SC 626

SYSTEMS AND CONTROL ENGINEERING LABORATORY

LINEAR QUADRATIC OPTIMAL CONTROL

Amey Samrat Waghmare

203230013

Systems and Control Engineering,

Indian Institute of Engineering, Bombay

April 20, 2021

Aim

To design a Linear Quadratic Optimal Controller for SBMH system.

Linear Quadratic Optimal Controller

In Linear Quadratic Optimal Controller, our objective is to bring the system from non zero initial state to zero initial state, i.e. the origin of state space. In other words, we want to design a state feedback law of the type

$$u(k) = -Gx(k)$$

which will bring the system to origin in an optimal way.
Suppose that we have the plant dynamics as,

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$

$$y(k) = Cx(k)$$

We need to formulate an optimization problem which can find an optimal sequence of inputs $u(0)$, $u(1)$, ...$u(N-1)$ that takes the system to origin in an optimal manner. Hence, we need to formulate an optimization cost function to achieve this.
For the Objective function, we can include a term that penalizes the deviation from origin. It can be formulated as the distance of states from origin, as shown below.

$$||x(k)||_2^2 = x(k)^T x(k)$$

The above term penalizes all the states equally. However, sometimes it is desirable to penalize some states more aggressively than others. This can be done by introducing a positive definite weighing matrix W_x as follows,

$$||x(k)||_2^2 = x(k)^T W_x x(k)$$

Many a times, it is also desirable that a controller does not result in an excessively large control moves. To ensure that small control moves are generated, similar to above weighting, we can also penalize the excess control inputs as follows,

$$||u(k)||_2^2 = u(k)^T W_u u(k)$$

Where W_u is a positive definite matrix.
By penalizing the states and control inputs, we can formulate an optimization cost function J as follows,

$$J = \sum_{k=0}^{N-1} \left[x(k)^T W_x x(k) + u(k)^T W_u u(k) \right] + x(N)^T W_N x(N)$$

Where N is terminal instant. The problem of designing an optimal controller is posed as minimization of this objective function with respect to decision variable $u(0)$, $u(1)$, ...$u(N-1)$. The solution of this optimization problem is solved using the Dynamic Programming method by Bellman. We solve the problem at time instant k asumming that the problem has been solved optimally upto time instant $k-1$. Hence, we first minimize $J(N)$, at terminal state, obtain the optimal solution $u(N-1)$ and corresponding $G(N-1)$, which are given as,

$$u(N-1) = -G(N-1)x(N-1)$$

$$G(N-1) = (W_u + \Gamma^T W_N \Gamma)^{-1} \Gamma^T W_N \Phi$$

which will give the minimum cost as $J(N-1) = x(N-1)^T W_N x(N-1)$. In a similar way, we solve the optimization problem at instant $N-2$, $N-3$,.. and so on in backward fashion. Here, we take $S(N) = W_N$. At k^{th} instant, we get

$$G(k) = (W_u + \Gamma^T S(k+1)\Gamma)^{-1} \Gamma^T S(k+1)\Phi$$

and

$$S(k) = \left[\Phi - \Gamma G(k)\right]^T S(k+1)\left[\Phi - \Gamma G(k)\right] + W_x + G(k)^T W_u G(k)$$

The above equation is called discrete time Riccati equation. As W_N and W_u are symmetric positive definite, $S(k)$ is also positive definite. When N (horizon) becomes very large, $S(k)$ tends to a constant matrix S_∞, which can be calculated by solving the Algebric Riccati Equation

$$G_\infty = (W_u + \Gamma^T S_{infty} \Gamma)^{-1} \Gamma^T S_\infty \Phi$$

$$S_{infty} = \left[\Phi - \Gamma G_{infty}\right]^T S_{infty}\left[\Phi - \Gamma G_{infty}\right] + W_x + G_{infty}^T W_u G_{infty}$$

The above ARE acn be solved in matlab by using the function $dlqr$. Then the controller can be implemented as

$$u(k) = -G_\infty x(k)$$

State Estimation

In the control law designed in previous section, we nead the state measurements at k^{th} instant. Generally the states are not measured. Hence, we need a way to estimate them. We can develop a state estimator as follows,

$$e(k) = y(k) - C\hat{x}(k|k-1)$$

$$\hat{x}(k+1|k) = \Phi\hat{x}(k|k-1) + \Gamma_u u(k) + L_p e(k)$$

Where L_p is the observer gain. We can use Luenberger observer or Kalman observer, which is based on solving Algebric Riccati Equations for observer. We can obtain the L_p gain matrix by using the command $kalman$. Once we estimate the states, we can close the state feedback loop and generate the control inputs as follows,

$$u(k) = -G_\infty \hat{x}(k|k-1)$$

In this way, observer is designed.

Both, the Controller and Observer can be designed independently and combined later. By separation principle, designing stable feedback law and stable observer independently will ensure the nominal closed loop stability. This makes our task easy as we can develop the controller and observer gains independently.

Regulation and Tracking

Suppose we are designing LQ controller for the following system,

$$x(k+1) = \Phi x(k) + \Gamma u(k) + \Gamma_\beta \beta_s$$

$$y(k) = Cx(k) + C_\eta \eta_s$$

where β_s is the input disturbance vector and η_s represents output disturbance vector. It is desired to control this system at an arbitrary setpoint r as $k \to \infty$. The steady state behaviour of the system is given by

$$x_s = \Phi x_s + \Gamma u_s + \Gamma_\beta \beta_s$$

$$x_s = (I - \Phi)^{-1}(\Gamma u_s + \Gamma_\beta \beta_s)$$

similarly, we can find r at steady state as

$$r = C(I - \Phi)^{-1}(\Gamma u_s + \Gamma_\beta \beta_s) + C_\eta \eta_s$$

$$r = K_u u_s + K_\beta \beta_s + C_\eta \eta_s$$

where $K_u = C(I - \Phi)^{-1}\Gamma$ and $K_\beta = C(I - \Phi)^{-1}\Gamma_\beta$

Now, we can define new perturbation variables as $\Delta x(k) = [x(k) - x_s]$, $\Delta u(k) = [u(k) - u_s]$ and $\Delta y(k) = [y(k) - r]$, and develop a transformed system as

$$\Delta x(k+1) = \Phi \Delta x(k) + \Gamma \Delta u(k)$$

$$\Delta y(k) = C \Delta x(k)$$

and develop an LQ control law of the form $\Delta u(k) = -G_\infty \Delta x(k)$
hence, the control law for the original system becomes,

$$u(k) = u_s - G_\infty [x(k) - x_s]$$

The above equations are implemented in MATLAB for Tracking control,

SIMULATIONS

The Controller gain was obtained using the $dlqr$ command. In the objective finction, W_x and W_u were chosen as diagonal matrix with all diagonal elements equal to 1.5 except $W_x(3,3) = 2$, $W_x(6,6) = 5$, $W_x(9,9) = 10$ (W_x matrix is 12x12 matrix in our case and showing it here would be difficult). Similarly, W_u weighing matrix is chosen as a diagonal matrix $W_u = 10I_4$, as we have 4 inputs. These are the tuning parameters and can be set as desired. $dlqr$ command solves the ARE and gives G_∞.

