Modern Control Systems

Laboratory

AWABULLAH SYED

11/27/2020

Name: Awabullah M. Syed

Student Number: 2002 000 33

Session: 2020 / 2021

Module Code: ACS 6129

Module Assessor: Dr Lingzhong Guo

Contents

Session 1 (Week 5) – System Analysis

Introduction

This section of the report aims to conduct system analysis of the chemical reactor. Figure 1 shows the state space representation of the chemical reactor system which was used to carry out system analysis. Chemical reactor system was implemented in MATLAB Simulink and is displayed in Figure 2. This was used to analyze the system behaviour to the unit step response.

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\\ \dot{x}_4(t)\end{bmatrix} = \begin{bmatrix}1.4 & -0.2 & 6.7 & -5.7\\ -0.6 & -4.3 & 0 & 0.7\\ 1.1 & 4.3 & -6.7 & 5.9\\ 0 & 4.3 & 1.3 & -2.1\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\\ x_4(t)\end{bmatrix} + \begin{bmatrix}0 & 0\\ 5.7 & 0\\ 1.1 & -3.1\\ 1.1 & 0\end{bmatrix}\begin{bmatrix}u_1(t)\\ u_2(t)\end{bmatrix}$$

$$\begin{bmatrix}y_1(t)\\ y_2(t)\\ y_3(t)\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & -1\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\\ x_4(t)\end{bmatrix}$$

Figure 1 – State Space representation of the chemical reactor

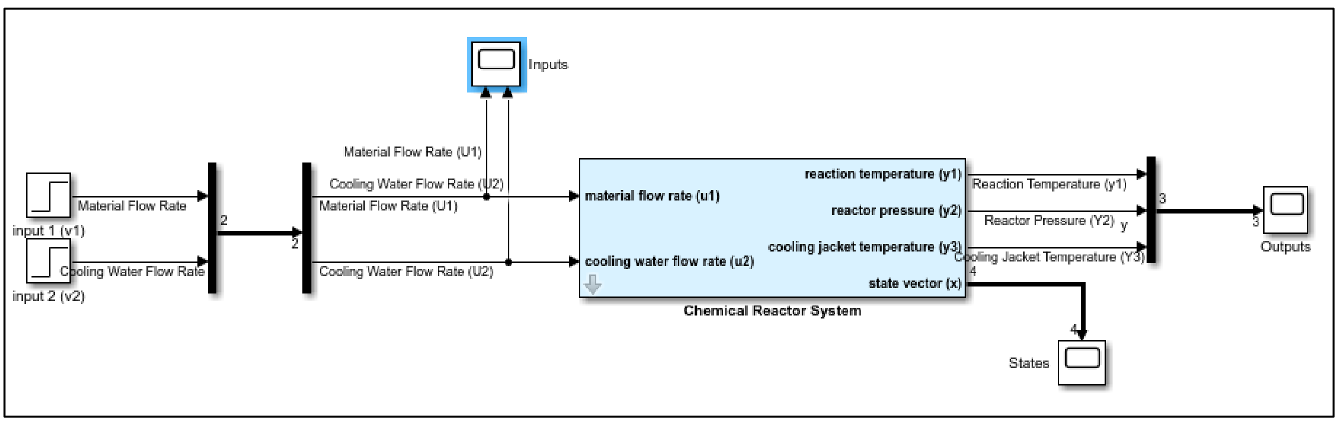

Figure 2 - Chemical reactor system implemented in MATLAB Simulink

Open-Loop Step Response of the Chemical Reactor System

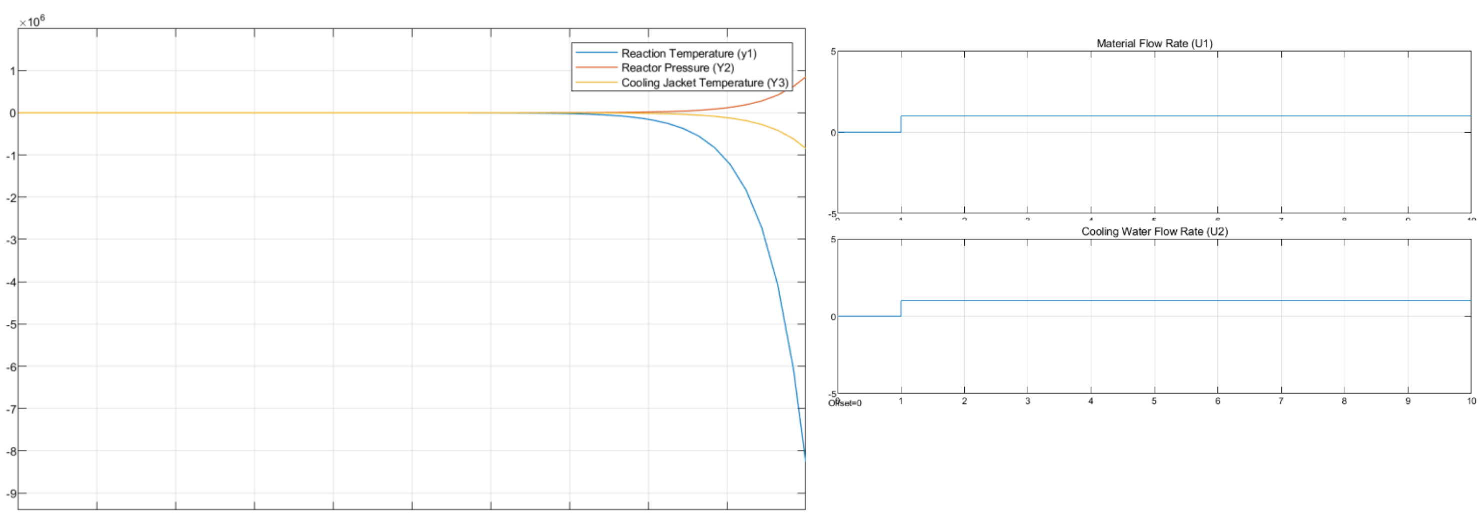

Figure 3 - Open Loop Step Response of the chemical reactor system

Figure 3 shows the step response of the chemical reactor system which indicates that the system was unstable since the open-loop response of the reaction temperature, y_1, Reactor Pressure, y_2, and Cooling Jacket Temperature, y_3, goes to infinity with time. In other words, the system was not bounded.

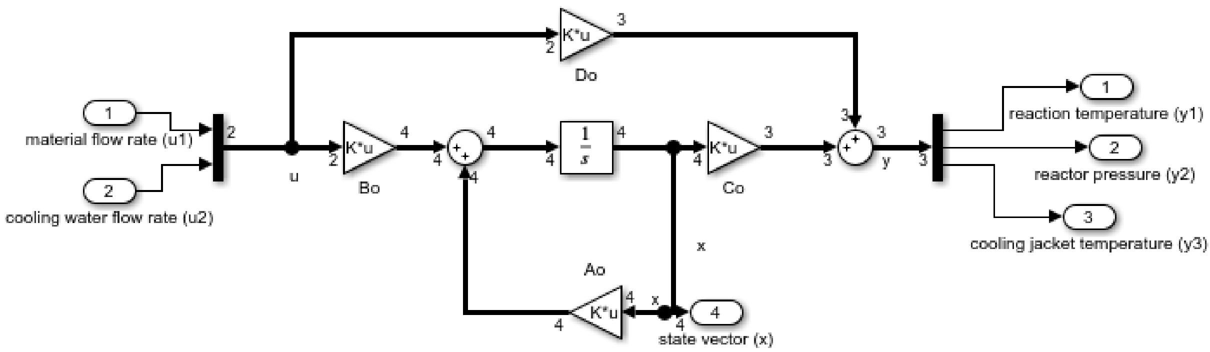

Figure 4 - Internal dynamics of the chemical reactor

It was evident from the internal structure of the chemical reactor shown in Figure 4 and the state-space representation shown in Figure 1 that the inputs and outputs of the system are coupled. This meant that each input signal did not excite certain output in isolation from any other input. Therefore, modal decomposition was carried out to make the system uncoupled.

Reachability and observability of the system

The state-space representation shown in Figure 1 was used to determine the matrices, A, B, C and D. shown in Equation 1 to 4, respectively.

System Matrices

$$A = \begin{bmatrix} 1.4 & -0.2 & 6.7 & -5.7 \\ -0.6 & -4.3 & 0 & 0.7 \\ 1.1 & 4.3 & -6.7 & 5.9 \\ 0 & 4.3 & 1.3 & -2.1 \end{bmatrix} \qquad Eq.(1)$$

$$B = \begin{bmatrix} 0 & 0 \\ 5.7 & 0 \\ 1.1 & -3.1 \\ 1.1 & 0 \end{bmatrix} \qquad Eq(2)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \qquad Eq.(3)$$

$$D = [0] \qquad Eq.(4)$$

MATLAB function $ctrb$ and $obsv$ were used in conjunction with the $rank$ function to determine the reachability and observability of the system. The system was reachable since the rank of reachability matrix, W_r was equal to the dimension of the state vector matrix, A as supported by the result obtained via MATLAB code shown in the appendix. The rank of the reachability matrix was four which was equal to the dimension of the dynamic matrix A. The system was observable since the rank of the observable matrix, W_o was equal to the dimension of dynamic matrix A. Hence, it was deduced that the chemical reactor system was reachable and observable.

Model Decomposition of the model defined by system matrices

MATLAB function, $canon$ was used to transform the system into model canonical form and matrices are described from Equation 7 to 10.

$$\dot{Z}(t) = \overline{A}\, z(t) + \overline{B}\, u(t) \qquad Eq.(5)$$

$$y(t) = \overline{C}\, z(t) + \overline{D}\, u(t) \qquad Eq.(6)$$

$$\overline{A} = \begin{bmatrix} 2.044 & 0 & 0 & 0 \\ 0 & 0.065 & 0 & 0 \\ 0 & 0 & -5.093 & 0 \\ 0 & 0 & 0 & -8.716 \end{bmatrix} \qquad Eq.(7)$$

$$\overline{B} = \begin{bmatrix} -0.52 & 0.62 \\ 3.6 & -0.52 \\ -5.27 & 0.282 \\ 0.06 & 0.83 \end{bmatrix} \qquad Eq.(8)$$

$$\overline{C} = \begin{bmatrix} -3.6 & -0.7 & -0.117 & 2.626 \\ 0.3851 & 0.36 & -0.87 & 0.237 \\ -0.34 & -0.096 & -0.044 & -3.48 \end{bmatrix} \qquad Eq.(9)$$

$$\overline{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad Eq.(10)$$

$$\lambda = \begin{bmatrix} 2.04 \\ 0.0655 \\ -5.09 \\ -8.71 \end{bmatrix} \qquad Eq.(11)$$

The Eigenvalues obtained are shown in Equation 11 which suggest that two modes $(2.04\,\&\,0.0655)$ were unstable since they belong to the positive/right side in the complex plane. Hence, the system was unstable. However, all modes are stabilizable including the unstable ones since they are reachable. This is because all modes pass through the state variable, $\overline{B}$ thus, making them reachable. Therefore, the feedback method can be used to make the unstable modes stable. Furthermore, from Equation 9 it was deduced that all the modes were observable.

Session 2 Part I (Week 6)

Introduction

This section of the report aims to develop state feedback and feedforward control system via pole placement method and analyze steady-state error to the step reference signal. Furthermore, the correlation between the dominant desired closed-loop eigenvalues and the step response of the system will be investigated.

State feedback control gain matrix, K – Pole Placement

Desired poles shown in Equation 12 were used to deduce the required gain matrix, K via pole placement method. The gain matrix was then implemented in MATLAB Simulink as shown in Figure 5 to make the chemical reactor system stable.

$$Eigenvalues_{Desired} = \begin{bmatrix} -1+0.4i \\ -1-0.4i \\ -2+0.0i \\ -3+0.0i \end{bmatrix} \qquad Eq.(12)$$

$$K = \begin{bmatrix} -0.1793 & -0.2354 & -0.0911 & 0.2056 \\ -0.7088 & -0.4603 & 1.1239 & -1.3256 \end{bmatrix} \qquad Eq.(13)$$

