SC 626

SYSTEMS AND CONTROL ENGINEERING LABORATORY

MULTI INPUT MULTI OUTPUT CONTROLLERS

Amey Samrat Waghmare
203230013
Systems and Control Engineering,
Indian Institute of Engineering, Bombay

April 7, 2021

AIM

To design and implement Multi Input Multi Output controllers, specifically

1. MIMO PI Controller
2. Decentralized PI Controller

For our system, T1-T3 will be changed and T2-T4 will be kept constant.

CHOICE OF INPUT OUTPUT PAIRING

Relative Gain Array

For any MIMO systems, interactions between output and all the inputs are present. Relative Gain array provides a quantitative assessment of the effect of changing a manipulated variable on different controlled variables, so that we can choose those input and output pairs which minimizes the interactions among all the loops. Relative Gain Array is given by,

$$\Lambda = K \odot (K^{-1})^T$$

where K is steady state Gain matrix and $\odot$ represents the element-wise product.
The elements of RGA indicates how the input output pairing should be selected based upon the following rules,

- $\lambda_{ij} = 1$, indicates that open loop gain and closed loop gain between y_i and u_j is identical. Hence, the Loop Interactions will not affect the the pairing y_i -u_j and is the Most recommended pairing.
- $\lambda_{ij} = 0$, indicates that the open loop gain between y_i and u_j is zero. This means that u_j has no direct effect on y_i and so this pairing should not be selected.
- $0 < \lambda_{ij} < 1$, indicates that open loop gain between y_i and u_j is smaller than the closed loop gain. Hence, there is interaction between loops and is most severe when $\lambda_{ij} = 0.5$ and such pairing is not recommended as it introduces interactions.
- $\lambda_{ij} > 1$, indicates that open loop gain between y_i and u_j is larger than the closed loop gain. Here, the interaction between the loops increases as the value of λ_{ij} increases. Hence this pairing is also not recommended.

- $\lambda_{ij} < 0$, indicates that the closed loop gain is in opposite direction from open loop gain. Hence it is not recommended to pair y_i -u_j as it introduces most severe interactions.

Condition Number

Condition number (CN) is the ratio of the largest to the smallest singular value and indicates if K, steady state gain matrix is ill-conditioned. It is a measure of sensitivity of the matrix properties to changes in a specific element. It is calculated as the ratio of largest to smallest singular value, obtained after singular value decomposition. Large value of CN indicates an Ill conditioned K whereas small value indicates Well conditioned system.

Niederlinski Index

The Niederlinski Index (NI) is a calculation used to analyze the stability of the control loop pairings using the result of the RGA, evaluated at Steady State as follows,

$$NI = \frac{|K|}{\prod_{i=1}^{n} k_{ii}}$$

A negative NI value indicates instability in the control loop. Hence the pairing analysis by RGA along with positive NI value can ensure stability.

0.0.1 RGA, CN and NI analysis on SBMH system

As instructed, the FOPTD model developed in the Week 1, part 2 of this lab is used for analysis and design of the controllers. Here although we have 5 inputs and 5 outputs, one of the inputs is the disturbance fan and it is not modelled in ARMAX model and hence needs to be neglected in FOPTD Model. So we have Steady state Gain Matrix, K of dimension 5x4 as shown below.

$$K = \begin{bmatrix} 1.0040 & 0.5139 & 0.3266 & 0.2993 \\ 0.3081 & 1.0627 & 0.3651 & 0.1780 \\ 0.2796 & 0.3982 & 0.9589 & 0.2551 \\ 0.5095 & 0.3697 & 0.3119 & 0.9000 \\ 0.4319 & 0.5202 & 0.4403 & 0.3135 \end{bmatrix}$$

The Corresponding RGA for this Non Square system is,

$$RGA = \begin{bmatrix} 1.3396 & -0.1379 & -0.0509 & -0.1863 \\ -0.1519 & 1.2579 & -0.1347 & -0.0207 \\ -0.0535 & -0.1557 & 1.2116 & -0.0526 \\ -0.1589 & -0.0285 & -0.0881 & 1.2531 \\ 0.0246 & 0.0642 & 0.0621 & 0.0065 \end{bmatrix}$$

This RGA suggests Diagonal pairing i.e. $u_1 - y_1$, $u_2 - y_2$ and so on.Lets ignore any one output as we have only 4 inputs and hence we can control only 4 outputs.
Consider only y_1, y_2, y_3, y_4 are present. The corresponding RGA, CN and NI are as follows,

$$RGA = \begin{bmatrix} 1.3514 & -0.1249 & -0.0415 & -0.1850 \\ -0.1476 & 1.2896 & -0.1222 & -0.0198 \\ -0.0496 & -0.1437 & 1.2446 & -0.0513 \\ -0.1541 & -0.0210 & -0.0809 & 1.2561 \end{bmatrix}$$

$$CN = 3.9332$$

$$NI = 0.561$$

We can see that RGA suggests Diagonal Pairing, Condition Number is small indicating a well conditioned system and NI is positive, indicating stability of the control loop.