To determine observer gain Kalman Predictor is used. Q is taken as white noise gaussian with variance of 0.015 and $R = diag([0.50.60.40.2])$. We obtain the observer gain as L_p. The setpoint changes were introduced and sampling period was 3.0936 secs, with the simulation time as 775 secs, and sampling instants $N_s = 250$. Setpoints were changed at various levels (indicated in graphs) and it was observed that excellent output tracking is achieved. It was also observed that the state estimation errors were very less in the order of 0.1.

Figure (1) shows the Output Tracking.

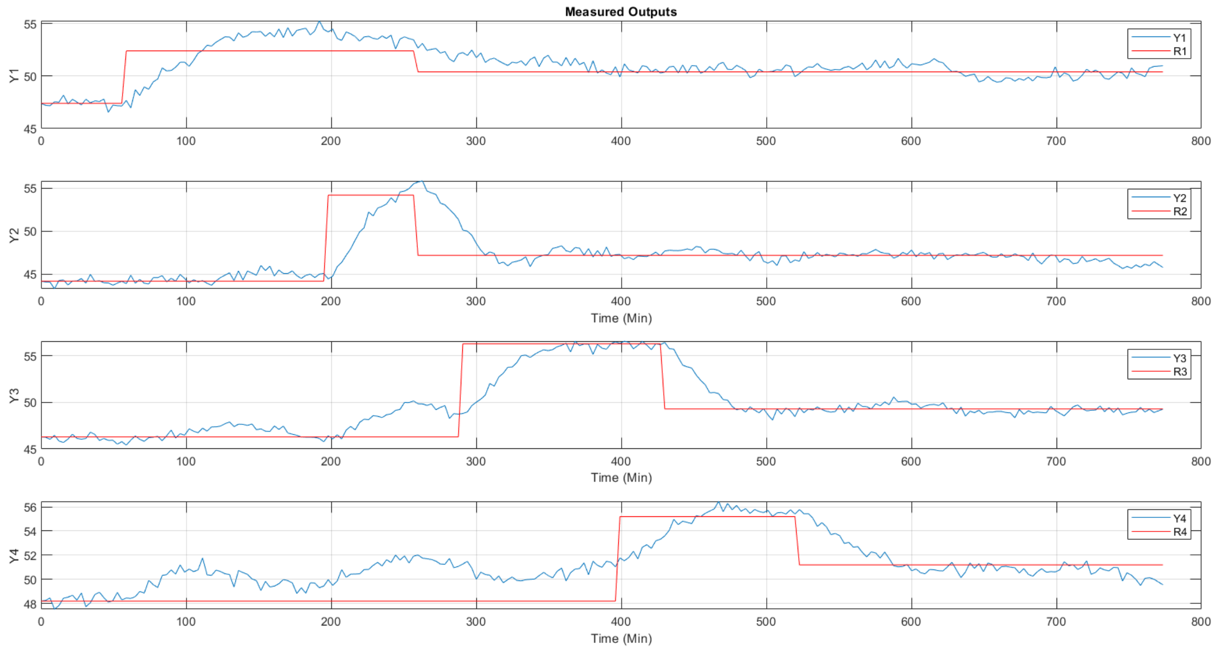

Figure 1: Outputs of the System

Figure (2) shows few states of the system and Figure (3) shows the state estimation errors. Figure (4) shows the control effort $u(k)$.

CONCLUSIONS

- LQOC was designed. Here, Optimal Quadratic Controller Gain G_{∞} and Kalman Estimator gain L_p were designed.
- Simulations were performed on state space models which were developed in previous weeks.
- Excellent Set point tracking was achieved. Controller was able to track multiple set point changes applied at different time instants.
- The control inputs generated were also within the specified limit. Max was 30 for all inputs.