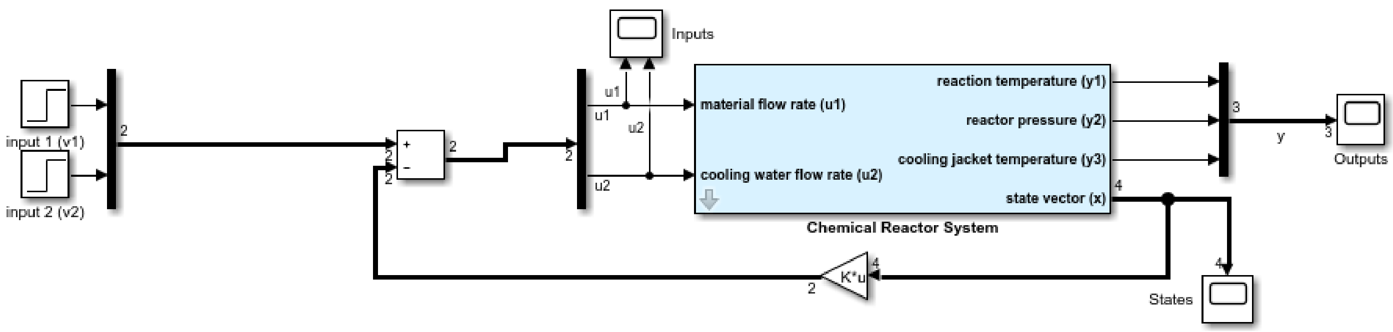

Figure 5 - Closed Loop feedback control system

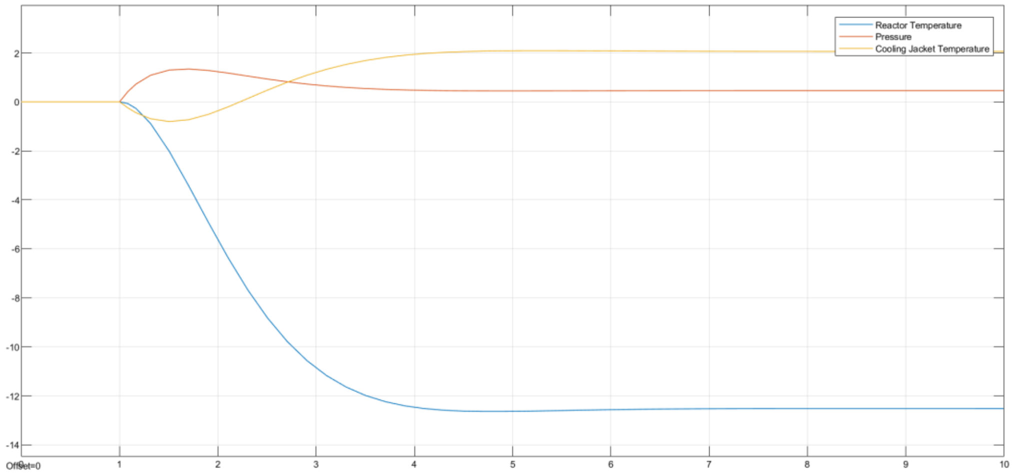

Figure 6 - Step response of the closed loop chemical reactor system with the matrix, K implemented

The state feedback control gain matrix K was calculated via pole placement method and was implemented in Simulink to obtain a step response of the closed loop system shown in Figure 6. Figure 6 suggest that the closed loop system of the chemical reactor was stable. Therefore, an unstable open-loop system shown in Figure 3 was made stable through the implementation of a closed loop feedback system with the gain matrix, K.

$$A_{Closed\ loop} = (A - BK)\lambda \qquad Eq. (14)$$

$$Eigenvalues_{Closed\ loop} = \begin{bmatrix} -1.0 + 0.4i \\ -1.0 - 0.4i \\ -2.0 + 0.0i \\ -3.0 + 0.0i \end{bmatrix} \qquad Eq. (15)$$

Equation 14 was used to confirm the eigenvalues/poles of the closed-loop system with feedback control gain matrix, K implemented. Eigenvalues shown in Equation 15 were deduced and were the same as the desired eigenvalues shown in Equation 12 however, it was different from open-loop eigenvalues shown in Equation 11.

Feedforward gain, K_r

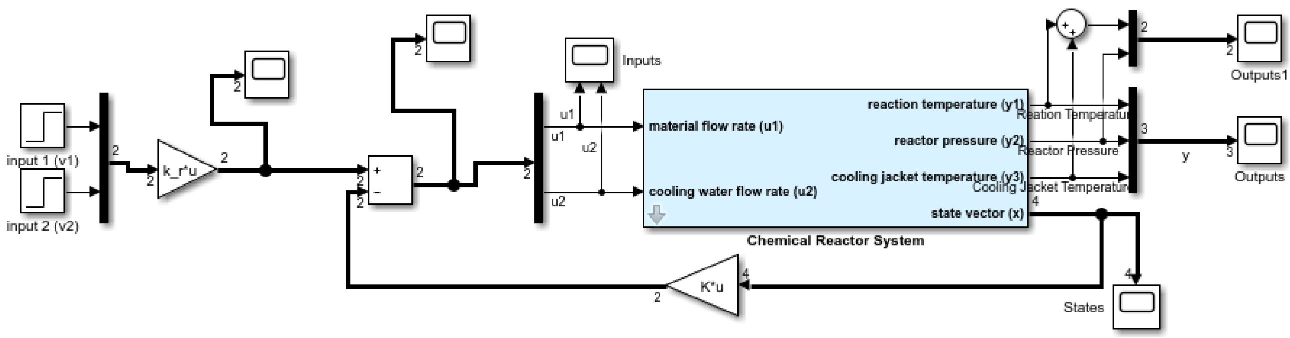

Figure 7 - Closed loop feedforward system

A feedforward gain, K_r was designed with C_r was defined in terms of $y_r = C_r x$. The output response to the step input reference signal is shown in Figure 8. It was deduced that the steady-state error for each of the outputs, y_1, y_2 and y_3 was not achieved since the reference output did not track the unit step input, (1) at steady state.

$$C_r = \begin{bmatrix} y_1 + y_3 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad Eq. (16)$$

$$k_r = \begin{bmatrix} -0.0687 & 0.4956 \\ -0.063 & -0.4737 \end{bmatrix} \qquad Eq. (17)$$

Step Response of individual outputs with feedforward K_r shown in Equation 17

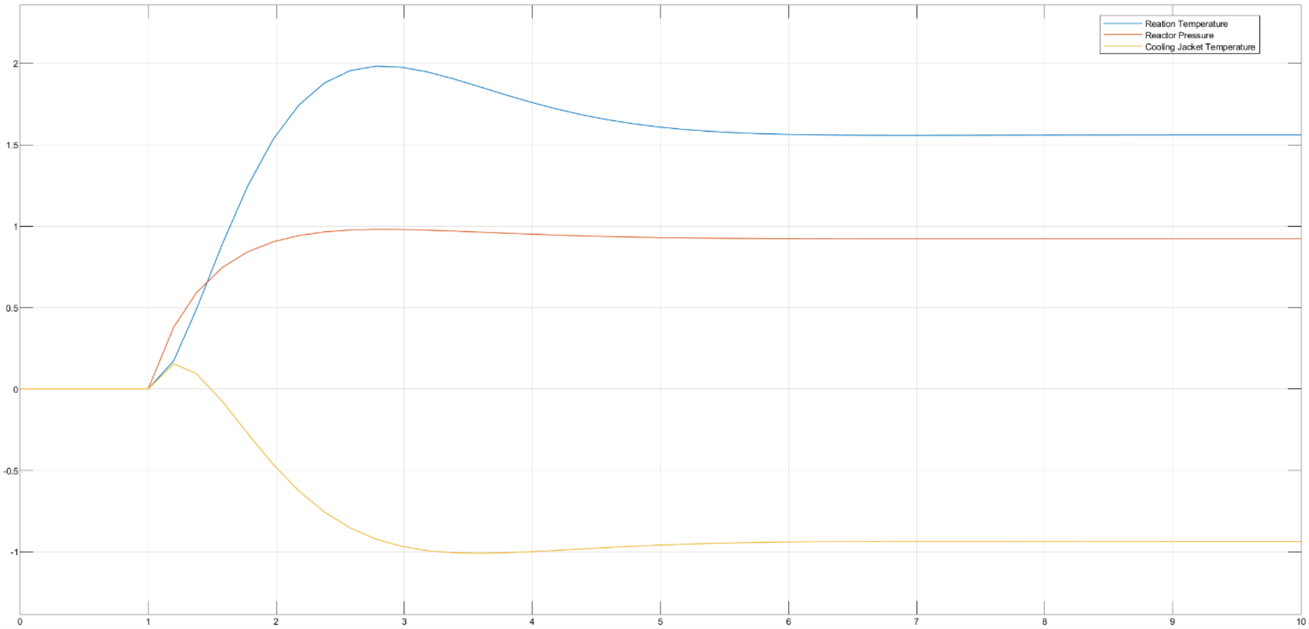

Figure 8 – Step response of the closed loop system with feedforward gain, K_r (Individual Outputs)

Step Response of signal $(y_1 + y_3)$ & y_2

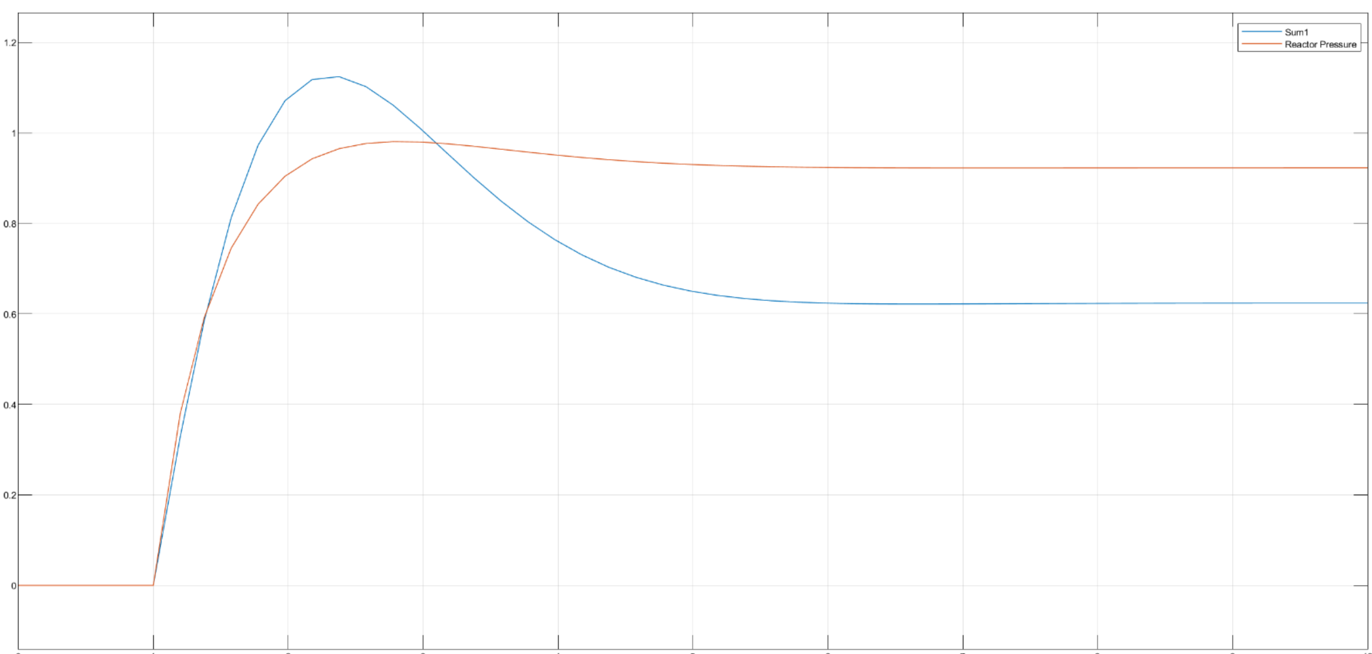

Figure 9 - Signal $y_1 + y_3$ & y_2 with the feedforward gain, K_r

Zero Steady State for each of the outputs y_1, y_2 & y_3

Zero steady-state error for each of the outputs, y_1, y_2 and y_3 cannot be achieved since the chemical reactor had two inputs, U_1 and U_2 but three outputs. Each output did not have a reference input hence, zero steady-state was not achieved.

The reference outputs, $y_1(Reation\ temperature)$, $y_2\ (Reactor\ pressure)$ and $y_3\ (Cooling\ jacket\ temperature)$ did not perfectly track the unit step inputs at steady state as indicated by Figure 9. The output signal did not perfectly track the reference steady-state signal of one. This was because a slightly different system matrix $(A_o, B_o\ \&\ C_o)$ were used to implement the chemical reactor into MATLAB Simulink while $A, B, \&, D$ were used to calculate feedforward gain, K_r and feedback control gain matrix, K.

A different set of desired closed-loop Eigenvalues

$$Eigenvalue_{New} = \begin{bmatrix} -3 + 0.8i \\ -3 - 0.8i \\ -5 \\ -8 \end{bmatrix} \quad Eq.(18)$$

$$K_{new} = \begin{bmatrix} 0.0157 & 0.804 & 0.25 & 0.297 \\ -1.88 & -0.334 & -0.682 & 0.296 \end{bmatrix} \quad Eq.(19)$$

$$K_{r_{new}} = \begin{bmatrix} 0.0177 & 3.3603 \\ -1.061 & -0.663 \end{bmatrix} \quad Eq.(20)$$

Desired closed-loop eigenvalues shown in Equation 18 were implemented in MATLAB code to obtain the new control gain matrix, K_{new} and the feedforward gain, $K_{r_{new}}$. While selecting the new eigenvalue, it was ensured that it belonged further in the left-hand side of the complex plane than the previous eigenvalues (Equation 15). This would allow observing the effect of the location of the dominant desired eigenvalues on the overshoot and the settling time of the system. Relevant matrices were then transferred to MATLAB Simulink and the step response of the system shown in Figure 10 was extracted.