Let us consider only y_1, y_3, y_4, y_5 are present. The corresponding RGA, CN and NI are as follows,

$$RGA = \begin{bmatrix} 1.7592 & -0.6537 & 0.0512 & -0.1567 \\ 0.0854 & -0.6309 & 1.5682 & -0.0227 \\ 0.0102 & -0.3231 & -0.0107 & 1.3236 \\ -0.8548 & 2.6077 & -0.6087 & -0.1441 \end{bmatrix}$$

$$CN = 12.3278$$

$$NI = 3.19$$

We can see that RGA suggests the pairing,

$$u_1 - y_1, u_2 - y_5, u_3 - y_3, u_4 - y_4$$

Condition Number is a little large (above 10) and NI is positive, indicating stability of the control loop.
As it is asked to control only y_1 and y_3 and keep others constant, from above analysis, Diagonal pairing is best as it also ensures a stable and well conditioned system.

MIMO PI Controller

A MIMO PI controller needs to be designed to control the system. As instructed, FOPTD model will be used to design the PI Controller and it will be implemented on ARMAX model developed in Week 2, part 2 of this lab. As we have to control only y_1 and y_3, we need to design PI Controllers only for the following Transfer Functions obtained from FOPTD model Transfer Function Matrix from week 1.

$$g_{11} = \frac{1.004}{1+44.4867s}e^{-3.0899s}$$

$$g_{33} = \frac{0.9589}{1+25.2969s}e^{-1.0362s}$$

Pole Placement method for FOPTD model is used to design PI controllers of the form,

$$C(s) = k\left(1+\frac{1}{T_i s}\right)$$

Controllers are designed for all possible pairings. For FOPTD system, combining this controller yields the following characteristics equation (K_p and T are parameters of FOPTD model),

$$s(T-kk_p)+kk_p = 0$$

, and suppose that the desired close loop pole is desired to be placed at T_{cl}, then by comparison, we get the following tuning rule,

$$k = \frac{1}{k_p}\frac{T}{L+T_{cl}}$$

$$T_i = T$$

Where $T_{cl} = 3T$ for robust controller and $T_{cl} = T$ for aggressive tuning. Suppose we desire for robust tuning, so we get the following PI values for the controllers,

1.
   - $k_{11} = 0.9318$
   - $T_{i_{11}} = 44.867$
2.
   - $k_{22} = 0.941$
   - $T_{i_{22}} = 29.129$

1.
   - $k_{33} = 1.0018$
   - $T_{i_{33}} = 25.2969$
2.
   - $k_{44} = 1.032$
   - $T_{i_{44}} = 26.1158$

The above PI controller is in Continous Time, to implement it in MATLAB, it needs to be discretized. This is explained below,

$$u(t) = k\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau\right)$$

Let,$\eta(t) = \int_0^t e(\tau)d\tau$, which is same as $\frac{d\eta}{dt} = e(t)$. Hence we can write,

$$\frac{\eta(k+1) - \eta(k)}{T} = e(k)$$

$$\eta(k+1) = \eta(k) + Te(k)$$

Hence, the input becomes,

$$u(k) = ke(k) + \frac{k}{T_i}\eta(k)$$

These equations are implemented to discretize the PI Controller, the T in the above equations is the sampling time which is 3.0936s for SBMH system.
This MIMO PI Controller was implemented, the unit step change was applied at 92.8 sec to u_1 input and at 201.08 sec to u_3 input, rest were kept at steady state. The obtained results is shown in the Figure(1)

The control efforts generated by PI Controller is shown in Figure (4)

Here, we observe that the PI controllers are able to regulate the system to appropriate values. However, we also see that the other outputs are also varying. This is due to the interloop interactions. By selecting the pairing using RGA we have only minimized the Interactions, but not eliminated. To eliminate them, we need to design a decoupler, whis is done in the next section.