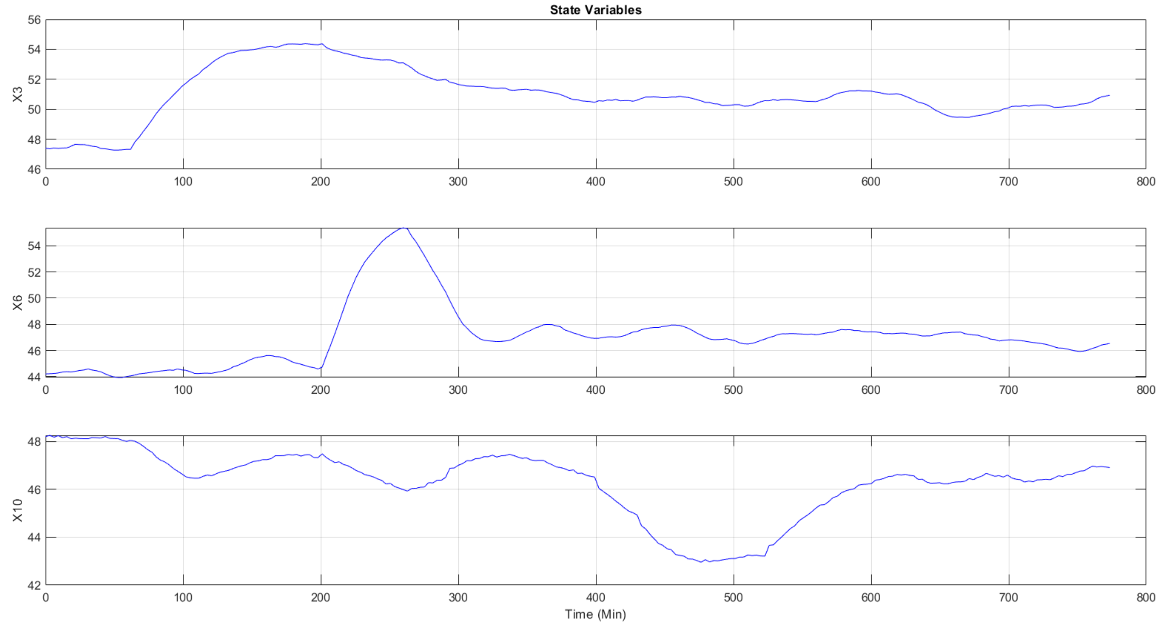

Figure 2: Few states of the system

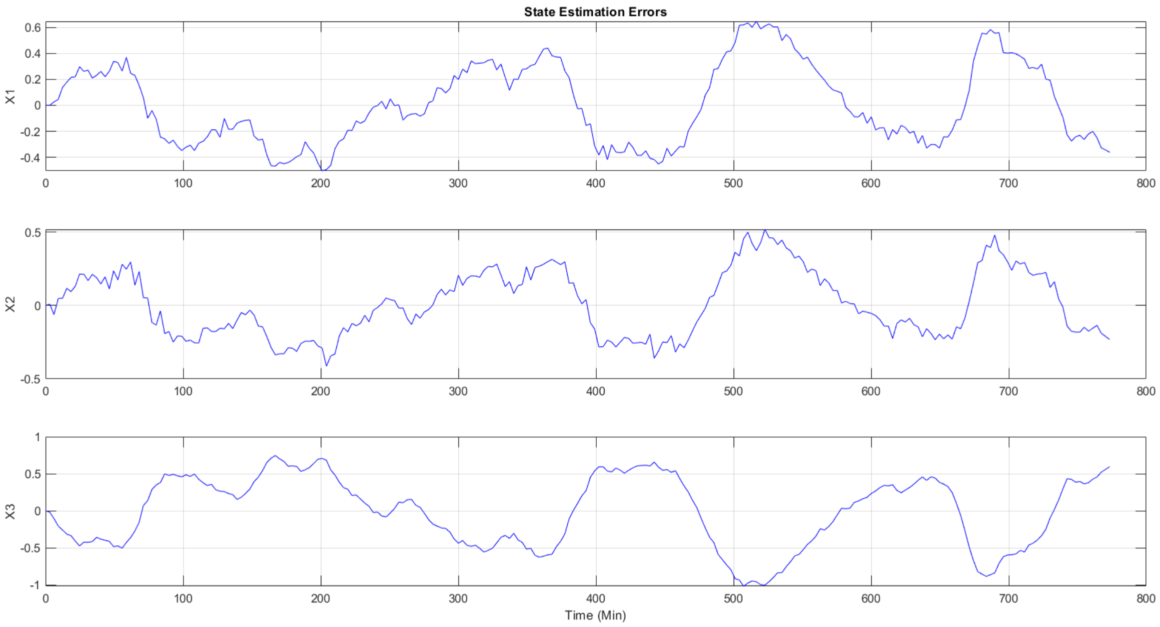

Figure 3: State Estimation Errors

- The State Estimation errors were also very small in the order of 0.1.

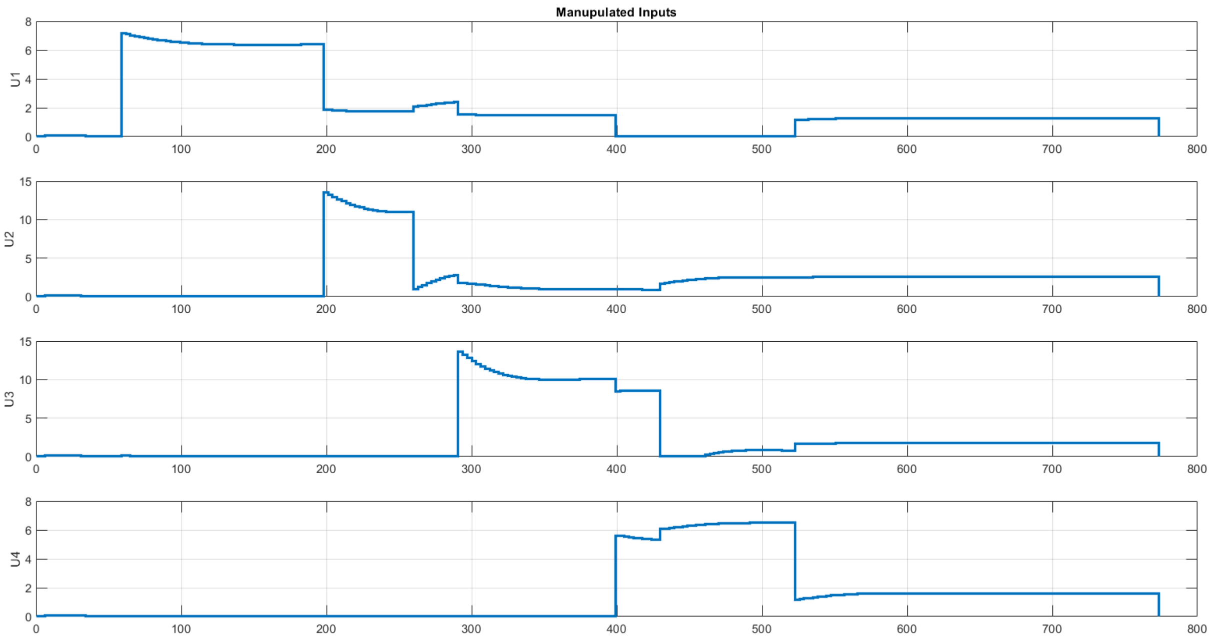

Figure 4: Control inputs generated by controller

- We have designed the Controller and Observer independently, and combined later. The nominal closed loop system was stable, confirming the separation principle.

References

1. CL686: Advance Process Control, IIT Bombay
2. Linear-quadratic (LQ) state-feedback regulator for discrete-time state-space system, MATLAB
3. Kalman filter for state estimation, MATLAB
4. K.J. Åstrom, B. Wittenmark - Computer-Controlled Systems - Theory and Design- Prentice Hall (1997)

SC 626

SYSTEMS AND CONTROL ENGINEERING LABORATORY

MIMO DECOUPLED CONTROLLERS

Amey Samrat Waghmare

203230013

Systems and Control Engineering,

Indian Institute of Engineering, Bombay

April 20, 2021

AIM

To design controllers for Multi Input Multi Output system using

- Simplified Decoupling
- Normalized Decoupling
- Inverted Decoupling Internal Model Control

MIMO SYSTEM

In most of the MIMO systems, couplings between input and output signals are present which may complicate the feedback controller design. Each output is affected by each input in this case and this may cause the response to deviate from the actual reference. Hence it is important to minimize the interactions caused due to these couplings. Decouplers are used to reduce these multiloop interactions so that desired response is achieved.
For designing the Simplified decoupler, RGA analysis is required which decides the pairing. For our MIMO system, this analysis is already done in week 3 and it was found that the diagonal pairing is the best pairing for designing decoupler.