Step Response of individual outputs

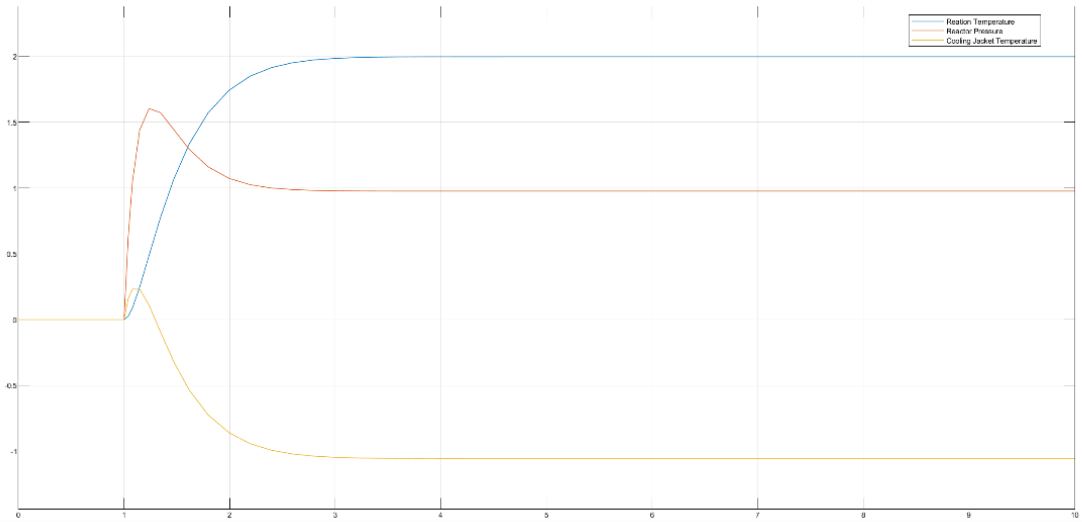

Figure 10 – Step response of the feedforward control system with the new desired eigenvalues

State Response

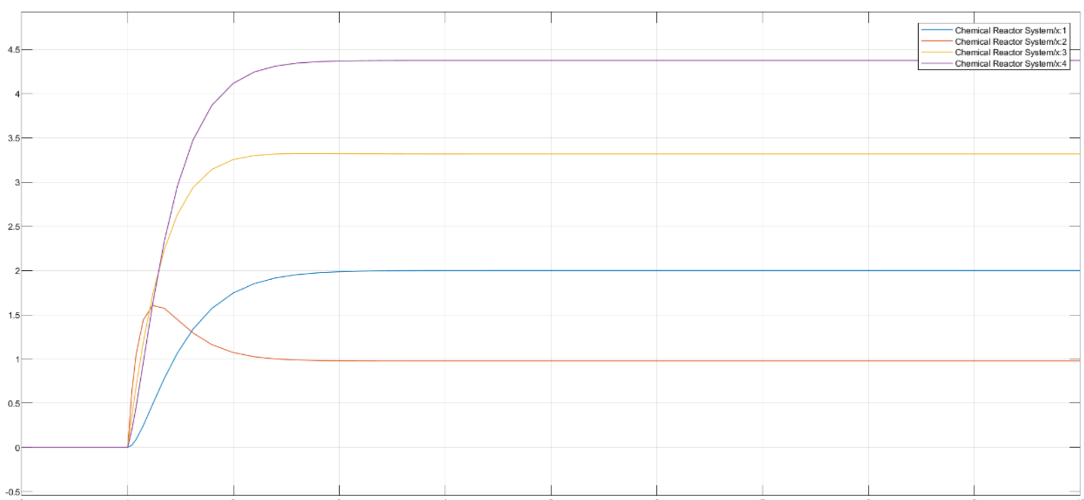

Figure 11 - Closed Loop (State Vector) response of the feedforward control system with new desired λ

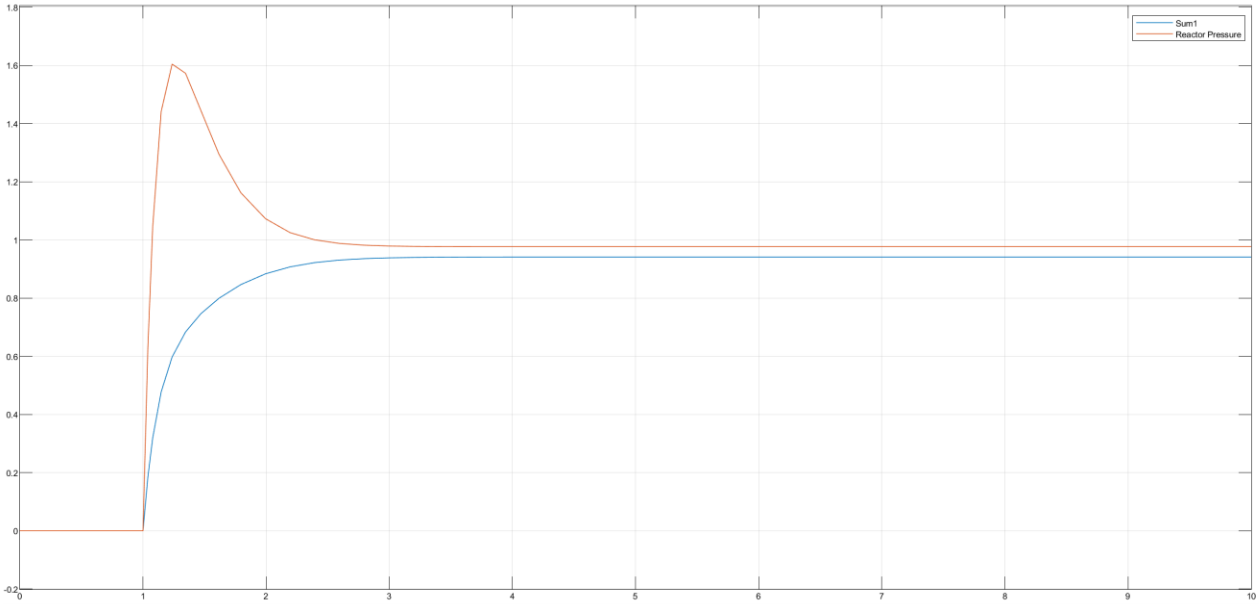

Figure 12 - signal $y_1 + y_3$ & $y_2$of system with new desired eigenvalue

Equation 21 and Equation 22 describes the two dominant poles of the previous and the new system with eigenvalues shown in Equation 15 and Equation 18, respectively.

$$Dominant\ Poles_{Previous} = \begin{bmatrix} -1 + 0.4i \\ -1 - 0.4i \end{bmatrix} \qquad Eq. (21)$$

$$Dominant\ Poles_{New} = \begin{bmatrix} -3 + 0.8i \\ -3 - 0.8i \end{bmatrix} \qquad Eq. (22)$$

$$\lambda = -\zeta\omega_0 \pm \sqrt{{\omega_0}^2(\zeta^2 - 1)} \qquad Eq. (23)$$

Assumption: Since the chemical reactor is a higher-order system, it was assumed that the two poles closest to the imaginary axis dominated the system.

Figure 9 shows the step response of the system with the previous eigenvalues shown in Equation 15 while Figure 11 shows the step response of the system with the new desired eigenvalues shown in Equation 18. The dominant poles of the system with the new eigenvalues were further to the left in the complex-plane than the previous system. Hence, as the dominant poles moved further away from the imaginary axis on the complex plane, the overshoot of the closed-loop system increased, and the settling time decreased as supported by Figure 10.

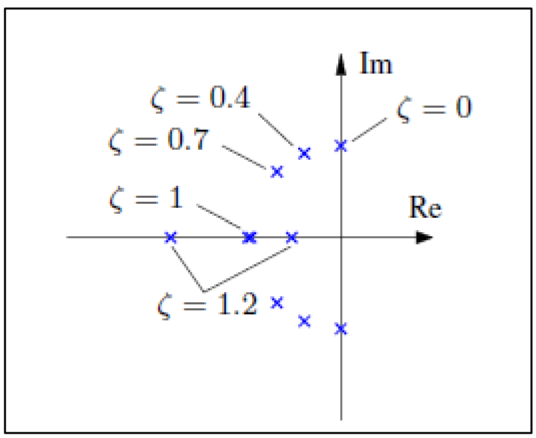

Figure 13 - Relationship between damping ratio, natural frequency, and the dominant poles of the system (Murray).

Furthermore, the dominant poles of both systems were complex numbers which suggest that the damping ratio, $(\zeta < 1)$ was less than one. Therefore, applying the relationship between the damping ratio and the dominant poles shown in Equation 23, it was deduced that increasing the damping ratio reduced the overshoot but increased the settling time as supported by Figure 9 and Figure 11. Moreover, (Murray) suggest that $e^{\lambda t}$ describes the decay/growth of the system while the rise time and settling time was determined by the natural frequency, ω_0. This was further supported by the step response obtained via Simulink of systems with different eigenvalues.

Session 2 Part II (Week 6)

Infinite Horizon Linear Quadratic Optimal Regular

Matlab function, lqr was used to determine the state feedback gain, K which was processed further using Equation 25 to deduce the eigenvalues of the closed-loop system shown in Equation 26.

$$K = \begin{bmatrix} 0.702 & 0.636 & 0.496 & -0.1539 \\ -2.378 & -0.1532 & -1.685 & 1.08 \end{bmatrix} \qquad Eq.(24)$$

$$A_{CL} = (A - BK)\,\lambda \qquad Eq.(25)$$

$$\lambda = \begin{bmatrix} -1.6 \\ -2.77 \\ -8.96 \\ -7.59 \end{bmatrix} \qquad Eq.(26)$$

Feedforward gain, k_r

Feedforward gain, k_r was deduced and implemented in Simulink in the same configuration shown in Figure 7 to obtain Figure 14 and Figure 15

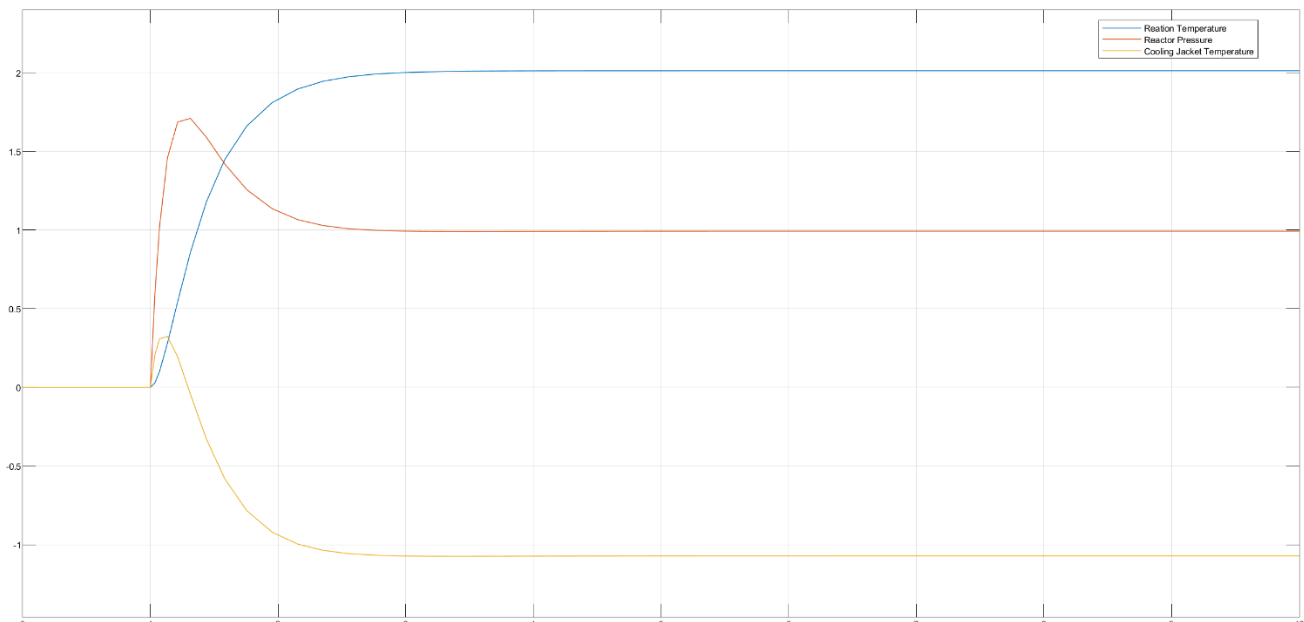

Figure 14 - Step response of the feedforward system designed using linear quadratic optimal regulator