0.1 Decentralized PI Controller

As we have seen in the previous section, if we change a particular input, keeping other inputs constant, all the outputs are affected. This is due to loop interactions amongst them. To

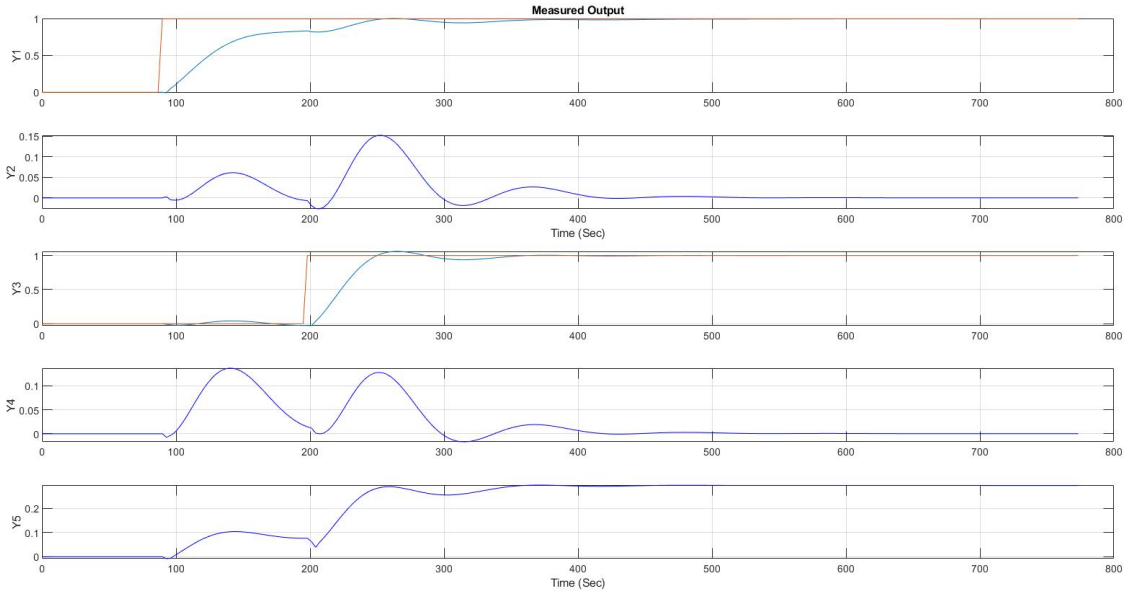

Figure 1: MIMO PI Controller

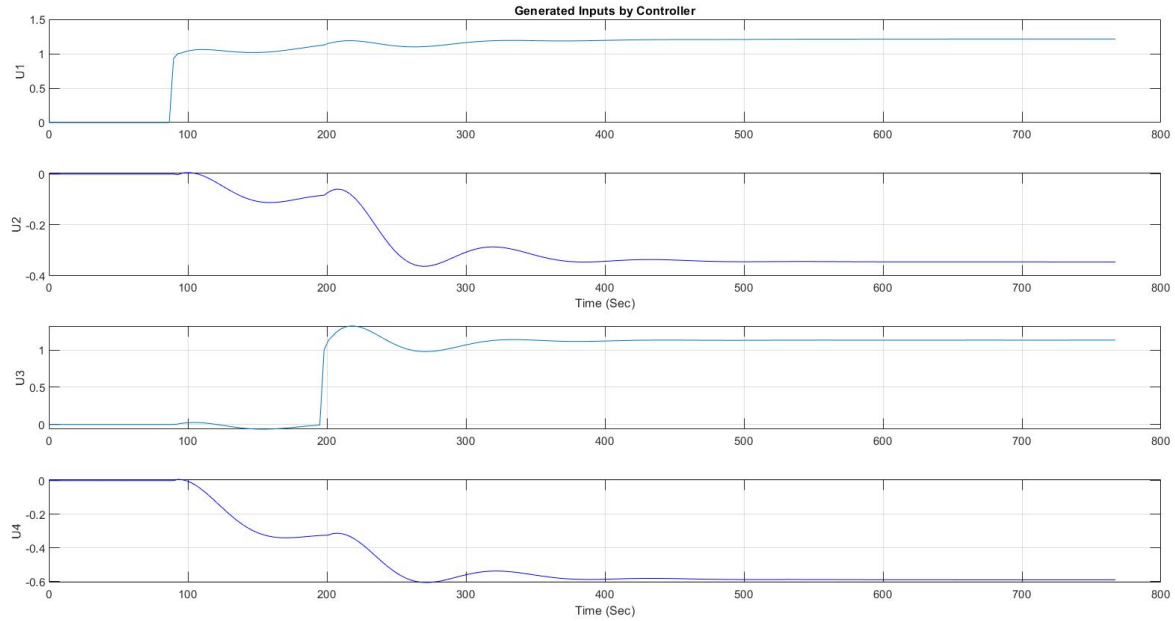

Figure 2: Control inputs generated by PI Control

reduce these interactions, we need to decouple the system. By decoupling, the effect of change in input only affects a particular output and not other outputs. This is desired, and