We are going to design the decouplers for the 2x2 system corresponding to $u_1 - y_1$ and $u_4 - y_4$ and the Transfer Function matrix is shown below,

$$G(s) = \begin{bmatrix} \frac{1.004}{44.4867s+1}e^{-3.0899s} & \frac{0.2993}{37.652s+1}e^{-5.2612s} \\ \frac{0.5095}{69.7322s+1}e^{-0.8568s} & \frac{0.9}{26.1158s+1}e^{-2.0074s} \end{bmatrix} \tag{1}$$

SIMPLIFIED DECOUPLER

The simplified Decoupler takes the form as shown in Figure (1). Here, we need to determine the decoupler transfer functions so that the overall decoupled system is of a diagonal form so that the interactions are not present.

The decoupler takes the following form

$$D(s) = \begin{bmatrix} 1 & d_{12}(s) \\ d_{21}(s) & 1 \end{bmatrix}$$

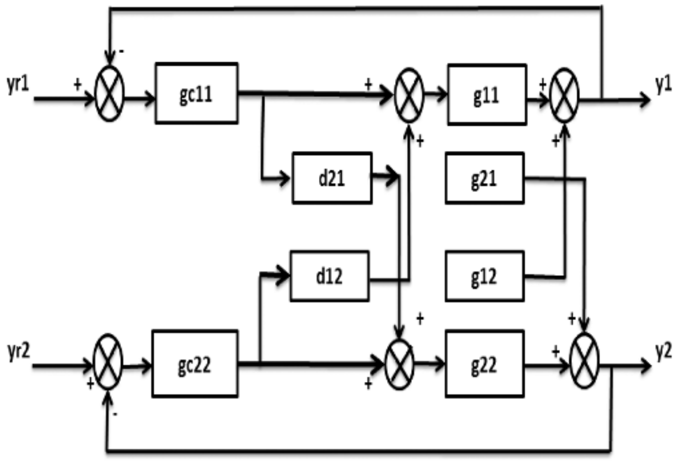

Figure 1: Simplified Decoupler

The resultant system will be,

$$G_p(s)D(s) = \begin{bmatrix} g_{11}^*(s) & 0 \\ 0 & g_{22}^*(s) \end{bmatrix}$$

Which were obtained to be,

$$g_{11}^*(s) = \frac{1.004}{1+44.4867s}e^{-3.0899s} - \frac{0.1694(26.1158s+1s)}{(37.652s+1s)(69.7322s+1)}e^{-5.2612s}$$

$$g_{22}^*(s) = \frac{0.9}{26.1158s+1}e^{-1.1506s} - \frac{0.1518(44.4867s+1s)}{(69.7322s+1)(37.652s+1)}e^{-5.262s}$$

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,

$$g_{11}^*(s) = \frac{0.836}{1+38.404s}e^{-3.334s}$$

$$g_{22}^*(s) = \frac{0.751}{1+22.018s}e^{-3.326s}$$

Correspondingly, the decoupler obtained is,

$$D(s) = \begin{bmatrix} 1 & \frac{-0.2981(44.4867s+1)}{(37.652s+1)}e^{-2.1713s} \\ \frac{-0.5661(26.1158s+1)}{(69.7322s+1)} & e^{-1.1506s} \end{bmatrix}$$

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.

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, $T_{cl} = \frac{T}{3}$ as a design criterion. The PI parameters are designed for FOPTD $g^*_{11}(s)$ and $g^*_{22}(s)$ and the values obtained are,

- $k_{11} = 2.847$
- $T_{i_{11}} = 38.404$
- $k_{22} = 7.332$
- $T_{i_{22}} = 22.018$

The decoupler were implemented in MATLAB and the obtained response is shown in Figure (2).

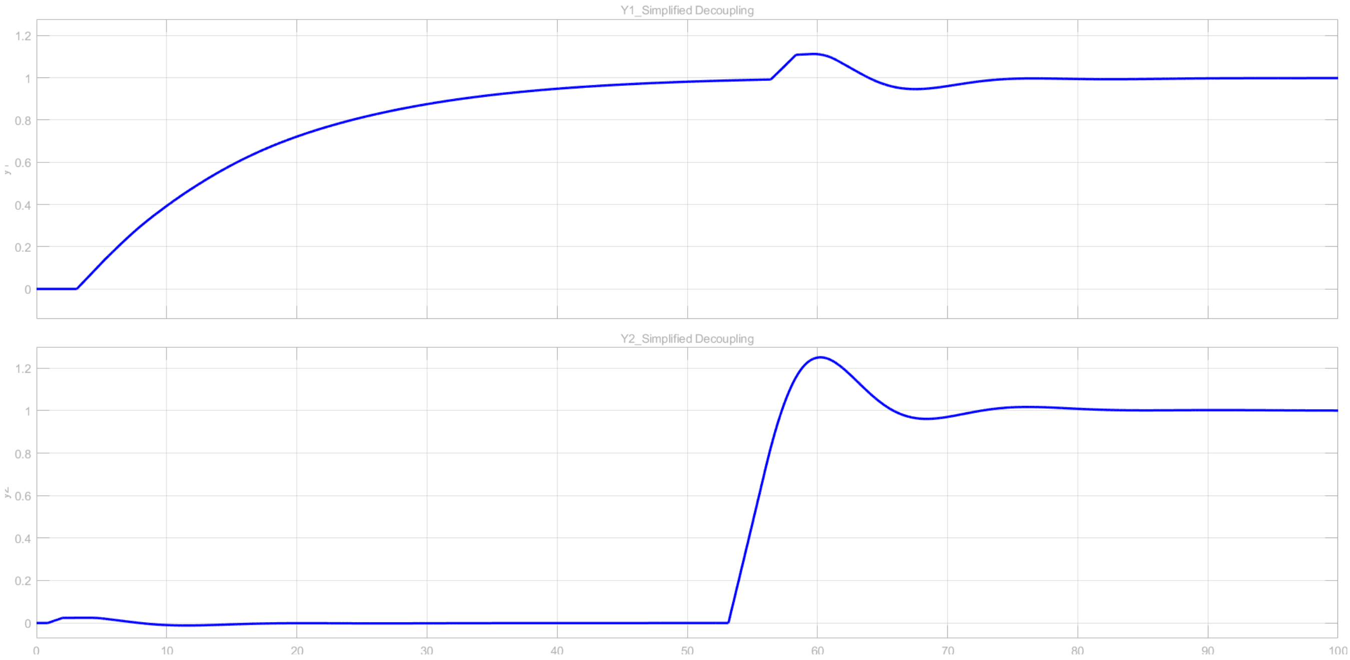

Figure 2: Response: Simplified Decoupling

Normalized Decoupling

Here, the objective is to add a transfer function block $G_I(s)$ between the diagonal controller $G_c(s)$ and the plant $G(s)$ so that the decoupled process $G_R(s) = G(s)G_I(s)$ as seen from controller output is diagonal. For this, the expression of decoupler is,