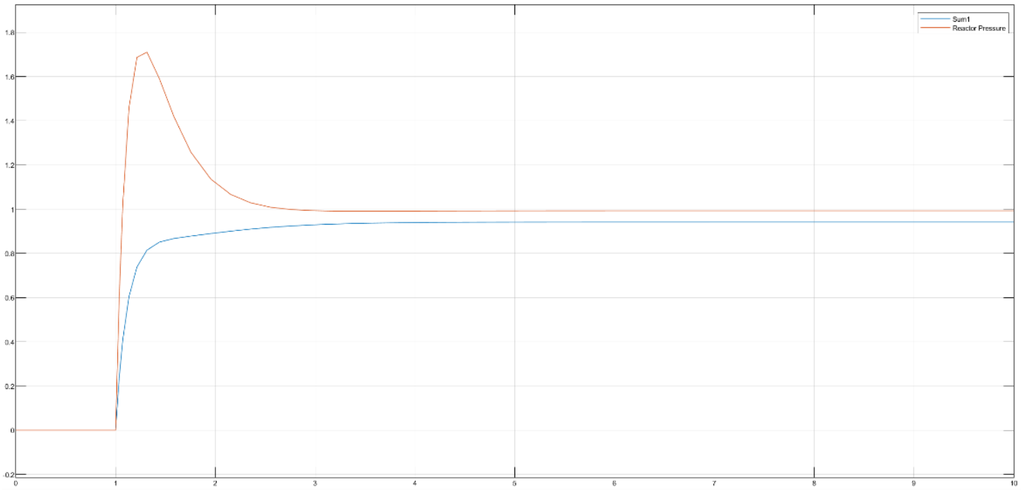

Figure 15 - signal $y_1 + y_3$ & $y_2$of system with optimal controller

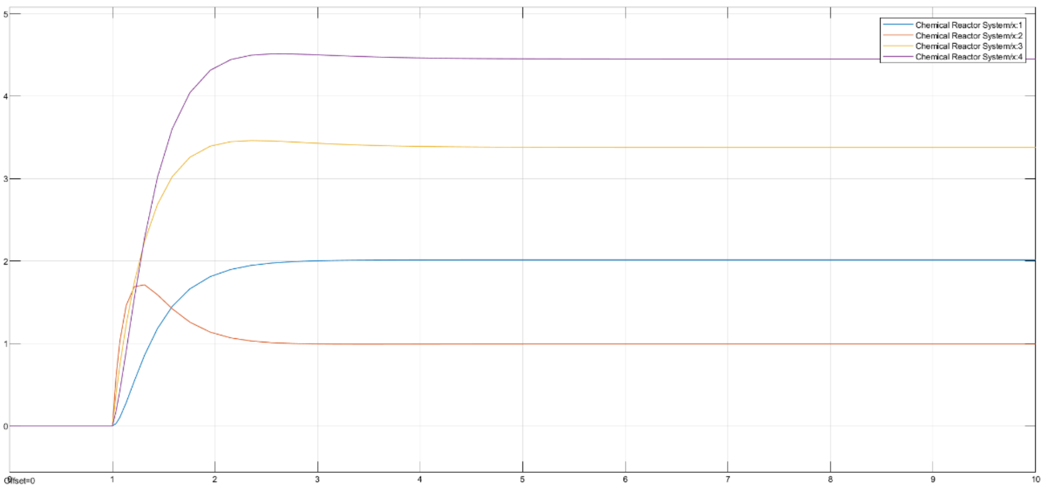

Figure 16 - State Response with optimal controller

The step response of the control system designed with the linear-quadratic optimal regulator is shown in Figure 14. Figure 14 suggest that the settling time and the overshoot of the system with the optimal regulator are the same as the system designed using eigenvalues(new) shown in Equation 18. However, when comparing with the initial feedforward control system designed using eigenvalues shown in Equation 15, the optimal control system has much smaller overshoot and the settling time as evident from Figure 14 and Figure 8.

The system designed via quadratic optimal regulator had a far better steady-state response than the system designed via pole placement method. This was because the system with the optimal regulator was able to better track the unit step response as evident from Figure 15 when compared with the system via pole placement method, Figure 9.

State Augmentation Method

$$\lambda_{Desired} = \begin{bmatrix} -1+0.4i \\ -1-0.4i \\ -2+0.0i \\ -3+0.0i \\ -4+0.0i \\ -5+0.0i \end{bmatrix} \qquad Eq. (27)$$

$$\lambda_{Dominant} = \begin{bmatrix} -1+0.4i \\ -1-0.4i \end{bmatrix} \qquad Eq. (28)$$

Augmentation method was used to determine the feedback matrix, K shown in Equation 29 and the integral control matrix, K_i shown in Equation 30. The augmentation procedure and the matrices are shown in the Matlab code (Appendix)

$$K_{AUG} = \begin{bmatrix} 0.048 & 1.005 & 0.319 & 0.740 \\ -2.85 & -0.063 & -2.00 & 1.43 \end{bmatrix} \quad K_r = \begin{bmatrix} -0.0147 & 1.3812 \\ -0.6787 & 0.4664 \end{bmatrix} \qquad Eq. (29)$$

$$K_i = \begin{bmatrix} -0.0748 & 2.755 \\ -0.475 & 0.351 \end{bmatrix} \qquad Eq. (30)$$

$$A_{CL} = \begin{bmatrix} -1.09 \\ -3.4+0.14i \\ -3.4-0.14 \\ -8.11 \end{bmatrix} \qquad Eq. (31)$$

Integral action state feedback controller for the 3-output case cannot be designed since the state, $\dot{x}_1$ does not pass through the state vector, B as supported by Figure 1. Hence, the state is not reachable.

$$W_r = \begin{pmatrix} B & AB & \dots & A^n B \\ 0 & CB & \dots & CA^{n-1}B \end{pmatrix}.$$

Figure 17 - Reachability of the augmented system (Murray)

Furthermore, Figure 17 describes the reachability condition of the augmented system. Therefore, implementing this in MATLAB, it was deduced that the rank of the matrix with three outputs was six and not seven. Therefore, one of the outputs was not reachable. Although the overall system was reachable since the rank of matrix A was six and $rank\left(ctrb(A_{aug}, B_{aug})\right) = 6$. Hence, the overall augmented system was reachable but cannot design the integral action for the 3-output case.

Simulink with Integral Action

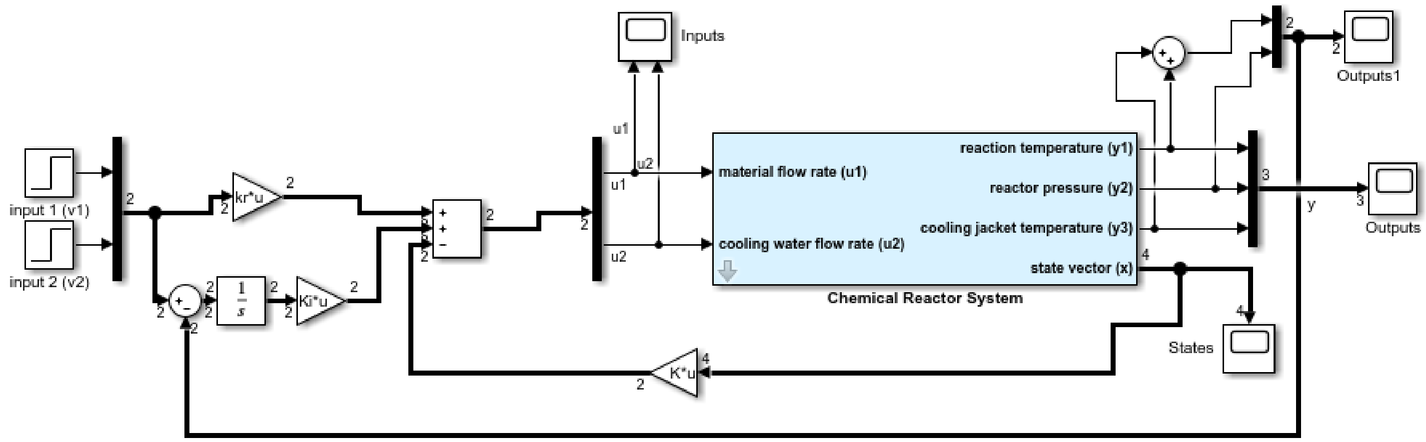

Figure 18 - Simulink with Integral Action

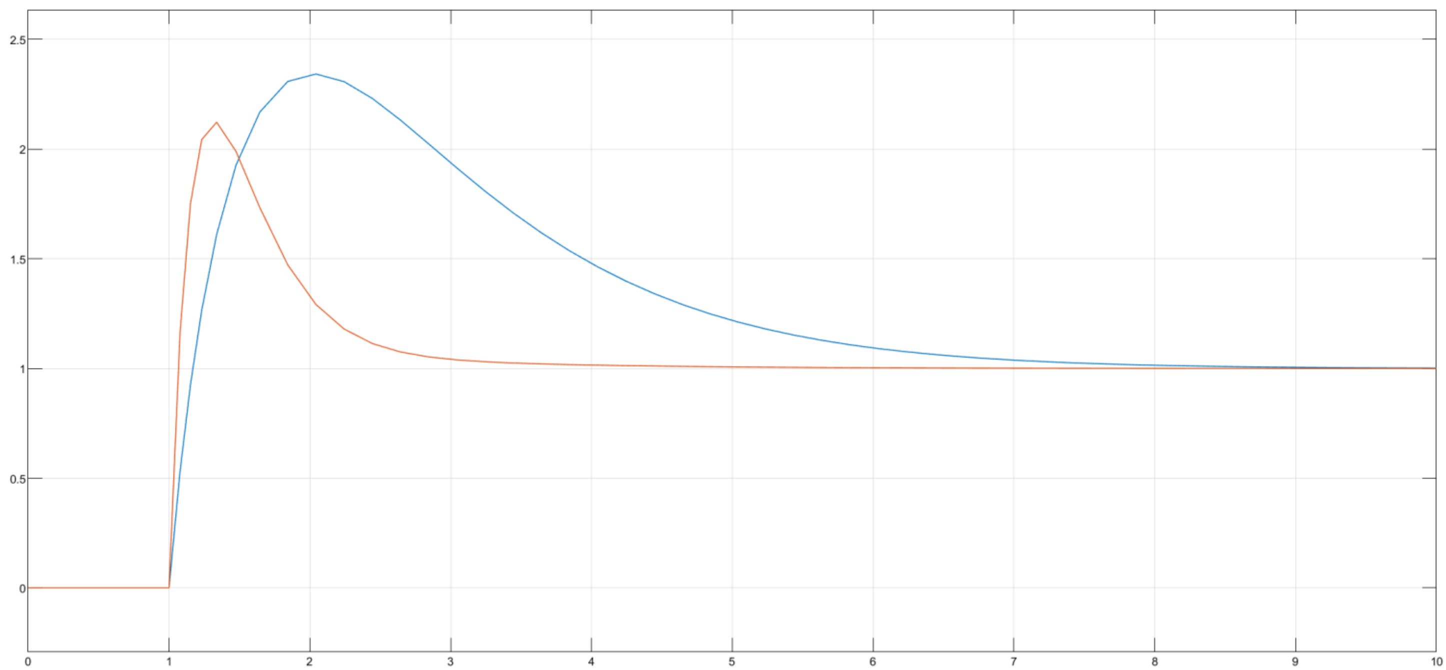

Figure 19 - Step Response of the System

Figure 19 shows the steady-state response of the system incorporated with integral action, K_i obtained via simulating Figure 18. The response suggests that the control system with integral action was able to track the unit step response reasonably well when compared with systems designed previously using only the feedforward, k_r.

A different set of desired Eigenvalues

$$\lambda_{Desired} = \begin{bmatrix} -3+0.8i \\ -3-0.8i \\ -5 \\ -8 \\ -9 \\ -10 \end{bmatrix} \qquad Eq.(32)$$

$$\lambda_{Dominant} = \begin{bmatrix} -3+0.8i \\ -3-0.8i \end{bmatrix} \qquad Eq.(33)$$

$$A_{CL} = \begin{bmatrix} 3.52 \\ -4.93 \\ -11.68 \\ -24.91 \end{bmatrix} \qquad Eq.(34)$$

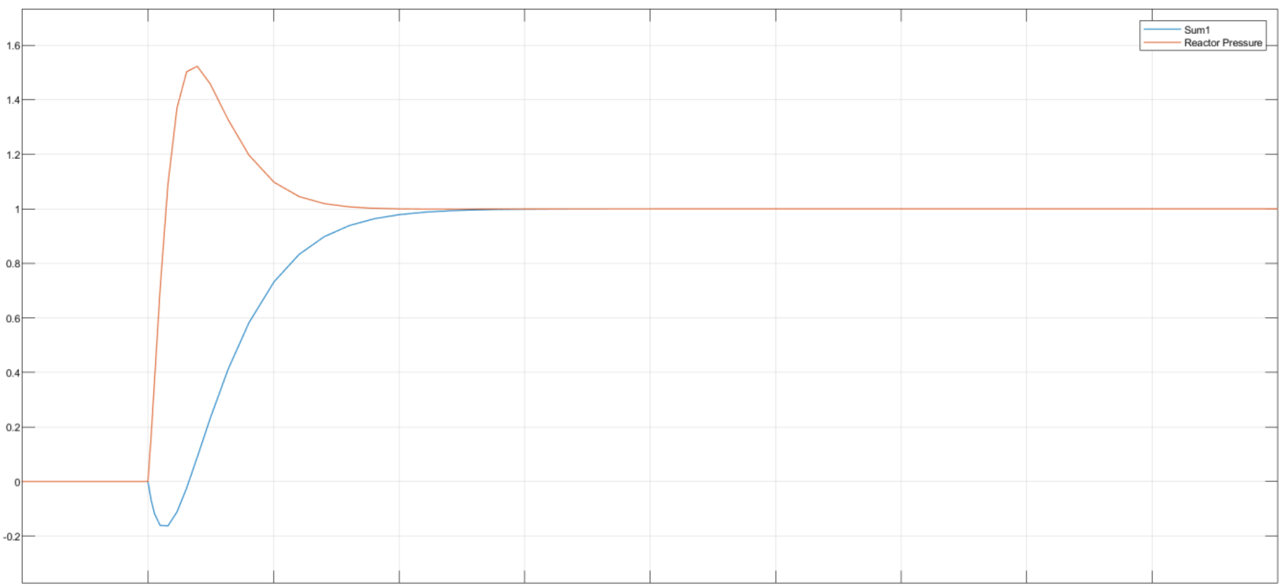

Figure 20 - Unit step response of the control system (New eigenvalues) with integral action

The newly selected eigenvalues are shown in Equation 32 which were processed further to obtain the steady-state response shown in Figure 20. The new eigenvalues were located further away from the imaginary axis of the complex plane than the previous one shown in Equation 27 which were used to obtain Figure 19. Unit step responses of both systems (Figure 19 and Figure 20) were evaluated to determine the trade-off between pole positions and response speed. It was concluded that the further away the pole position from the imaginary axis of the complex plane, the quicker the response speed of the control system with the integral action as supported by Figure 20.