then we can design a PI controller for this decoupled system.
There are two methods to decouple the system; Ideal decoupling and Simplified Decoupling. A simplified decoupler is used for decoupling the system. as we are asked to change only u_1 and u_3, we have only decoupled this Two input and Two output system. The decoupler takes the following form

$$D(s) = \begin{bmatrix} 1 & d_{13}(s) \\ d_{31}(s) & 1 \end{bmatrix}$$

The TITO system is

$$G_p(s) = \begin{bmatrix} \frac{1}{1+44.48s}e^{-3.1s} & \frac{0.33}{1+61.02s} \\ \frac{0.28}{1+80.93s}e^{-5.2s} & \frac{0.96}{1+25.3s}e^{-1.03s} \end{bmatrix}$$

The resultant system will be,

$$G_p(s)D(s) = \begin{bmatrix} g_{11}^*(s) & 0 \\ 0 & g_{33}^*(s) \end{bmatrix}$$

Which were obtained to be,

$$g_{11}^*(s) = \frac{1}{1+44.48s}e^{-4.13s} - \frac{48.125(10+253s)}{(50+3051s)(100+8093s)}e^{-5.2s}$$

$$g_{33}^*(s) = \frac{9.6}{10+253s}e^{-4.13s} - \frac{18.48(25+1112s)}{(50+3051s)(100+8093s)}e^{-5.2s}$$

It is very difficult tpo design PI parameters for above systems, hence they are first reduced to FOPTD model using model reduction by step input and then the PI Controllers were designed for this reduced model. The FOPTD models obtained are,

$$g_{11}^*(s) = \frac{0.951}{1+44.5s}e^{-4.2s}$$

$$g_{33}^*(s) = \frac{0.96}{1+73s}e^{-4.2s}$$

Correspondingly, the decoupler obtained is,

$$D(s) = \begin{bmatrix} e^{-1.03s} & \frac{-29.17-737.9s}{100+8093s}e^{-5.2s} \\ \frac{-16.5-733.9s}{50+3051s} & e^{-3.1s} \end{bmatrix}$$

The diagonal terms are delay instead of 1 to take into account the time advance terms resulted due to combining the decoupler with the plant. This decoupler was converted into state

space model and was discretized in order to implement it with our system. Standard MATLAB commands like c2d were used to achieve this.

The PI parameters for FOPTD model were obtained by Lambda Tuning rule which is as follows,

$$k = \frac{1}{k_p}\frac{T}{L+T_{cl}}$$
$$T_i = T$$

where, $T_{cl} = 3T$ for robust controller and $T_{cl} = T$ for aggressive tuning. We have taken aggressive tuning as our design parameter and hence the obtained PI parameters are,

- $k_{11} = 0.961$
- $T_{i_{11}} = 44.5$
- $k_{33} = 0.985$
- $T_{i_{33}} = 73$

The ARMAX model was simulated with above parameters and the response obtained is shown in Figure(3)

