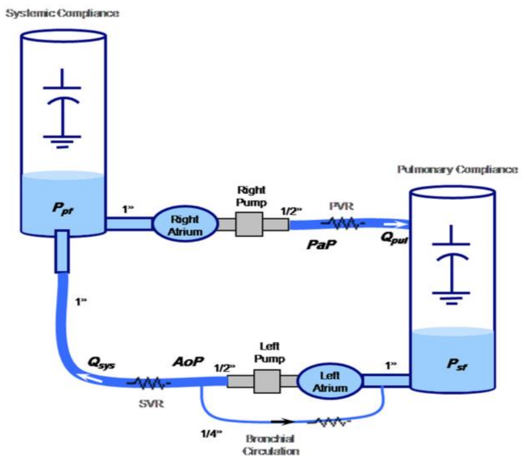
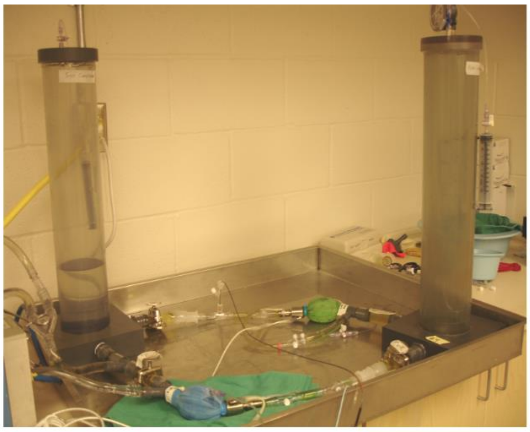
Robust Multivariable Control of a Continuous Flow Total Artificial Heart (CFTAH) embedded within a Mock Circulatory System (MCS)



Schematic of the Mock Loop



Picture of the Actual Mock Loop

Rahul Vivek Sawant

UHID – 1632656

Cullen College of Engineering, UH

MECE 7362 – Robust Multivariable Control

Dr. Matthew A. Franchek

UNIVERSITY of HOUSTON

CULLEN COLLEGE of ENGINEERING

Department of Mechanical Engineering

Table of Contents

Abstract

In this project, we will look at three feedback design methodologies for Multiple-Input-Multiple-Output Continuous Flow Total Artificial Heart (CFTAH) embedded within a Mock Circulatory System (MCS) and design the robust multivariable control. The methods we will be going through are Diagonally Dominant method, Precompensated Static Decoupled method and Precompensated Dynamically Decoupled method. Purpose of designing robust control for MIMO systems is to avoid any unnecessary, unwanted changes in the system after deployment. At the end of this project, we will understand how to work with MIMO systems and design robust controllers for such systems. Along with this, we will learn to use commands like SISOTOOL from MATLAB as well as Nichols Chart, bode plot.

1. Introduction

The system we are dealing with is Continuous Flow Total Artificial Heart (CFTAH) embedded with Mock Circulatory System (MCS). It is a real system that led to the preclinical studies at Texas Heart Institute. The control objective is to emulate the native heart by robustly regulating blood flow in the presence physiological changes due to vascular dynamics, blood property changes and continuous flow pump degradation.

2. Background

2.1 Feedback Controller Design

Designing a feedback controller include several steps and careful tuning of the feedback loop. There is no fix technique to design a controller.

Almost in each case, the plant dynamics are always given to the user. The first task is to identify the nature of plant e.g. if the given system is minimum phase or non-minimal phase (unstable). If the system is non-minimal, then we cannot use the same methodology of minimal phase systems. Once the nature of the system is identified, we can look over the performance objectives required for the feedback controller. Some of the open loop performance objectives include settling time, crossover frequency, percent overshoot. Whereas closed loop performance objectives include phase margin or M-circle constraint. Steady state performance objectives are given by internal model principal. It turns out to be essential to add an integrator to each feedback controller we design.

After considering performance objectives, one can proceed to loop shaping. Loop shaping is performed on Nichols chart. To start off with the process, controller poles that are lowest common multiple in the input denominator can be taken in consideration. To meet the M-circle criteria and crossover frequency specification, one needs to cancel the plant dynamics with the help of a controller. Here comes the problem for non-minimal phase systems. Since, NMP systems consists of unstable poles, we cannot directly cancel out. Systems with unstable poles are most dominant poles. The final step in designing includes addition of high frequency poles to make the controller strictly proper i.e. order of denominator must be higher than order of numerator in transfer function of the controller such that controller should not contain differentiator in it. Differentiator increases the noise. Hence, we avoid ‘s’ raise to positive number terms in the controller.

