Radiative Surface Temp
(T_s)','Heat Flux/Unit Area (q)')

X = [(Tg-T) (Ts.^4 - T.^4)*10^-10
Y = q;
B=inv(X'*X)*(X'*Y)

qh=X*B;
r= Y-qh;
SigmaSquare = (1/(20-2))*(r'*r)
VarB = (1/(20-2))*inv(X'*X)*(r'*r)

figure; plot (q)
hold; plot (qh, 'r')
legend ('Actual Heat flux (q)', 'Estimated Heat flux (q_e_s_t)')
figure; plot (r)
```

Question # 3

```
clear all; close all
load che562_assn5q3_2002

figure; plot (X(:,1), 'b:')
hold; plot (X(:,2), 'r-.'); plot (r, 'k-')
legend ('Concentration of Reactant 1, (x_1)', 'Concentration of Reactant 2, (x_2)',
'Reaction Rate (r)')

Y = 1./r;
U = [1./(X(:,1).*X(:,2))  X(:,1)./X(:,2)];
Y1=Y;
B1=inv(U'*U)*(U'*Y1)

Yh=U*B1;
resid= Y1-Yh;
SigmaSquare = (1/(27-2))* (resid'*resid)
VarB = (1/(27-2))*inv(X'*X)*(resid'*resid)

figure; plot (Y)
hold; plot (Yh, 'r')
legend ('Actual rate of reaction', 'Estimated rate of reaction')
figure; plot (resid)

B2=LSQCURVEFIT('kinetics',[ 0 0],X,r)
rh=B2(1).*X(:,1).*X(:,2)./(1+B2(2)*X(:,1).^2);
resid2 = r-rh;
SigmaSquare2 = (1/(27-2))* (resid2'*resid2)

figure; plot (r)
hold; plot (rh, 'r-')
legend ('Actual rate of reaction', 'Estimated rate of reaction')
figure; plot (resid2)

figure; plot (X(:,1), X(:,2), 'ro')
xlabel ('x_1'); ylabel ('x_2');
```

--- X ---