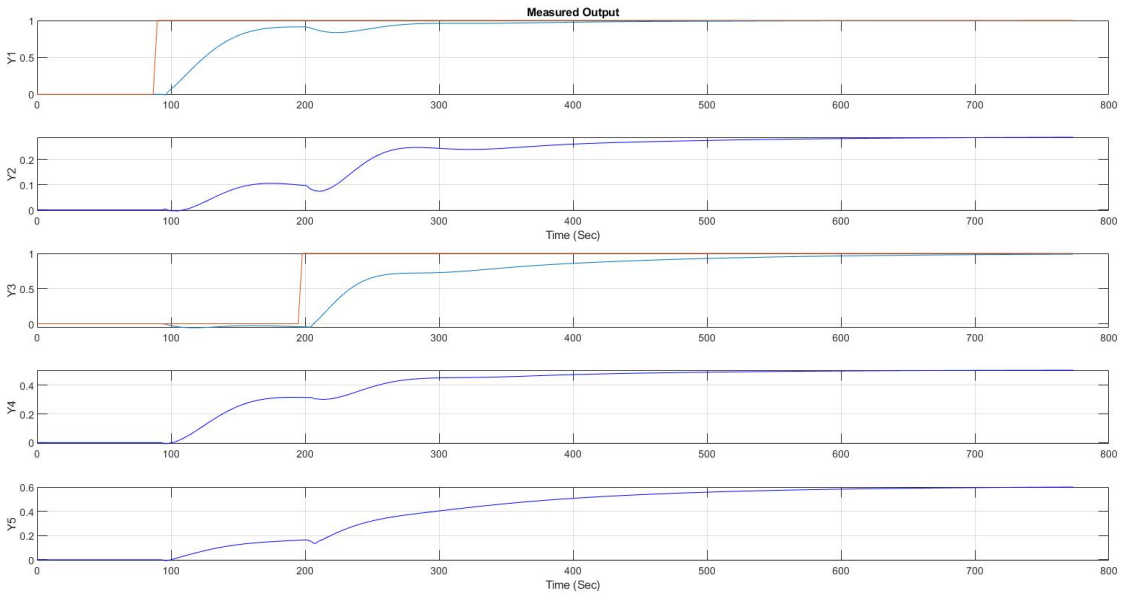

Figure 3: Decentralized MIMO Control

The control efforts generated by Decentralized PI Controller is shown in Figure (4)

The comparison between MIMO PI Controller and Decentralized MIMO PI Controller is shown below, where we have used RMSE as a performance index, for all the outputs,

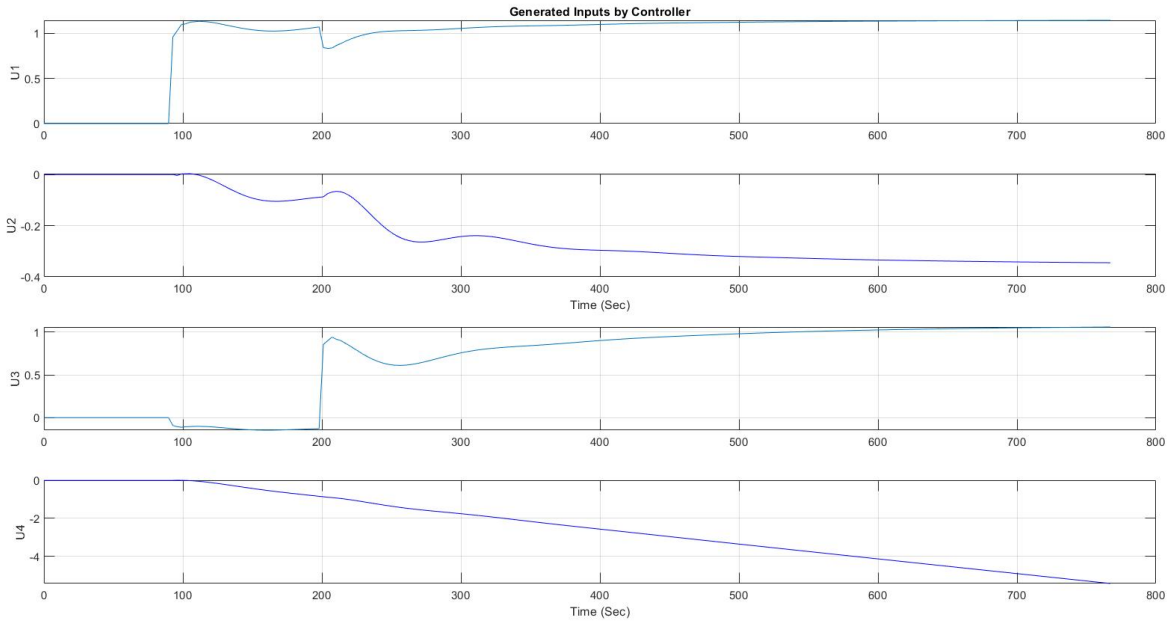

Figure 4: Control inputs generated by Decentralized PI Control

RMSE		
-	MIMO PI (^oC)	Decentralized MIMO PI (^oC)
y_1	0.1961	0.2056
y_2	0.0362	0.0259
y_3	0.1645	0.2445
y_4	0.0433	0.1879
y_5	0.2440	0.2693

CONCLUSION

1. The MIMO PI Controller and Decentralized PI Controller were implemented for the given system.

2. TO decide the Pairing, FOPTD model was used and based on the RGA Analysis, Condition Number and Niederlinski Index, the Diagonal Pairing, i.e $u_1 - y_1$, $u_2 - y_2$,... was most suitable pairing.