$$G_I(s) = G^{-1}(s)G_R(s)$$

Here, the problem is that this may result in a complicated $G_I(s)$. Even if we use simplified decoupling or ideal decoupling, it may result in complex $G_R(s)$ as seen in previous section. Hence, the decoupler is designed by using Equivalent Transfer Function (ETF) and this acts as an approximate inverse for the plant transfer function matrix. Then the decoupler Transfer Function are appropriately selected to maintain stability, causality and properness.
Here, we assume that the process transfer function is in the FOPTD form. We can define the Normalized Gain for a particular transfer function as,

$$k_{N,ij} = \frac{k_{ij}}{\tau_{ij} + \theta_{ij}}$$

where the k_{ij}, τ_{ij} and θ_{ij} are the parameters of FOPTD plant process. In this way we can calculate normalized gains for all transfer functions and combine them in a Normalized Gain Array K_N.

Similar to RGA, we can also define Normalized Relative Gain array between output and input pair as follows,

$$RNGA = K_N \otimes K_N^{-T}$$

where, the operator $\otimes$ is the element wise multiplication.
RNGA captures combined changes in both steady state and dynamic when other loops are open. When all the loops are closed, to capture the dynamic behaviour we determine the Relative Average Residence Time Array (RARTA).

$$RARTA = RNGA \odot \Lambda$$

where, the operator $\odot$ is element wise division.
Now, by using the above arrays, we can determine the elements of ETF, which takes the FOPTD form as follows,

$$\hat{k} = k \odot \Lambda$$

$$\hat{T} = RARTA \otimes T$$

$$\hat{L} = RARTA \otimes L$$

And then the inverse of ETF matrix is denoted by $\hat{G}(s)$,

$$\hat{G}(s) = \begin{bmatrix} \frac{1}{\hat{g}_{11}(s)} & \frac{1}{\hat{g}_{12}(s)} \\ \frac{1}{\hat{g}_{21}(s)} & \frac{1}{\hat{g}_{22}(s)} \end{bmatrix}$$

Based on this, we can determine the decoupler $G_I(s)$ as,

$$G_I(s) = \hat{G}^T(s) G_R(s)$$

We take the elements of $G_R(s)$ in the form,

$$g_{R,ii}(s) = \frac{k_{R,ii}}{\tau_{R,ii}+1} e^{-\theta_{R,ii}}$$

And this will give rise to the elements of the decoupler as,

$$G_{I,ij} = \begin{bmatrix} \frac{1}{\hat{g}_{11}(s)} & \frac{1}{\hat{g}_{21}(s)} \\ \frac{1}{\hat{g}_{12}(s)} & \frac{1}{\hat{g}_{22}(s)} \end{bmatrix} x \begin{bmatrix} g_{R,11}(s) & 0 \\ 0 & g_{R,22}(s) \end{bmatrix}$$

$$G_{I,ij} = \begin{bmatrix} \frac{g_{R,11}(s)}{\hat{g}_{11}(s)} & \frac{g_{R,22}(s)}{\hat{g}_{21}(s)} \\ \frac{g_{R,11}(s)}{\hat{g}_{12}(s)} & \frac{g_{R,11}(s)}{\hat{g}_{22}(s)} \end{bmatrix}$$

Now, the elements of $G_R(s)$ are chosen such that the elements of $G_I(s)$ are stable, proper and causal. Stability and properness is is already ensured as the model of the system is in FOPTD form. For causality, we must ensure that $(\theta_{R,ii} - \hat{\theta}_{ij}) \geq 0$, and this is ensured by chosing,

$$\theta_{R,ii} = Max_{j=1,2}\hat{\theta_{ij}}$$

.

Now, the Controller can be designed for the process $G_R(s)$. We have used Lambda Tuning for this purpose. The necessary steps to identify the decoupler are indicated below, corresponding to our Two Input Two Output system in Equation (1).

$$RGA = \begin{bmatrix} 1.2030 & -0.2030 \\ -0.2030 & 1.2030 \end{bmatrix}$$

$$K_N = \begin{bmatrix} 0.0211 & 0.0070 \\ 0.0072 & 0.0320 \end{bmatrix}$$

$$RNGA = \begin{bmatrix} 1.0805 & -0.0805 \\ -0.0805 & 1.0805 \end{bmatrix}$$

$$RARTA = \begin{bmatrix} 0.8982 & 0.3967 \\ 0.3967 & 0.8982 \end{bmatrix}$$

From these, the parameters of Effective Transfer Function were calculated by using the formulas in previous section, and the resultant ETF is shown below,

$$\hat{G}^T(s) = \begin{bmatrix} \frac{39.9577s+1}{0.8346} e^{2.7753s} & \frac{27.6652s+1}{-2.5094} e^{0.3399s} \\ \frac{14.9379s+1}{-1.4745} e^{2.0873s} & \frac{23.4571s+1}{0.7481} e^{1.803s} \end{bmatrix}$$

The Diagonal Forward Transfer Function becomes,

$$G_R(s) = \begin{bmatrix} \frac{1}{39.9577s+1} e^{-2.7753s} & 0 \\ 0 & \frac{1}{23.4571s+1} e^{-1.803s} \end{bmatrix}$$

We have to design PI parameters for this diagonal system.
And the Decoupler becomes

$$G_I(s) = \begin{bmatrix} \frac{1}{0.8346} & \frac{27.6652s+1}{-2.5094(23.4571s+1)} e^{-0.3399s} \\ \frac{14.9379s+1}{-1.4745(39.9577s+1)} e^{-2.0873s} & \frac{1}{0.7481} \end{bmatrix}$$

The PI parameters are,

- $k_{11} = 2.4827$
- $T_{i_{11}} = 39.9577$
- $k_{22} = 2.438$
- $T_{i_{22}} = 23.4571$

Figure (3) shows the response of the system after the decoupler is implemented.