Response to a ramp/sinusoid input

Unit Ramp/sinusoid Response of the system with previous λ

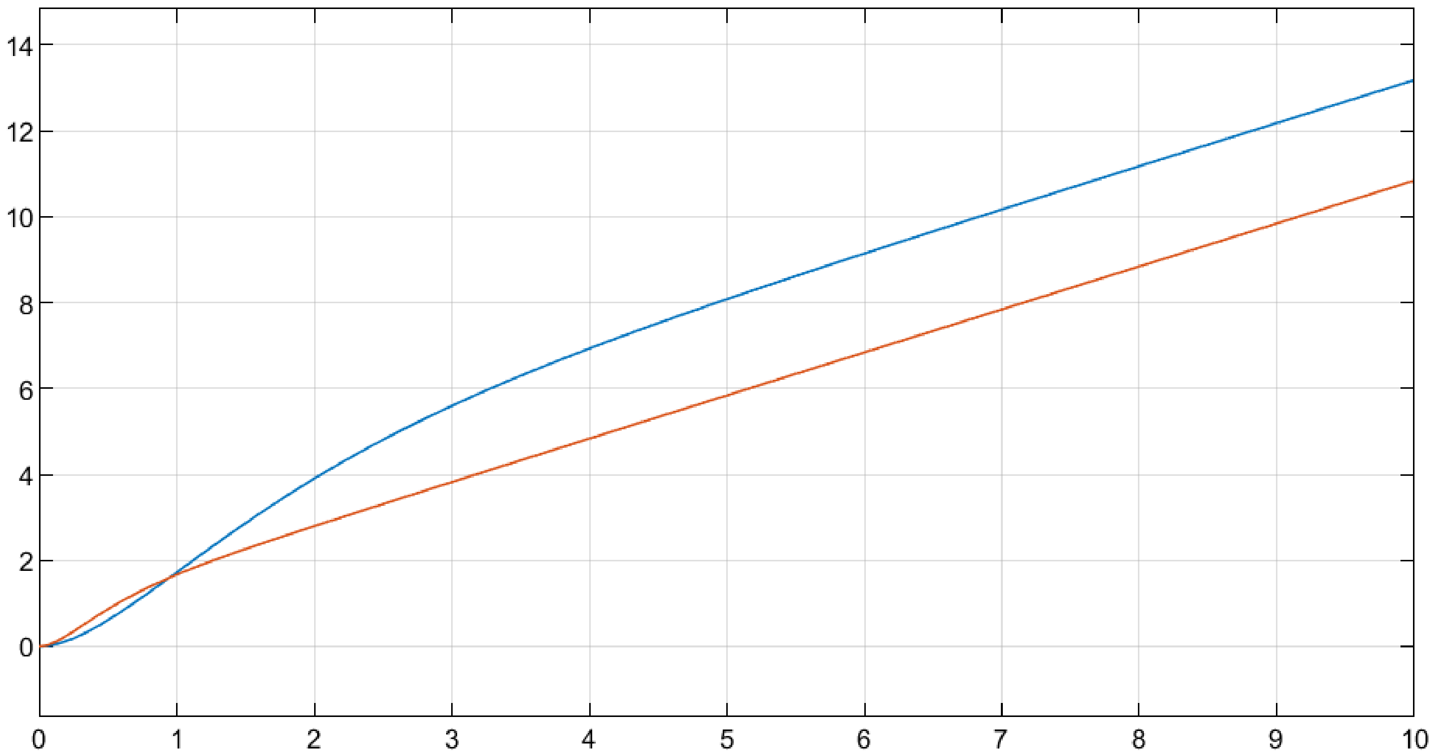

Figure 21 - Unit ramp response of the control system with previous eigenvalues shown in Equation 27

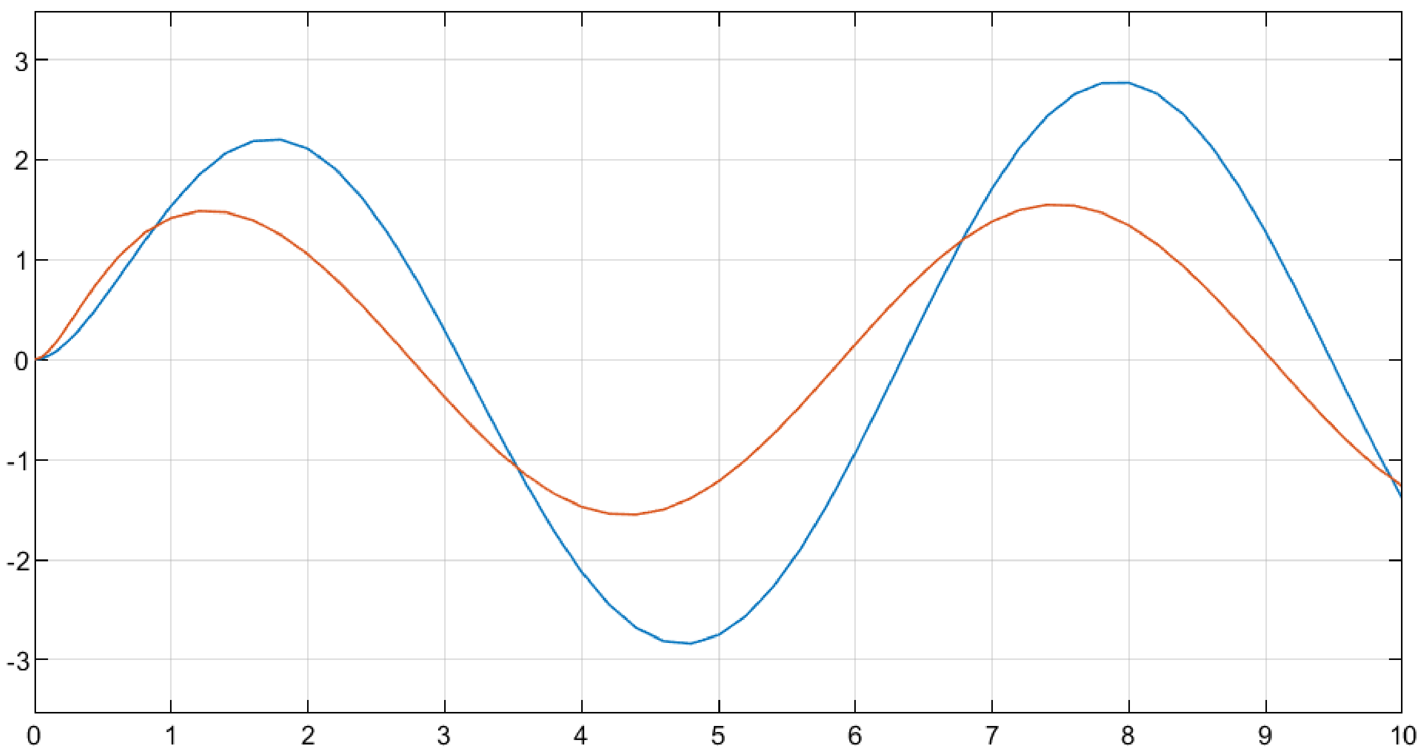

Figure 22 - Unit sinusoid response of the control system with previous eigenvalues shown in Equation 27

Zero tracking error to a unit ramp and sinusoid input was not achieved as displayed in Figure 21 and Figure 22, respectively. This was due to the use of single integrator, implementing a double integrator setup would enable to get closer if not zero tracking error.

Session 3 Part I (Week 7)

Observer Design

$$\lambda_{Desired} = \begin{bmatrix} -2+0.4i \\ -2-0.4i \\ -4 \\ -6 \end{bmatrix} \qquad Eq.(35)$$

$$Observer\ Gain\ Matrix, L = \begin{bmatrix} 7.38 & -1.43 & 6.75 \\ -0.85 & 0.16 & 0.56 \\ 4.075 & 5.36 & -4.34 \\ 3.22 & 4.12 & 0.90 \end{bmatrix} \qquad Eq.(36)$$

NOTE: Feedforward gain, K_r and integral action, K_i shown in Equation 29 and Equation 30 were used, respectively, while implementing the full-state observer and the control system in MATLAB Simulink.

Simulink of a full-state observer

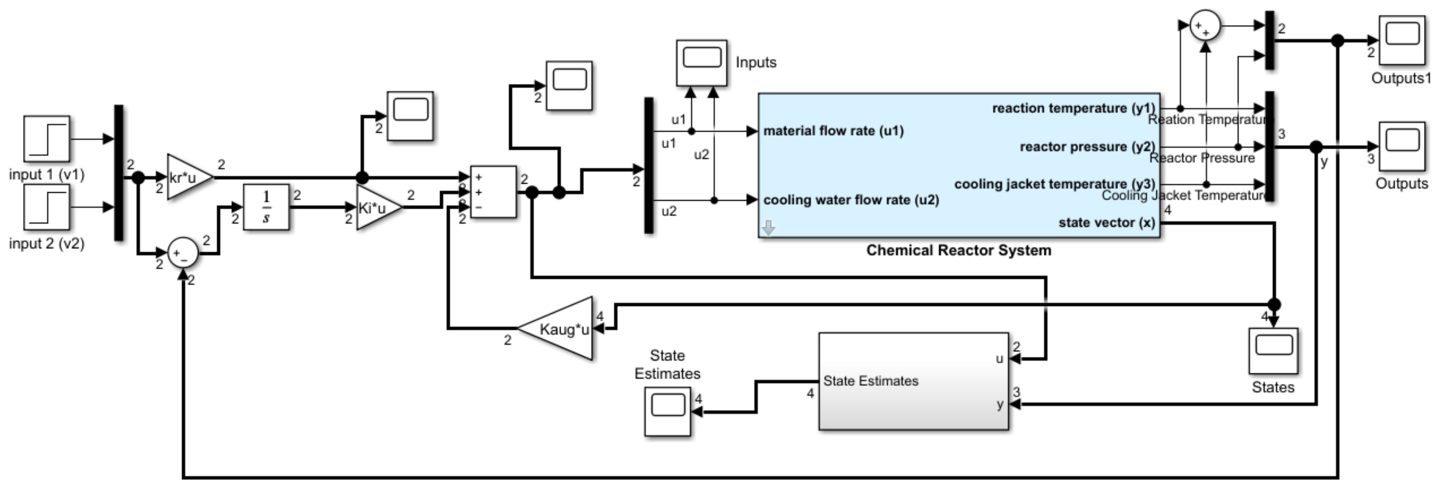

Figure 23 - Simulink model of the system with full-state observer

Figure 23 was used to extract relevant data to analyze the estimation error of the system and the relationship between the eigenvalue locations and the magnitude of observer gains.