3. For MIMO PI Controller, the PI Controller were directly designed for the corresponding FOPTD systems and were tuned for certain design parameters.

4. It was observed through simulations that the designed MIMO Pi controller works well, except that due to interactions present, the other outputs are also affected.

5. To avoid the problem of interactions, a decoupler was designed. Simplified Decoupling was implemented on Two Input Two Output system. Lambda Tuning was used to design PI controller.

6. After simulations, it was observed that there were negligible interactions present in the system, indicating that the decoupler was doing its task of minimizing interactions as compared to MIMO PI Controller.

SC 626

SYSTEMS AND CONTROL ENGINEERING LABORATORY

MULTI INPUT MULTI OUTPUT CONTROLLERS

Amey Samrat Waghmare
203230013
Systems and Control Engineering,
Indian Institute of Engineering, Bombay

April 7, 2021

AIM

To develope ARMAX model using Recursive Least Square Estimation

LEAST SQUARE MODEL

Least Square Estimation is a basic method for estimating the parameters of a model. Basically Karl Friedrich Gauss formulated the principle of least squares estimation and he stated that the unknown parameters of a mathemetical model should be chosen in such a way that the sum of the squares of the differences between the actually observed and the computed values, multiplied by numbers that measure the degree of precision, is minimum. The least squares method can be applied to a large variety of problems. For a mathematical model shown below, we can formulate the Least Squares Estimation problem as,

$$y(i) = \phi_1(i)\theta_1 + \phi_2(i)\theta_2 + ... + \phi_n(i)\theta_n$$

$$y(i) = \phi^T(i)\theta \tag{1}$$

The above model is a linear one and the parameters of this model are θ,

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ . \\ . \\ \theta_n \end{bmatrix}$$

Here, y is the observed variable, i is the observation number and ϕ_1,ϕ_2,...ϕ_n are the known functions that may depend on the other variables. The variables ϕ are also called regression variables or the regressors and the Equation (1) is called regression model.

The problem is to determine the parameters in such a way that outputs computed from the regression model Equation (1) should be as close as possible with the measured variable y in the sense of the least squares. This means that the parameter θ should be chosen to minimize the following least squares loss function,

$$V(\theta, t) = \frac{1}{2}\sum_{i=1}^{t}\left(y(i) - \phi^T\theta\right)^2 \tag{2}$$

To solve above problem, let us use the following notations,

$$\mathscr{Y} = \begin{bmatrix} y(1) & y(2) & . & . & y(t) \end{bmatrix}^T$$

$$\mathscr{E} = \begin{bmatrix} \epsilon(1) & \epsilon(2) & . & . & \epsilon(t) \end{bmatrix}^T$$

$$\Phi = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ . \\ . \\ \phi^T(t) \end{bmatrix}$$

$$P(t) = \left(\Phi^T(t)\Phi(t)\right)^{-1}$$

where, $\mathscr{E}$ are the residuals, defined by

$$\epsilon(i) = y(i) - \phi^T(t)\theta$$

By using the above notations, we can rewrite the loss function as,

$$V(\theta, t) = \frac{1}{2}||\mathscr{E}||^2$$

Where, $\mathscr{E}$ can be written in vectorized notation as,

$$\mathscr{E} = \mathscr{Y} - \Phi\theta$$

LEAST SQUARES ESTIMATION

The loss function in Equation (2) have a minimum and is given by,

$$\hat{\theta} = \left(\Phi^T(t)\Phi(t)\right)^{-1}\Phi^T(t)\mathscr{Y} \tag{3}$$

where, the solution is unique if $\Phi^T\Phi$ is non-singular.

The above method to determine the parameters of the model is simple, however faces problems. If the dataset is very large, it is not possible to invert the very large resulting matrix

$\Phi^T\Phi$. This is computationally challenging. To overcome this difficulty, an extension to the above method is developed which is called Recursive Least Squares Estimation (RLSE).