INVERTED DECOUPLING INTERNAL MODEL CONTROL

For a square MIMO system, we can design an Inverted Decoupler scheme $Q(s)$ to obtain a decoupled response. However, in this scheme, $Q(s)$ is split into two blocks, $Q_d(s)$ and $Q_o(s)$. This is shown in Figure (4). $Q_d(s)$ is the direct path between error signal and control signal, and $Q_o(s)$ is in a feedback loop in opposite direction. $Q_d(s)$ is chosen such that it has only n non zero elements, whereas $Q_o(s)$ is chosen such that it has n zero elements, with transpose

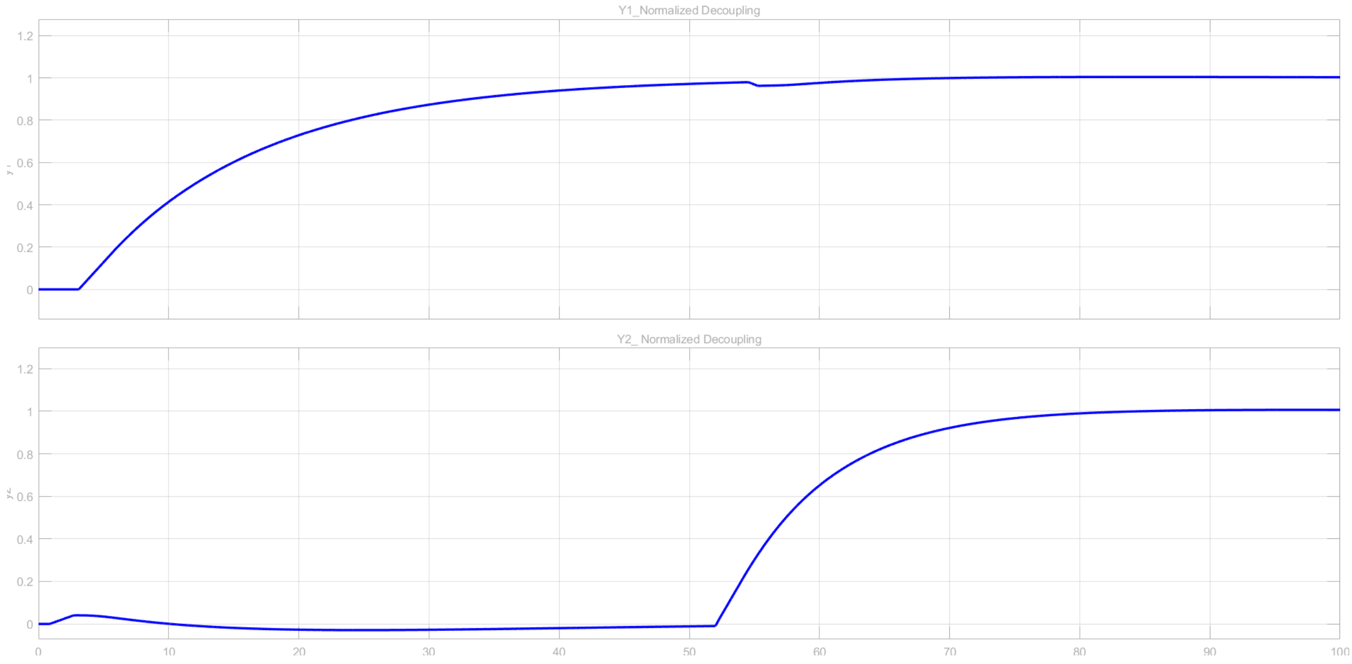

Figure 3: Response: Normalized Decoupling

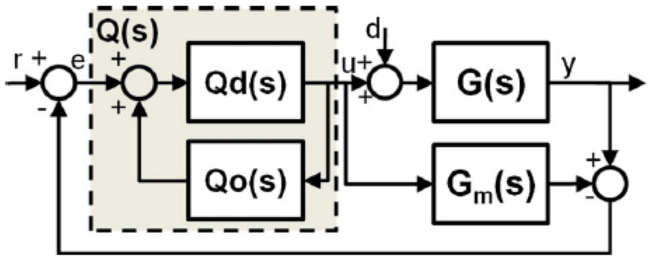

Figure 4: Inverted Decoupling Internal Model Control

to non zero elements of $Q_d(s)$.
Consider $T(s)$ is the transfer function matrix from reference to outputs, and it is in diagonal form, and it is given by,

$$T(s) = G(s)Q(s)$$

Now, we can write,

$$Q_d^{-1}(s) - Q_o(s) = T^{-1}(s)G(s)$$

for a 2x2 system, the above equation translates to,

$$\begin{bmatrix} qd_{11}(s) & 0 \\ 0 & qd_{22}(s) \end{bmatrix}^{-1} - \begin{bmatrix} 0 & qo_{12}(s) \\ qo_{21}(s) & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{qd_{11}(s)} & -qo_{12}(s) \\ -qo_{21}(s) & \frac{1}{qd_{22}(s)} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{qd_{11}(s)} & -qo_{12}(s) \\ -qo_{21}(s) & \frac{1}{qd_{22}(s)} \end{bmatrix} = \begin{bmatrix} \frac{g_{11}(s)}{t_1(s)} & \frac{g_{12}(s)}{t_1(s)} \\ \frac{g_{21}(s)}{t_2(s)} & \frac{g_{22}(s)}{t_2(s)} \end{bmatrix}$$

Hence, the elements of the decoupler can be chosen as,

$$qd_{11}(s) = \frac{t_1(s)}{g_{11}(s)}$$

$$qo_{12}(s) = \frac{-g_{12}(s)}{t_1(s)}$$

$$qo_{12}(s) = \frac{-g_{21}(s)}{t_2(s)}$$

$$qd_{22}(s) = \frac{t_2(s)}{g_{22}(s)}$$

However, we must also ensure that the above calculated elements are actually realizable. In other words, we mean to avoid poles in RHP and time advance elements in the decoupler elements. Suppose that the given transfer function model $G(s)$ is in FOPTD form, then the elements of the transfer function $T(s)$ takes the form,

$$t_i(s) = \frac{e^{-\theta_i s}}{\lambda_i s + 1}$$

where, λ_i is desired closed loop time constant for reference tracking and θ_i is taken as minimum time delay for a particular output. These values are taken by ensuring that the decoupler elements do not have any time advance element.