2.2 Problem Description

The left axial pump supplies oxygenated blood to the systemic circulatory system through the aorta and the right pump takes the deoxygenated blood to the lungs (Pulmonary system). The diagonally dominant linearized model for the system is

$$\begin{bmatrix}\delta Q_{sys}\\ \delta Q_{pul}\end{bmatrix}(s) = \begin{bmatrix} 0.76\,\dfrac{(\frac{s}{0.31}+1)}{(\frac{s}{0.42}+1)(\frac{s}{3.23}+1)} & 0.53\,\dfrac{1}{(\frac{s}{0.42}+1)(\frac{s}{2.64}+1)} \\ 0.66\,\dfrac{1}{(\frac{s}{0.42}+1)(\frac{s}{2.64}+1)} & 0.66\,\dfrac{(\frac{s}{0.20}+1)}{(\frac{s}{0.42}+1)(\frac{s}{3.19}+1)} \end{bmatrix}\begin{bmatrix}\delta V_L(s)\\ \delta V_R(s)\end{bmatrix}$$

where the outputs, perturbations in flow $\delta Q_{sys}(s)$ and $\delta Q_{pul}(s)$, have units of liters/min and the inputs, perturbation in pump input motor voltages $\delta V_L(s)$ and $\delta V_R(s)$, have units of volts.

A) Design a diagonal integral multivariable controller by treating the MIMO system as two independent SISO system by ignoring loop interactions. Design your controller using Nichols chart to achieve crossover frequency of 2 rad/sec.

B) Design a diagonal multivariable controller in-series with Static Decoupling controller for the same system. Identify your decoupling controller using the DC gain of the MIMO plant model. Design your controller using Nichols chart to achieve crossover frequency of 2 rad/sec.

C) Design a diagonal multivariable controller using Dynamic Decoupling method for the same system. Design your controller using Nichols chart to achieve crossover frequency of 2 rad/sec.

3. Methodology

3.1 Diagonally Dominant Multivariable Control

In this type of control methodology, we treat MIMO system as two independent SISO systems. Here, we design a controller matrix by considering only diagonal terms in the plant transfer matrix which are P_{11} & P_{22}. In this method one must ignore the loop interactions. We need to design first controller that is g_{11} by considering Loop 1 as,

$$L_1(s) = g_{11}(s)P_{11}(s) = g_{11}(s) * 0.76 \frac{(\frac{s}{0.31}+1)}{(\frac{s}{0.42}+1)(\frac{s}{3.23}+1)}$$

and second controller that is g_{22} by considering Loop 2 as,

$$L_2(s) = g_{22}(s)P_{22}(s) = g_{22}(s) * 0.66 \frac{(\frac{s}{0.20}+1)}{(\frac{s}{0.42}+1)(\frac{s}{3.19}+1)}$$

where the feedback controller is,

$$G(s) = \begin{bmatrix} \frac{K_1}{s} & 0 \\ 0 & \frac{K_2}{s} \end{bmatrix}$$

The general block diagram for Diagonal Multivariable Control can be given as follows

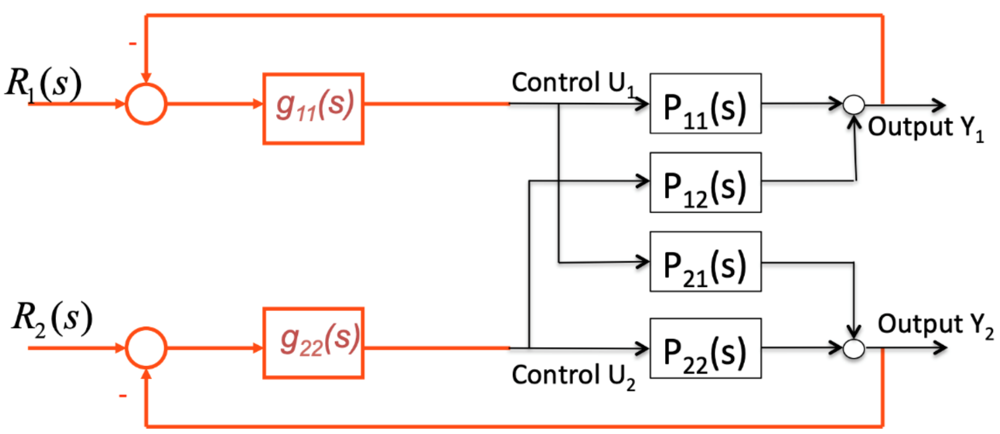


figure 1 - Block diagram for diagonally dominant multivariable control

3.1.1 Design G_{11}

To design a feedback controller in MATLAB, we use SISOTOOL. To do this we first input the transfer function for P_{11} in MATLAB command window.