RECURSIVE LEAST SQUARES ESTIMATION

Let us assume that the matrix $\Phi(t)$ is a full rank matrix, indirectly it means that $\Phi^T\Phi$ is non-singular for all $t \geq t_0$. Given $\theta(\bar{t_0})$ and $P(t_0) = (\Phi^T(t_0)\Phi(t_0))^{-1}$, the least squares estimate $\theta(\hat{t})$ then satisfies the recursive equations,

$$\theta(\hat{t}) = \theta(\hat{t-1}) + K(t)(y(t) - \phi^T(t)\theta(\hat{t-1})) \tag{4}$$

$$K(t) = P(t)\phi(t) - P(t-1)\phi(t)(\lambda I + \phi^T P(t-1)\phi(t))^{-1} \tag{5}$$

$$P(t) = (I - K(t)\phi^T(t))P(t-1)/\lambda \tag{6}$$

Where,

$$\theta^T = \begin{bmatrix} a_1 & \dots & a_n & b_1 & \dots & b_n & c_1 & \dots & c_n \end{bmatrix}$$

$$\Phi^T = \begin{bmatrix} -y(k-1) & \dots & -y(k-n) & u(k-1) & \dots & u(k-n) & e(k-1) & \dots & e(k-1) \end{bmatrix}$$

Here, the order n is to be given by the user. It determines the complexity of the model. For example, having large n is good as many parameters are used to represent our model but at the same time due to many parameters, the model becomes more complex.
For the simulation purpose, we have taken $n = 2$, in order to reduce the computational load. Hence, Φ will contain 14 columns. A ARMAX model (MISO) obtained corresponding to output y_5 and converted into state space realization is shown below,

$$\phi = \begin{bmatrix} 0.2129 & 1 \\ 0.7413 & 0 \end{bmatrix}$$

$$\gamma_u = \begin{bmatrix} 0.0274 & 0.0436 & 0.0306 & 0.0363 & -0.013 \\ -0.0116 & -0.0169 & -0.009 & -0.0188 & 0.0037 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 0.2129 \\ 0.7413 \end{bmatrix}$$

Similarly, MISO model for y_2 is obtained and indicated below,

$$\phi = \begin{bmatrix} -0.2804 & 1 \\ 1.2203 & 0 \end{bmatrix}$$

$$\gamma_u = \begin{bmatrix} 0.0096 & 0.0536 & 0.0245 & 0.0068 & -0.0057 \\ -0.0014 & 0.0104 & -0.0126 & -0.0042 & -0.0056 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 0.2804 \\ -1.2203 \end{bmatrix}$$

Figure (1) shows how the parameters of the ARMAX model converge to optimum value. These plots are corresponding to output y_5

The obtained model was compared with that obtained during Week 2 of the lab and the results are not exactly matching. This is because the size of polynomial was set to 4 in Week 2 experiment, whereas here the polynomial size is set to 2. So less parameters to represent the model, but it does a fairly good job. This comparison is shown in Figure (reffig:week4comparison) where the blue line indicates the response of MISO ARMAX model of week 2 and the red line indicates the MISO ARMAX model of this week. The graphs are plotted corresponding to output y_2.

CONCLUSION

1. The ARMAX model was developed from first principles, however the MATLAB's System Identification Toolbox uses different algorithms to identify ARMAX model such as use of Kalman Filter, Extended Kalman Filter, etc. Due to this the model developed here will not be same as obtained by System Identification Toolbox.

2. All the parameters converge to a value as seen in the waveforms. This indicates that with time/ iterations, the model is developing and converging to an optimal value.

3. These parameters are the parameters of the ARMAX model and they have been converted into equivalent state space MISO models.