Even if by ensuring this we do not get any realizable elements, then we need to add extra dynamics to the original system. They will be added in terms of diagonal transfer function block $N(s)$ so that the new process becomes $G^N(s) = G(s)N(s)$

For our FOPTD system in Equation (1), the system gave rise to non realizable decoupler elements, hence, extra dynamics were added with $n_{11} = e^{-1.1506s}$ and $n_{22} = 1$ to get the following form,

$$G(s) = \begin{bmatrix} \frac{1.004}{44.4867s+1}e^{-4.2405s} & \frac{0.2993}{37.652s+1}e^{-5.2612s} \\ \frac{0.5095}{69.7322s+1}e^{-2.0074s} & \frac{0.9}{26.1158s+1}e^{-2.0074s} \end{bmatrix} \tag{2}$$

The transfer function $T(s)$ have elements,

$$t_1(s) = \frac{e^{-4.2405s}}{\lambda_1 s + 1}$$

$$t_2(s) = \frac{e^{-2.0074s}}{\lambda_2 s + 1}$$

And by using the above formulas, we get the decoupler elements as,

$$qd_{11}(s) = \frac{44.4867s + 1}{1.004(\lambda_1 s + 1)}$$

$$qd_{22}(s) = \frac{26.1158s + 1}{0.9(\lambda_2 s + 1)}$$

$$qo_{12}(s) = \frac{-0.2993(\lambda_1 s + 1)}{37.652s + 1} e^{-1.0207s}$$

$$qo_{21}(s) = \frac{-0.5095(\lambda_2 s + 1)}{69.7322s + 1}$$

λ is the tuning parameter. It is chosen based upon the desired rise time of the system $t_i(s)$. For $t_i(s)$, if we take the inverse laplace transform, we get

$$y_i(t) = 1 - e^{\frac{-(t-\theta_i)}{\lambda_i}}$$

for $t \geq \theta_i$ for rise time i.e time taken to reach to 90%,

$$0.9 = 1 - e^{\frac{-(t_{ri}-\theta_i)}{\lambda_i}}$$

$$2.3026\lambda_i + \theta_i = t_{ri}$$

$$\lambda_i = \frac{t_{ri} - \theta_i}{2.3026}$$

where, t_{ri} is the desired rise time.

For simulation of System (1), the desired rise times chosen are $t_{r1} = 10$ and $t_{r2} = 20$and we get the tuning parameters as,

$$\lambda_1 = 2.5013$$

$$\lambda_2 = 7.814$$

The obtained response is shown in Figure (5).

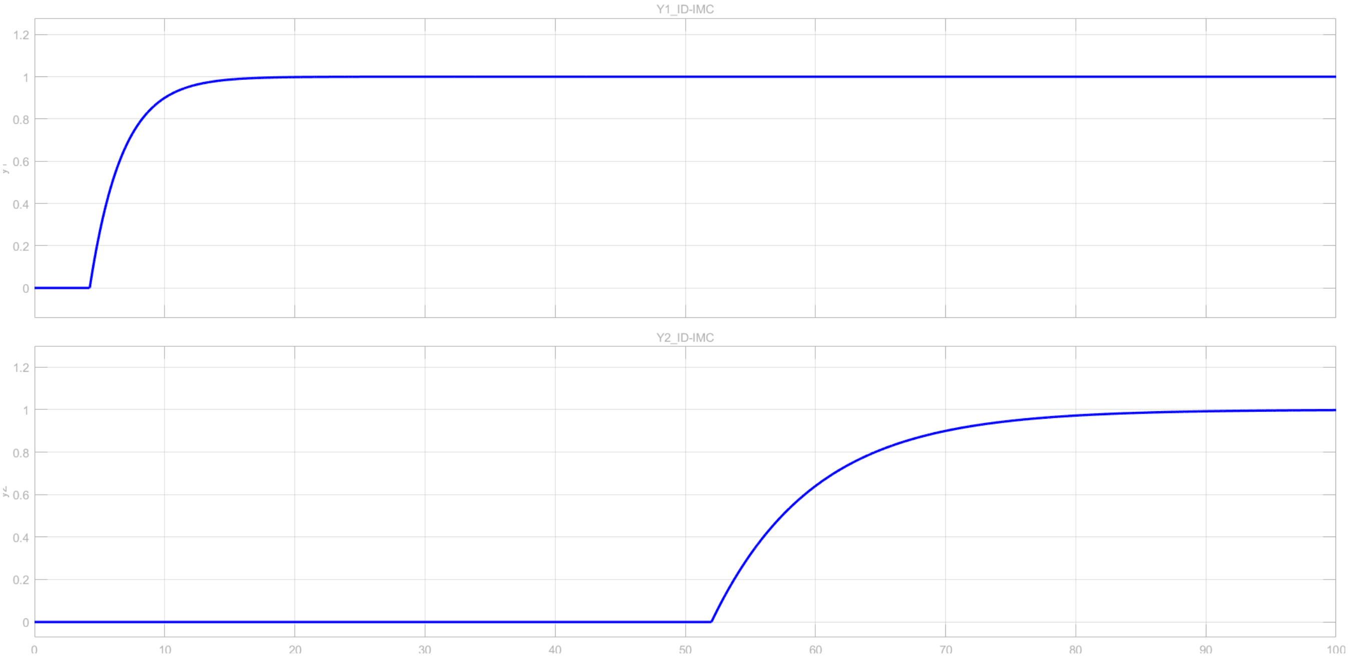

Figure 5: Response: Inverted Decoupling Internal Model Control

COMPARATIVE STUDY OF ALL THE DECOUPLERS

Figure (6) shows the response of the system for all decouplers. Here we see that for the simplified decoupler, outputs are affected by other loops like output y_1 is even affected by u_2. For other decouplers, the loop interactions were very less and negligible for Inverted decoupling Internal Model Control.

Figure (7) shows the control effort required to control the Two Input Two Output system. The control effort is very high for ID-IMC control for controlling y_1 and is moderate for Simplified Decoupling for y_2.