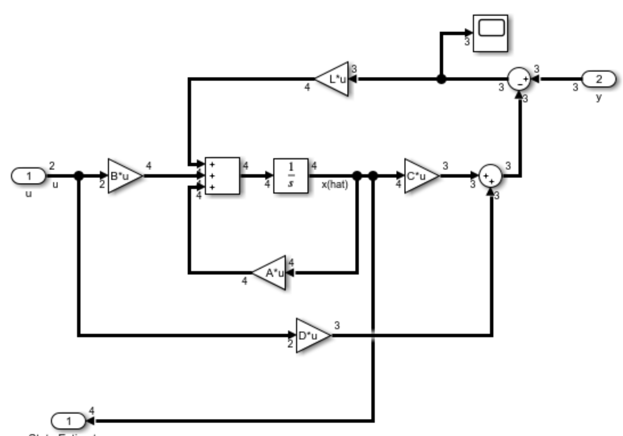

Figure 24 - Ful-state observer

Error Dynamics of the observer

NOTE: $[1 - 1\ 0.5 - 0.5]'$ was assigned as the initial state estimate

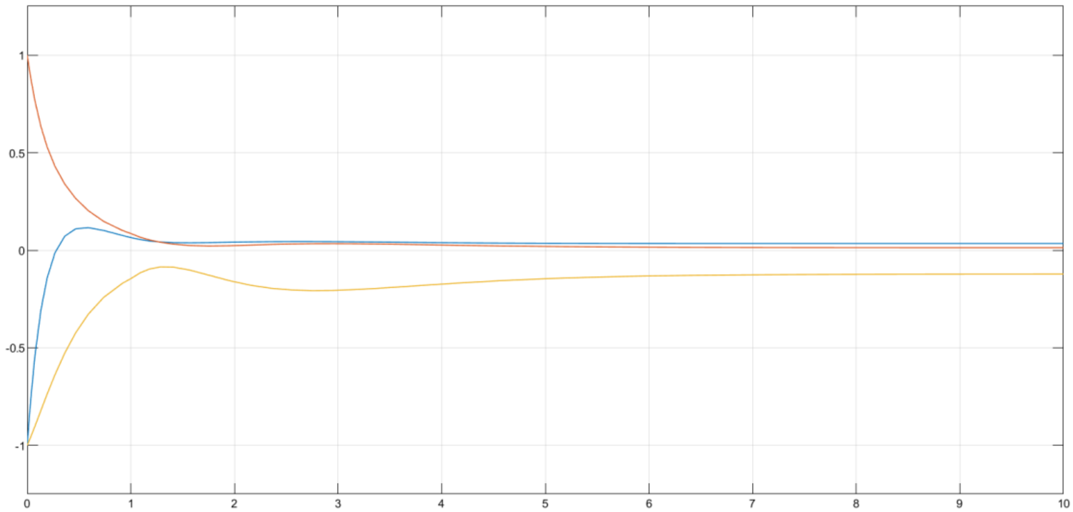

Figure 25 - Output error, $\tilde{y}$ of the observer

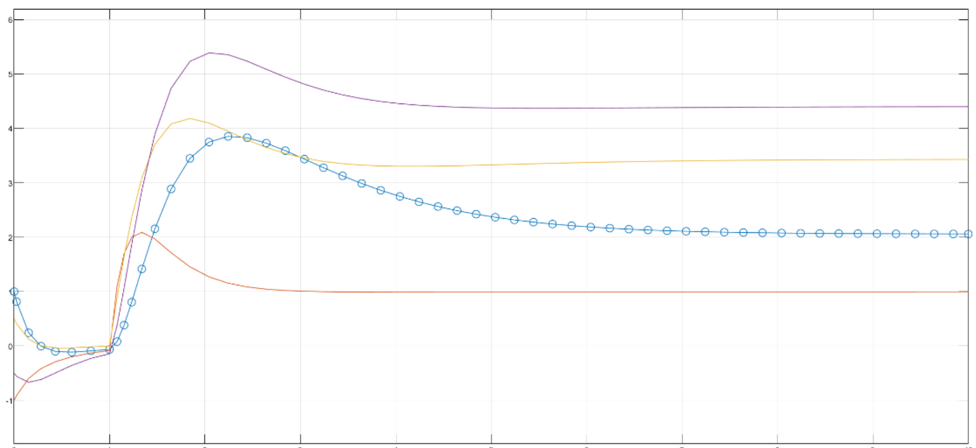

Figure 26 - State estimates of the system

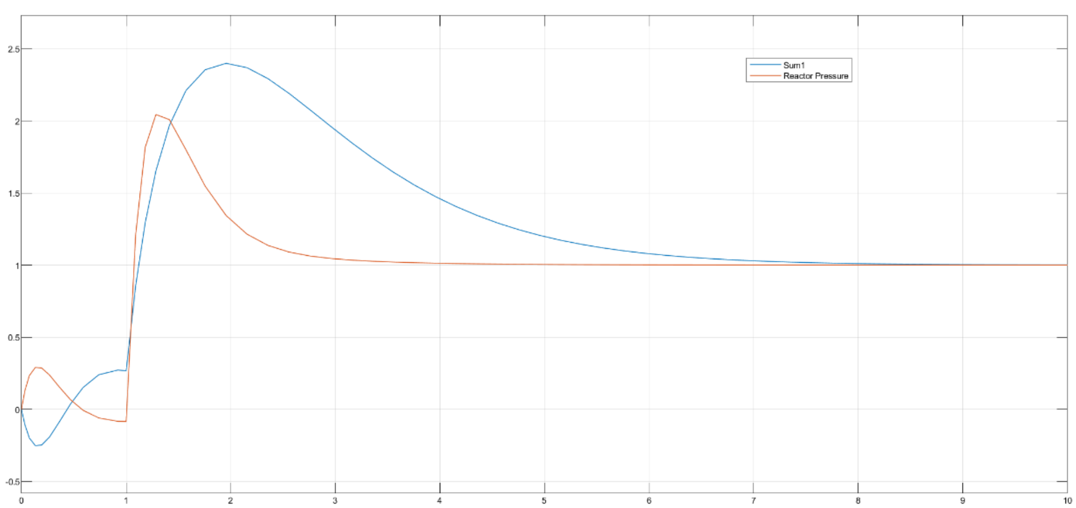

Figure 27 - Steady State response of the system with the observer

The observer shown in Figure 24 was implemented in Figure 23 to obtain the output error, $\tilde{y}$ of the observer shown in Figure 25. The estimation error of the observer did not decay to zero as evident from Figure 25 and Figure 26 since the system matrices used to determine the observer gain matrix, L, were not the same as the ones used in the Simulink model. Although, the system with the observer implemented had a zero steady-state error as shown in Figure 27. This suggests that the system was able to effectively track the unit step response.

A different set of desired eigenvalues

A different set of desired eigenvalues described in Equation 37 were considered to investigate the correlation between the location of the eigenvalues and the magnitude of the observer gain, L. While selecting the new desired eigenvalues, it was ensured that it belonged further left from the imaginary axis on the complex plane when compared with the previous eigenvalues (Equation 35)

$$\lambda_{Desired} = \begin{bmatrix} -3+0.8i \\ -3-0.8i \\ -6 \\ -8 \end{bmatrix} \qquad Eq.(37)$$

$$New\ Observer\ Gain\ Matrix\ L_{new} = \begin{bmatrix} 7.23 & -1.04 & 5.77 \\ -1.53 & 2.14 & 0.21 \\ 4.24 & 7.70 & -3.03 \\ 3.89 & 6.50 & -1.96 \end{bmatrix} \qquad Eq.(38)$$

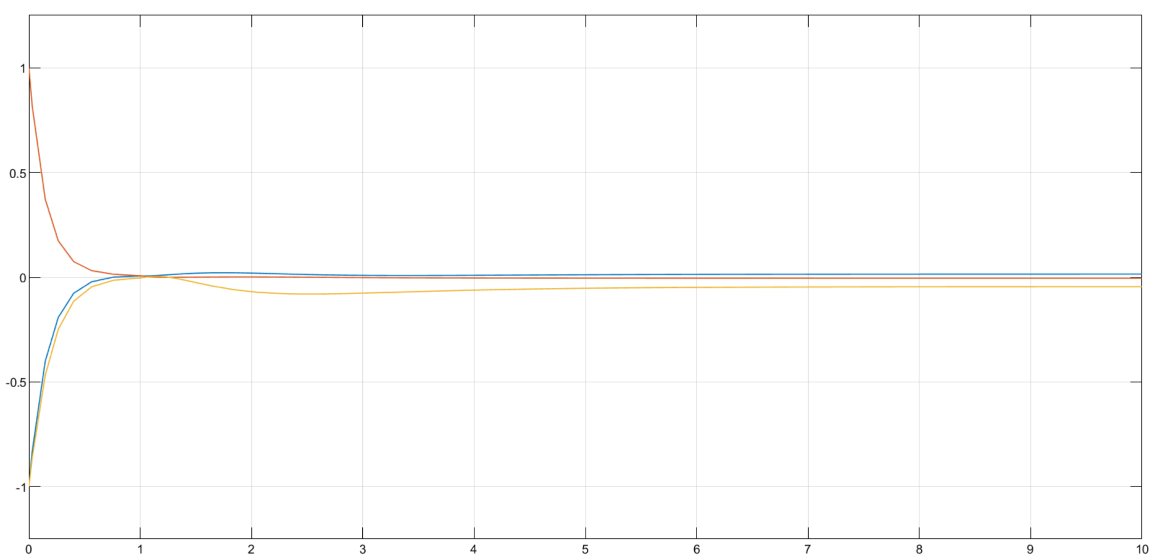

Figure 28 - Output error, $\tilde{y}$ of the new observer

The magnitude of both gain matrix; previous and new observer (Equation 36 and 38) was calculated via MATLAB $norm$ function and is shown in Equation 39 and Equation 40, respectively.

$$Magnitude\ of\ Gain\ Matrix, L_{Previous\ Observer} = 10 \qquad Eq.(39)$$

$$Magnitude\ of\ Gain\ Matrix, L_{New\ Observer} = 12 \qquad Eq.(40)$$

The magnitude of the gain matrix, L of the new observer was 12 and the magnitude of the gain matrix, L of the previous observer was 10. While the eigenvalue location of the new observer was further away from the imaginary axis in the complex plane when compared with the previous observer. Furthermore, the output error, $\tilde{y}$ of the new observer gain matrix was significantly less than that of the previous observer as supported by Figure 28. Hence, it was deduced that the further away the eigenvalue location from the imaginary axis in the complex plane, the higher the magnitude of the observer gain matrix, L and lower the output error, $\tilde{y}$.

Modified Simulink Compensator (Controller + Estimator) model

The Simulink model was modified by feeding the state estimates to the feedback controller as shown in Figure 29

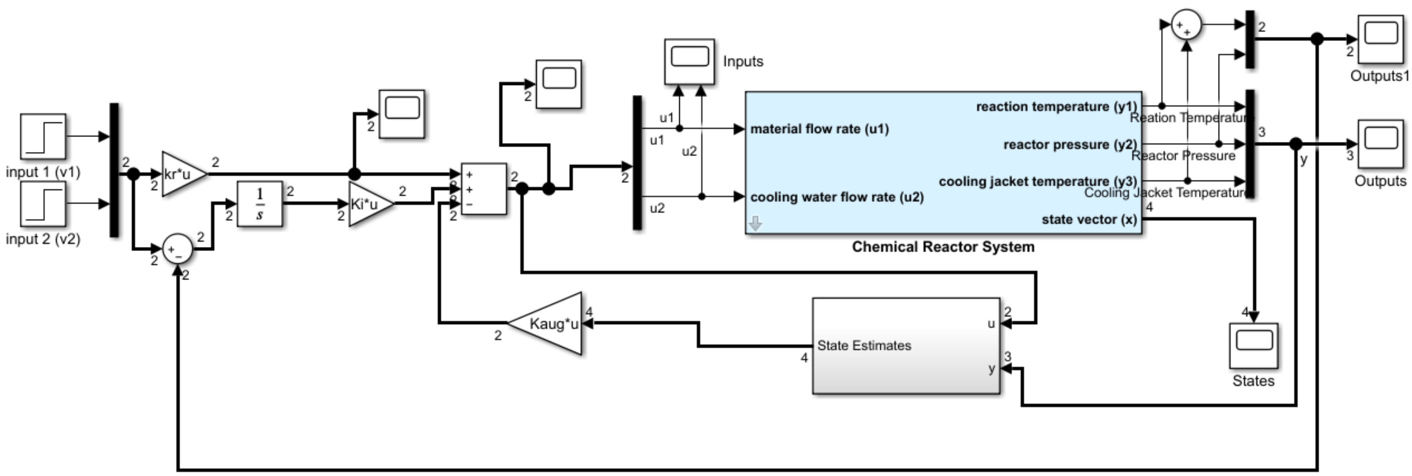

Figure 29 – Modified Compensated Simulink Model

The closed loop eigenvalues of the compensated system were determined via the MATLAB function, eig. The controller designed using the augmented method and the desired eigenvalues shown in Equation 31 was implemented in the compensated system (Figure 29).

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\tilde{x}}(t) \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix} \Rightarrow \dot{x}_{cl} = A_{cl} x_{cl}$$

Figure 30 - Closed-loop dyanmics of the compensated / modified system (Murray)

$$\lambda_{State\ Feedback} = eig(A - BK) = \begin{bmatrix} -3.40 + 0.13i \\ -3.40 - 0.14i \\ -1.09 \\ -8.11 \end{bmatrix} \qquad Eq.(41)$$

$$\lambda_{Observer} = eig(A - LC) = \begin{bmatrix} -2 + 0.4i \\ -2 + 0.4i \\ -4 \\ -6 \end{bmatrix} \qquad Eq.(42)$$

NOTE: Eigenvalues of integral action was ignored

Step response of the system with zero and non-zero initial error estimates

Zero initial error estimates

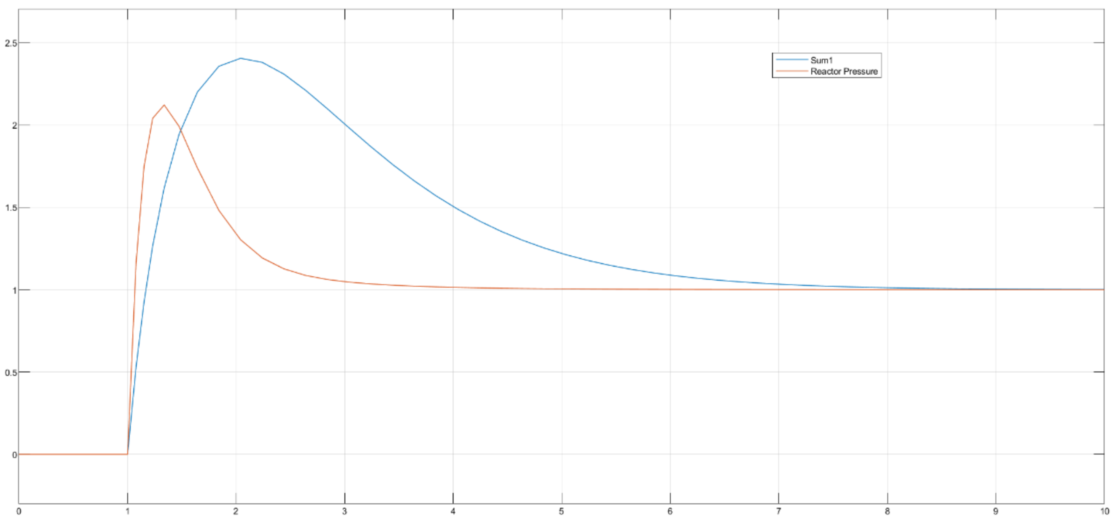

Figure 31 - Step response of the system with zero initial error estimate

Non – Zero initial error estimates

Initial state estimate – [1 -1 0.5 -0.5]'