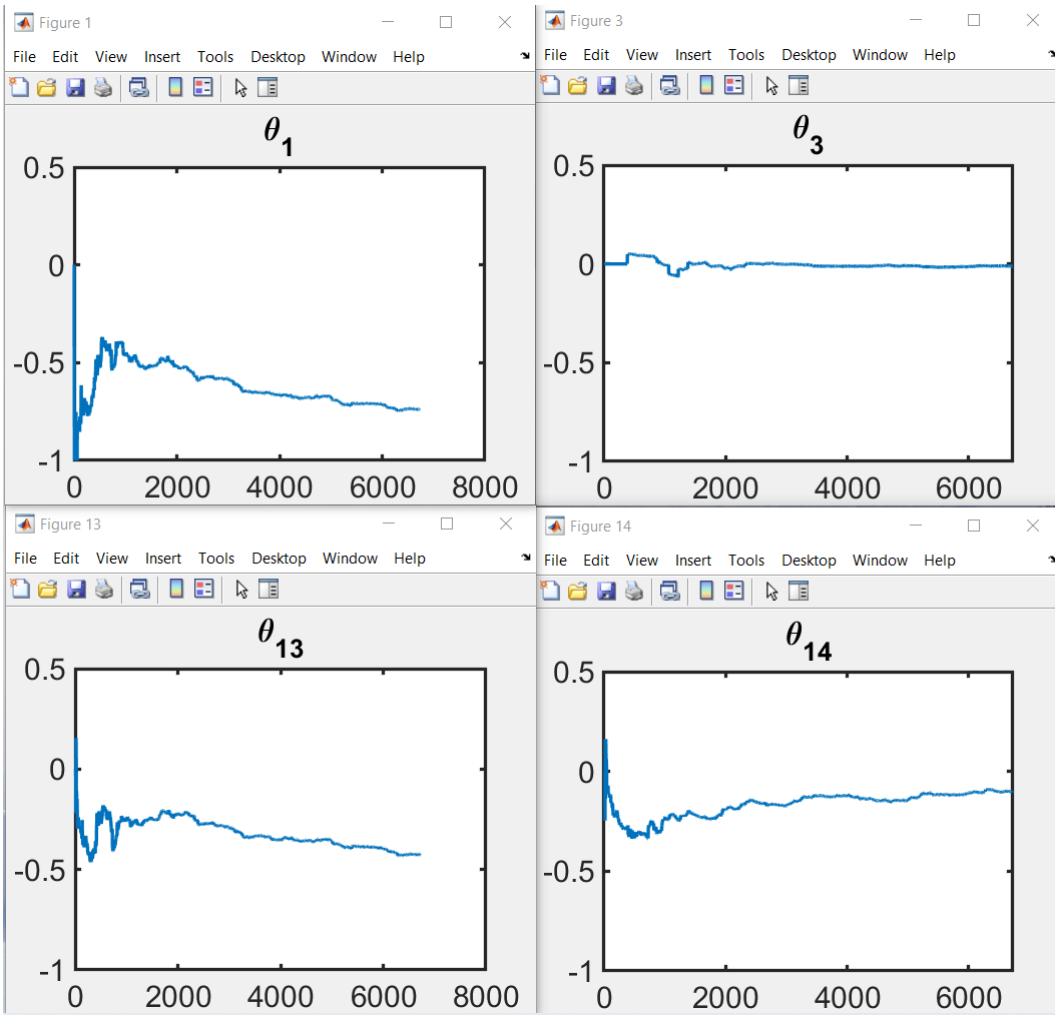

Figure 1: Parameters of ARMAX model

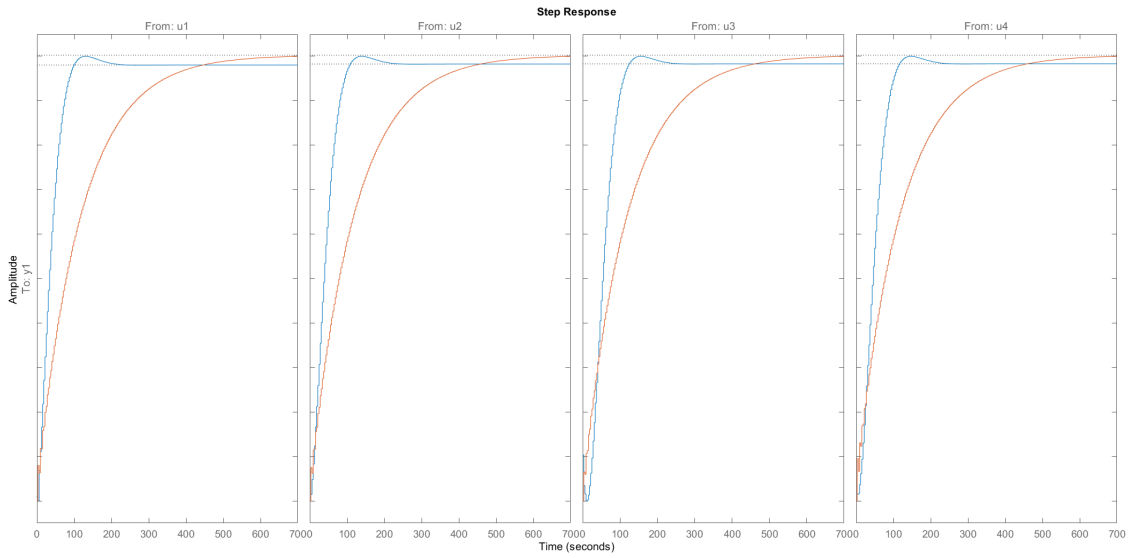

Figure 2: Comparison of MISO ARMAX Models for output y_2