Following Table shows the RMSE values for the decouplers,

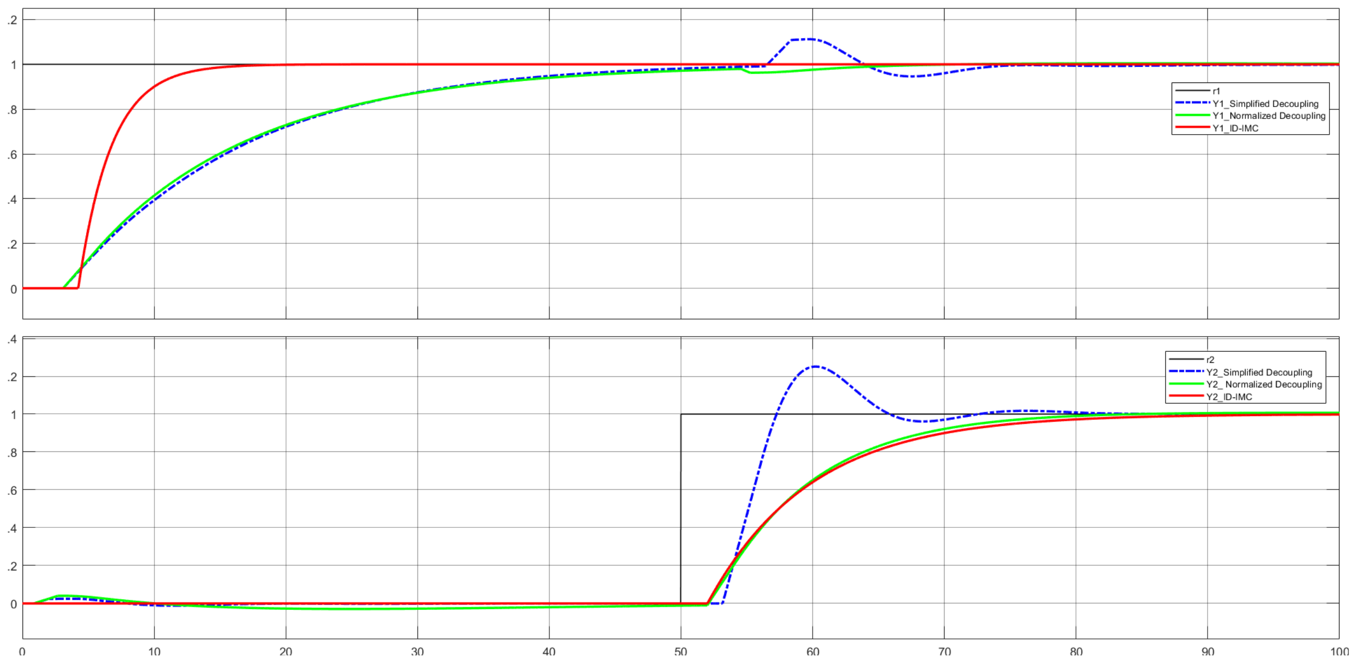

Figure 6: Comparison of Response

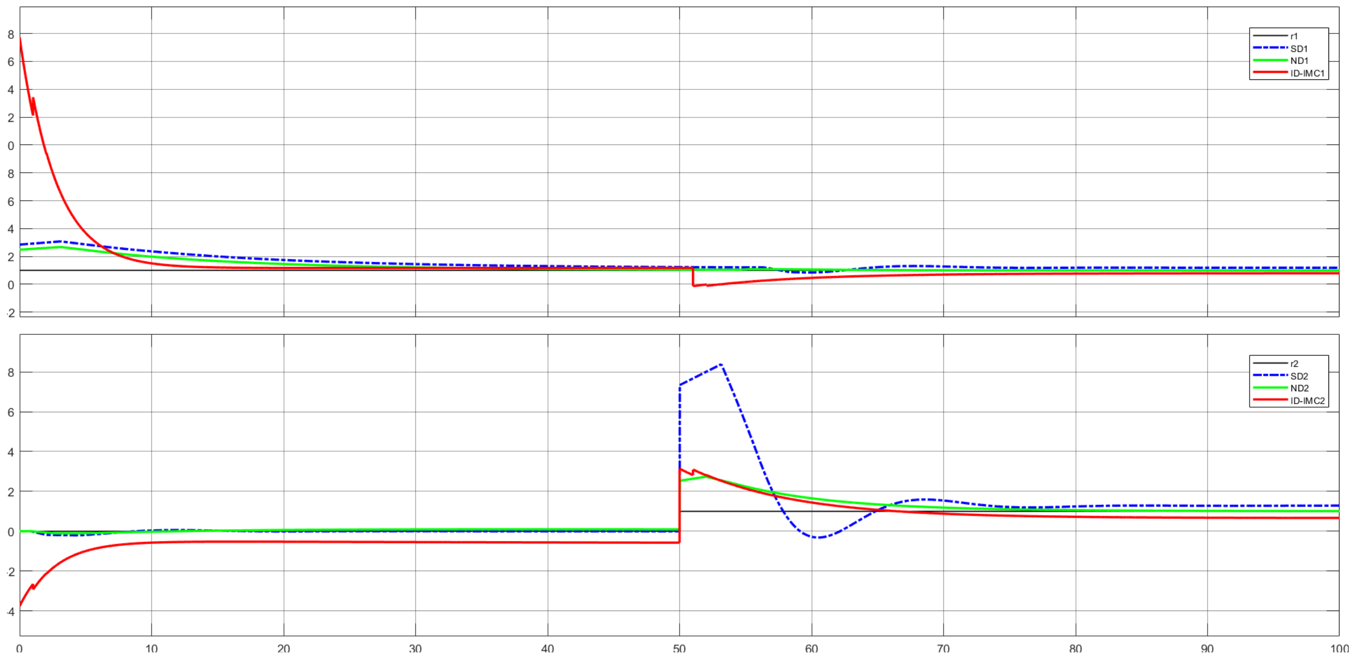

Figure 7: Control Inputs

RMSE		
-	y_1 (^oC)	y_2 (^oC)
Simplified Decoupling	0.3151	0.2174
Normalized Decoupling	0.3104	0.2451
ID-IMC	0.2346	0.2432

CONCLUSIONS

- Various Decouplers were designed for a 2x2 system.
- It was observed that designing the controllers for Simplified decoupling is very difficult as it resulted in very higher order system. We first had to reduce the order of system and then design a PI controller.
- Normalized Decoupling avoided the problem of simplified decoupling by the use of Effective Transfer Function. The Transfer Functions for which the controller needs to be designed were already in simple form so that we can design PI controllers easily.
- The obtained response indicated excellent control for the plant and even the squared errors were very less for all the decouplers.
- The Control efforts were high for ID-IMC and Simplified decoupling.

REFERENCES

1. SC645: Intelligent Feedback and Control Course, IIT Bombay
2. Normalized Decoupling New Approach for MIMO Process Control System Design, Wen-Jian Cai, Wei Ni, Mao-Jun He, and Cheng-Yan Ni
3. Inverted decoupling internal model control for square stable multivariable time delay systems, Juan Garridoa, Francisco Vázqueza, Fernando Morilla