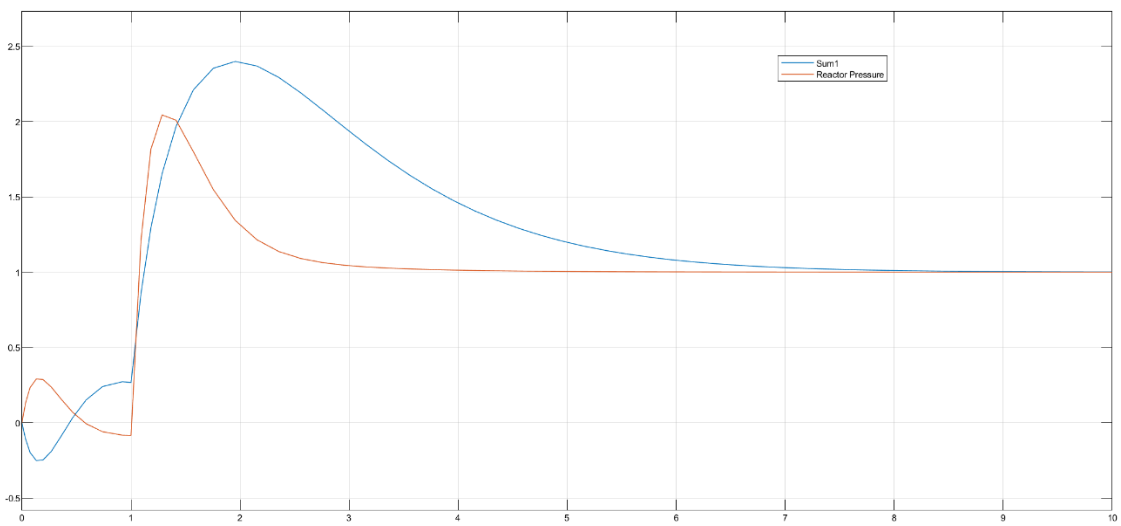

Figure 32 - Step response of the system with non-zero initial error estimate

The step response of the system with the zero initial error estimate is shown in Figure 31 while the step response of the system with the initial state error of $[1 - 1\ 0.5 - 0.5]'$ is shown in Figure 32. It was concluded that the initial state estimates only affected the response of the system before the unit step input kicks in. Once the unit step input was infused in the system, the step response including the overshoot and the settling time remained the same. In other words, the initial state estimate did not affect the overall system performance as the rise time and the overshoot remained the same. However, initial movement in the response was noticed for non-zero error estimate due to the system trying to adjust to the initial state of the system.

Compare the closed-loop system performance

In this sub-section, the effect on the shape of the system response by the observer eigenvalues being further away from the eigenvalues of the system will be evaluated.

A new observer was designed with the poles (Equation 43) being 100 times further away to the left on the complex plane from the previous observer's eigenvalues shown in Equation 35.

$$\lambda_{Desired} = \begin{bmatrix} -200 + 40i \\ -200 - 40i \\ -400 \\ -600 \end{bmatrix} \qquad Eq.(43)$$

The eigenvalues described in Equation 43 was processed further and was implemented in MATLAB Simulink to obtain the system response shown in Figure 33

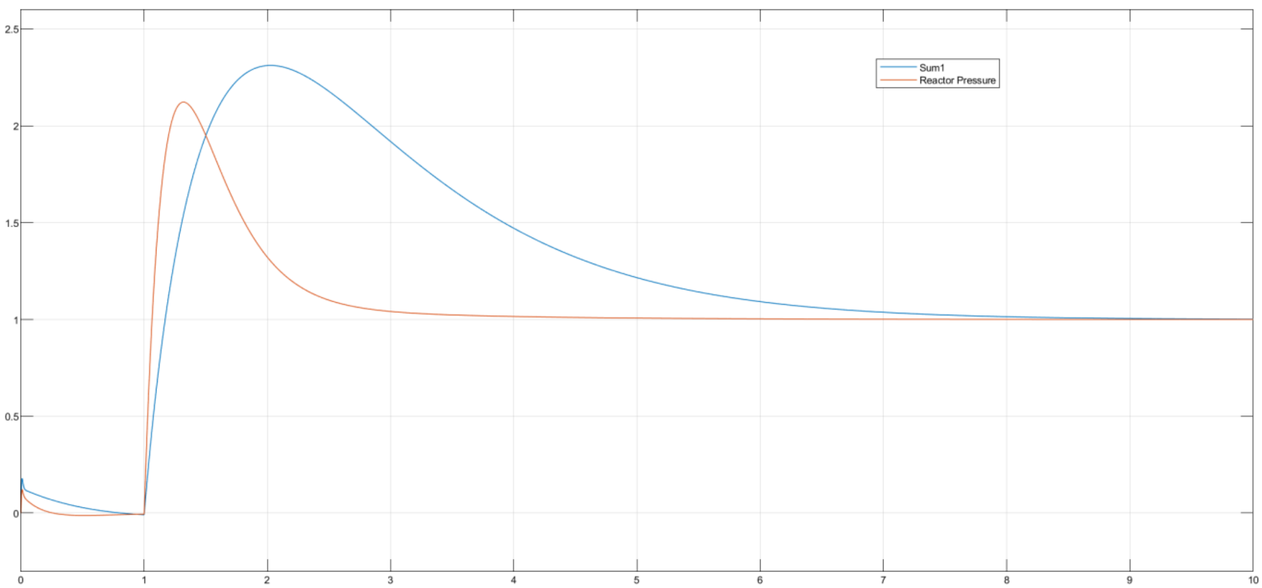

Figure 33 - System response with the observer eigenvalues being 100 times further from the previous observer's eigenvalue

Figure 33 shows the system response when the poles of the observer were further away from the controller poles while Figure 27 shows the system response when observer poles were closer to the controller poles. Although the overall system response indicates similar transient performance i.e., the overshoot, settling time and the peak time of both systems, however, a smoother system response was observed when the observer poles were further away from the controller poles.

Session 3 Part II (Week 7)

Kalman Filter Design

$$R_w = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Eq.(44)$$

$$R_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Eq.(45)$$

The variance matrices are shown in Equation 44 and Equation 45 were used in conjunction with the $kalman$, MATLAB function to determine the observer eigenvalues shown in Equation 46.

$$\lambda_{Observer} = (A - LC) = \begin{bmatrix} -1.55 \\ -3.59 \\ -9.53 \\ -9.18 \end{bmatrix} \qquad Eq.(46)$$

Error dynamics of the Kalman Filter

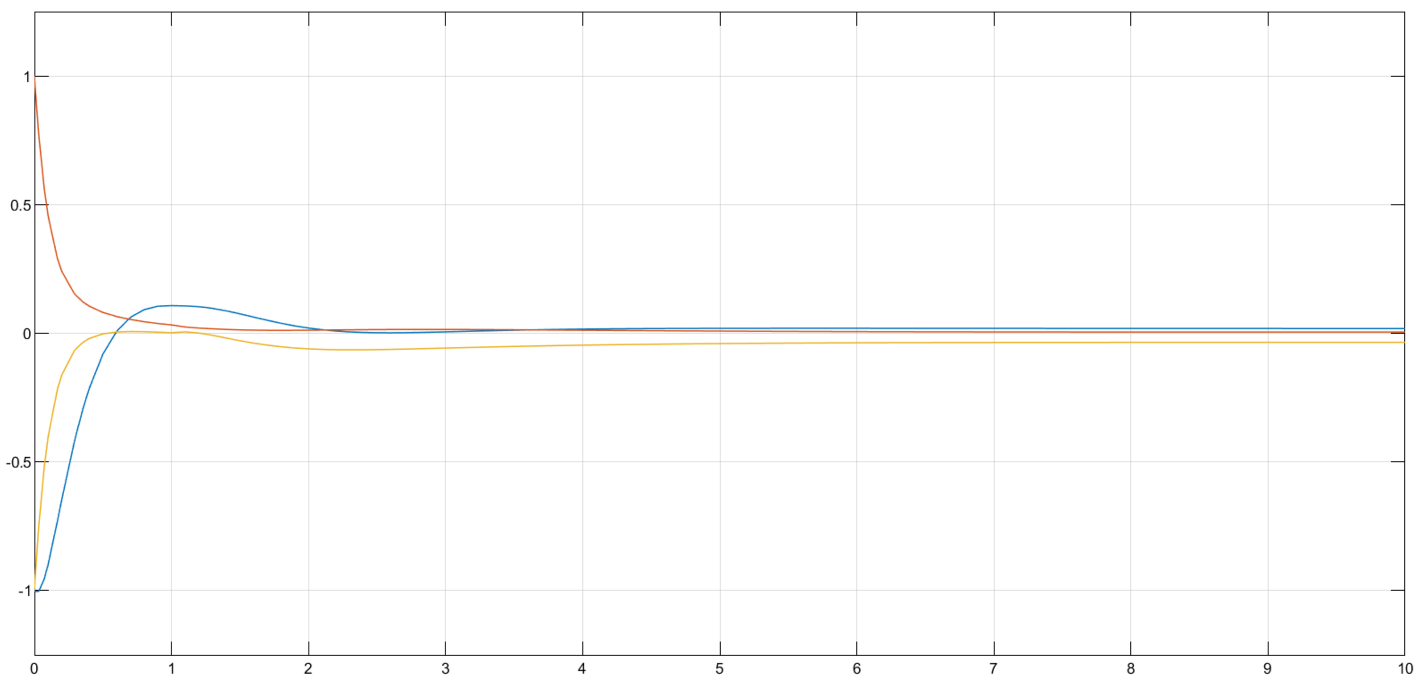

Figure 34 - Error dynamics of the (Kalman filter)

The output error, $\tilde{y}$ of the Kalman Filter is shown in Figure 34 while the error dynamics of the observer designed via pole placement method is shown in Figure 25. It was deduced after comparing the result of both the observers that the performance of the Kalman Filter was significantly superior since there was less output error, $\tilde{y}$ as supported by Figure 34.

Different design matrices & Measurement Noise

While selecting the design matrices, it was assumed that the measurement noise (wet noise) was relatively noisily when compared with the noise of the state (process noise). Therefore, it was ensured that the selected R_w was larger than the selected R_w

$$R_w = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \qquad Eq.(47)$$

$$R_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Eq.(48)$$

$$Noise\ Power - \ 1 \times 10^{-5} \qquad Eq.(49)$$

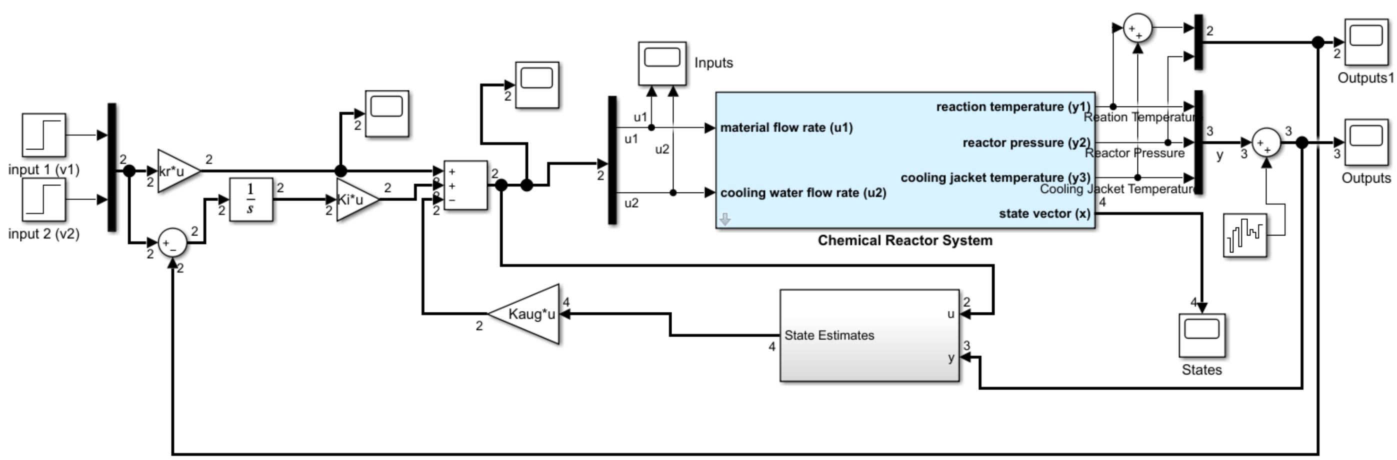

Figure 35 - Simulink model with the Kalman Filter and the Measurement Noise

Step Response of the system with the measurement noise

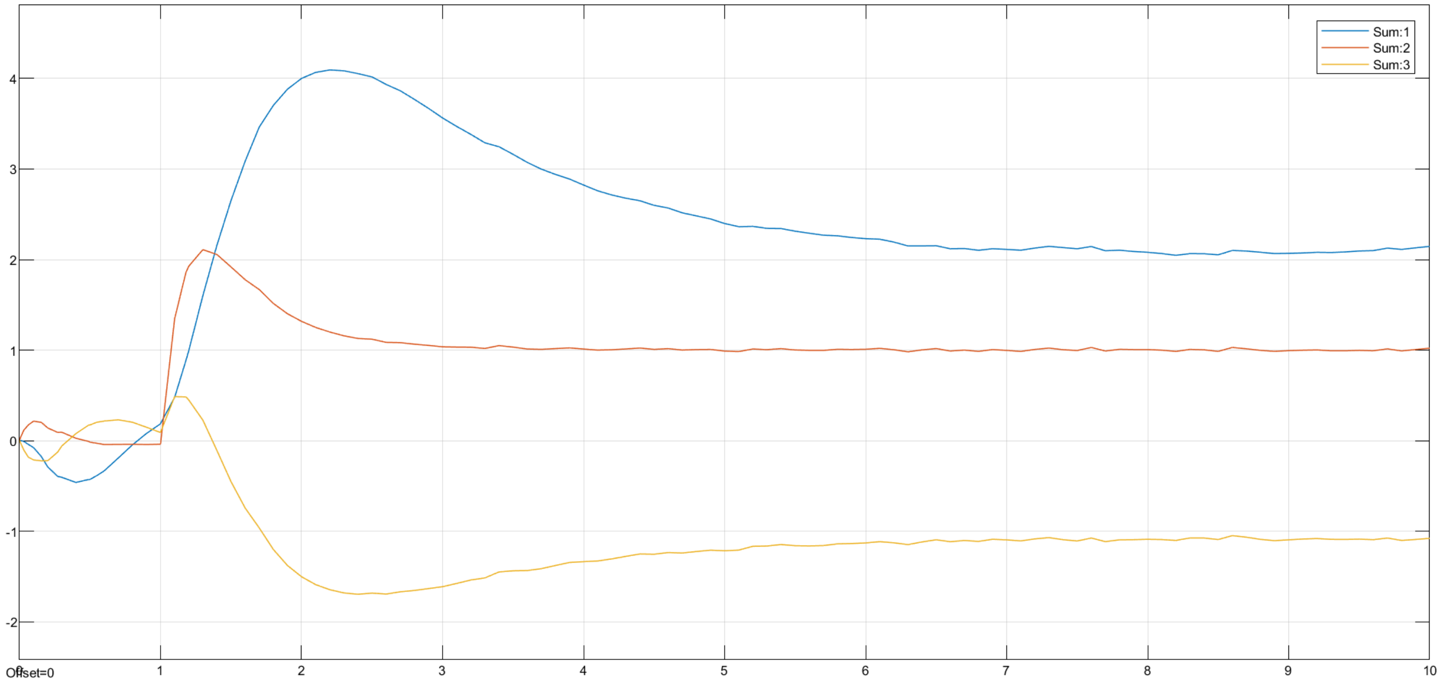

Figure 36 - Step output response of the system with measurement noise

Estimation Error Dynamics

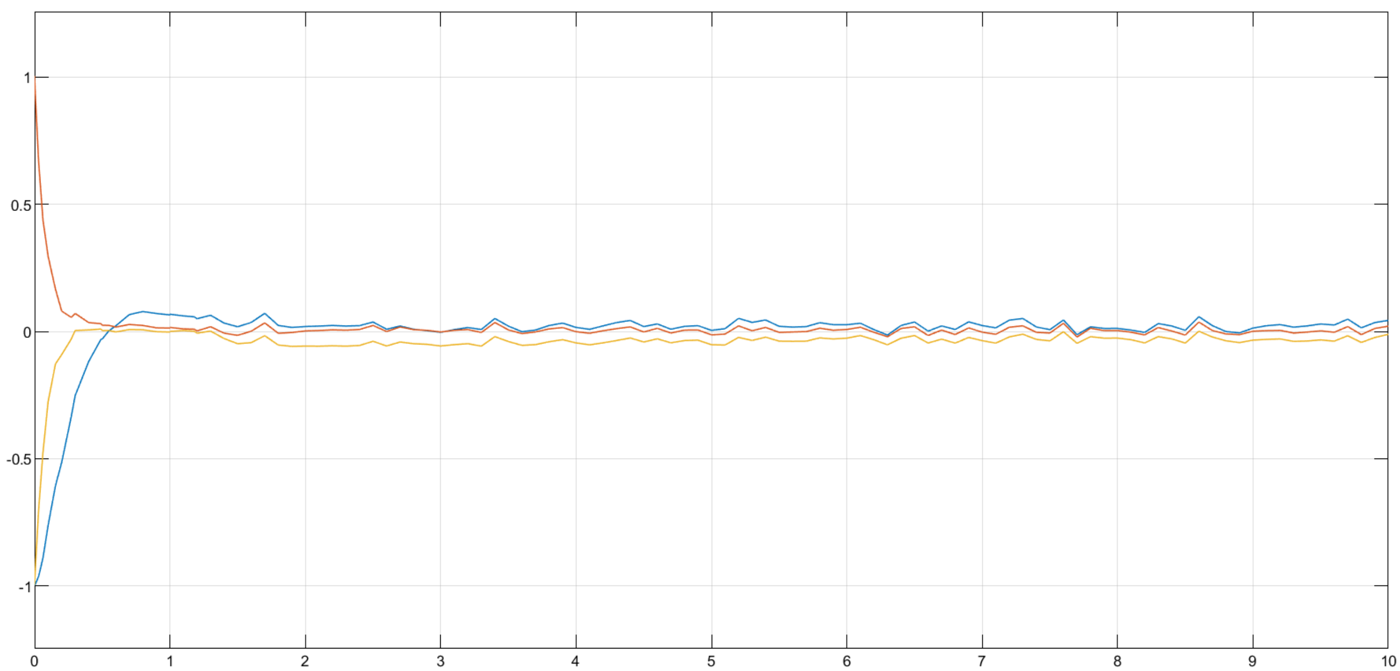

Figure 37 - Output error, $\tilde{y}$ estimation error dynamics of the Kalman Filter

The introduction of measurement noise in the system caused a slight variation/discrepancy in the step response of the system as well as in the estimation error dynamics as shown in Figure 36 and Figure 37, respectively. This discrepancy was caused due to the measurement noise in the system hence, it was deduced that as the measurement noise increases, the estimation error dynamics would be negatively affected.

Appendix A – Matlab Code

```
%%%%%%%%%%%%% ACS6129 Lab%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initialisation

% Clear all
clc;
%% Initialise Model
sslabsc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DO NOT MODIFY ANY LINES ABOVE
%
% Develop your programme below
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----------------------------------------------------------------%
%Author: Awabullah M Syed
%Student ID: 2002 000 33
```

Session 1 (Week 5) - System Analysis

```
%%
% 2) Modelled System Matrices
A = [1.4 -0.2 6.7 -5.7 ; -0.6 -4.3 0 0.7 ; 1.1 4.3 -6.7 5.9 ;
    0 4.3 1.3 -2.1]; %Dynamic Matrix
B = [ 0 0 ; 5.7 0 ; 1.1 -3.1 ; 1.1 0 ]; % Control (Input) Matrix
C = [1 0 0 0 ; 0 1 0 0 ; 0 0 1 -1]; % Sensor (Output) Matrix
D = [0 0 ; 0 0 ; 0 0]; % Direct Term for Input to Output Matrix

% Reachability and Observability properties
% -----------------Reachability----------------------------------
Wr = ctrb(A,B); %Reachability Matrix
reach_check = rank(Wr); %Must equal to 4(n=4) since Matrix A is 4x4
if reach_check == 4
    disp('The system is reachable ')
else
    disp('The system is not reachable')
end
% This system is reachable
%-----------------Observability----------------------------------
Wo = obsv(A,C); %Observability Matrix
obser_check = rank(Wo); %Must equal to 4(n = 4) since Matrix A is 4x4
if obser_check == 4
    disp('The system is observable')
else
    disp('The system is not observable')
```

```
end
%This system is observable
% 3) Modal Canonical Form representation
sys = ss(A,B,C,D);
csys = canon(sys) %Model Canonical Form
E_value = diag(csys.a); %E.Value / Lambda / mode of the system
disp('The Eigen Values of the system are '); disp(E_value)
%---------------Stability of modes------------------------------
for i= 1:4
    mode = E_value(i,1)
    if mode < 0
        disp('The mode is a stable')
    else
        disp('The mode is a unstable')
    end
end
%----------------Reachability & Observability of each modes --------------

Wo_1 = rank(obsv(csys.a,csys.c)) %Observability of the system
Wr_1 = rank(ctrb(csys.a,csys.b)) %Reachability of the system
sys_minreal = minreal(csys)
```

Session 2 Part I - State feedback and feedforward control

```
%%
% 1)
% Feedback gain K (0:0.05:8)
t_vec = (0:0.1:10)'; %Time vector for simulation
x0 = [1;0]; %Initial state vector
%--------------------Open Loop---------------------------%
sys_OL = ss(A,B,C,D); %Open Loop Tranfer Function
%---------Desired EigenValue / Poles----------------------------%
eig_cl = [-1+0.4i;-1-0.4i;-2;-3] %Desired Poles
K = place(A,B,eig_cl); %Gain Matrix K

%---------------------- 2) FeedForward gain Kr-----------------%
t_s = (0:0.1:15)';
r = ones(length(t_s),1); %Unit step function
cr_1 = [C(1,:) + C(3,:)];
cr_2 = [C(2,:)];
Cr = [cr_1 ; cr_2]; %Define Cr (Y1 + Y3 & Y2)
D_u = [0 0 ; 0 0 ]; %Define D
sys1 = ss(A,B,Cr,D_u);
kxku = [A B ; Cr D_u]\([zeros(4,2); eye(2)])
k_x = kxku(1:4,:) %extract k_x
```

```
k_u = kxku(5:6,:)
k_r = k_u + K*k_x
%-------------------------Different CL Eigenvalues / Poles--------%
eig_new = [-3+0.8i;-3-0.8i;-5;-8] %Desired Poles
K_new = place(A,B,eig_new); %Gain Matrix K
kxku_new = [A B ; Cr D_u]\([zeros(4,2); eye(2)])
k_x_new = kxku_new(1:4,:) %extract kx
k_u_new = kxku_new(5:6,:) %Extract ku
k_r_new = k_u_new + K_new*k_x  %kr
```

Session 2 Part II - Optimal control and Integral Control

```
% 1) LQR control
% Design matrices
Q_x = C' * eye(3) * C %Qx as diagonal matrix
rho = 1;
Q_u = rho * eye(2) %Qu as diagonal matrix
sys_lqr = ss(A,B,eye(4),0)
% State Feedback gain K
K_2 = lqr(sys_lqr,Q_x,Q_u)
%--------------closed loop system eigenvalues----------------%
P = K_2 / (inv(Q_u)*B.') %P should be greater than 0
eig_value = eig(A - B*K_2) %A_cl = (A - BK)* lambda

% 2)
% Feedforward gain kr
kxku = [A B ; Cr D_u]\([zeros(4,2); eye(2)])
k_x = kxku(1:4,:) %extract k_x
k_u = kxku(5:6,:) %Extract k_u
kr = k_u + K_2*k_x

% 3) Integral Control
%--------------------desired eigenvalues------------%
eig_des = [-1+0.4i ; -1-0.4i; -2;-3;-4;-5] %desired Eigenvalues

% augmented matrices
rho = 1;
Q_u = rho * eye(2)
cr_1 = [C(1,:) + C(3,:)];
cr_2 = [C(2,:)];
Cr = [cr_1 ; cr_2]; %Define Cr

Caug = [Cr zeros(2,2)]
D_u = [0 0 ; 0 0 ];

Aaug = [A zeros(4,2); -Cr zeros(2,2)]
```

```
Baug = [B; -D_u]
Daug = D_u
sys_aug = ss(Aaug,Baug,Caug,Daug)
% State feedback gain K
Qx1 = eye(6) %Changing QX to match 6x6
%K_aug = lqr(sys_aug,Qx1,Q_u) %using LQR
K_aug = place(Aaug,Baug,eig_des)
Kaug = K_aug(:,1:4) % Gain K matrix
Ki = -K_aug(:,5:end) % Integral Control Matrix
Acl = eig(A - B*Kaug)
%------------Different Eigenvalues, Augmented--------%
eig_des1 = [-3+0.8i;-3-0.8i;-5;-8;-9;-10]
K_aug1 = place(Aaug,Baug,eig_des1)
Kaug1 = K_aug1(:,1:4)
Ki1 = -K_aug1(:,5:end)
Acl1 = eig(A - B*Kaug1)
```

Session 3 Part I - Observer design

```
%Observer design
% 1)
% desired eigenvalues
desired_eig = [-2+0.4i;-2-0.4i;-4;-6]; %Observer E.values
% Observer gain matrix L
K3 = place(A',C',desired_eig);
L = K3'             %Gain of the observer
N = norm(L)
%-------------Different Eigenvalues -------%
desired_eig1 = [-3+0.8i;-3-0.8i;-6;-8] %New E.values of obs
K4 = place (A',C',desired_eig1)
L1 = K4' %New Gain of the observer
N1 = norm (L1)
%--------------Eigenvalues of closed loop compensated sysed-----%
Closed_eigen = eig(A- L*C) %Observer eigenvalues of
Closed_eigen2 = eig(A-B*Kaug) %Controller Eigenvalues (Previous Controller used)
%-----------Observer with factor of 100x further left--------%
desired_eig2 = [-200+40i;-200-40i;-400;-600]
K5 = place (A',C',desired_eig2)
L2 = K5'
```

Session 3 Part II - Kalman Filter design

```
% KF design
sys_kalman = ss(A, [B B], C, 0);
% Design matrices
Rv = eye(3)
Rw = eye(4)
[kest,L,P] = kalman(sys_kalman,Rw,Rv,0); %L = gain matrix
%----------------- Different design matrices - Measurement noice---%
Rv1 = eye(3);
Io = [3 3 3 3];
Rw1 = diag(Io);
sys_kalman1 = ss(A, [B B], C, 0);
[kest1,L1,P1] = kalman(sys_kalman1,Rw1,Rv1,0);

% End
```

Bibliography

Murray, K. J. (n.d.). *Feedback Systems* .