After this, we type in the command SISOTOOL in MATLAB command window which will open new window in which there are different control architectures available to design a feedback controller.

After opening SISOTOOL, we need to select the architecture and need to import the plant transfer function in the architecture.

Once the plant transfer function is imported in the required architecture, the compensator is ready to tune with Nichols chart.

Since, here the controller type is restricted to the $\frac{K}{s}$, we only need to add integrator in the design.

After adding an integrator, we can adjust the gain K for the controller on Nichols chart to meet the required performance specification which is having a crossover frequency at 2 rad/sec. Tuning the controller performance to required specifications, the gain value obtained at $K_1 = 2.308$.

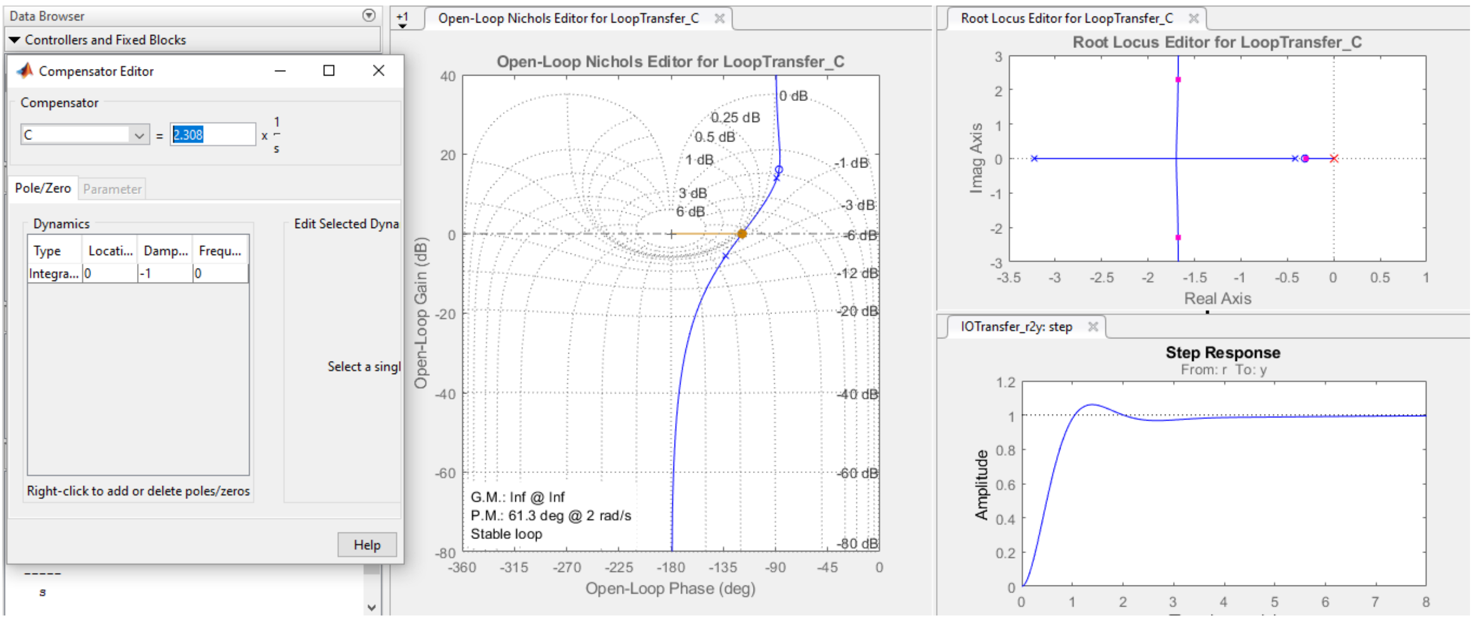


figure 2 - Controller design for Loop 1

Below is the transfer function used for controller design using loop 1.

```
G =

       2.452 s + 0.76
  ------------------------
  0.7371 s^2 + 2.691 s + 1

Continuous-time transfer function.
```

3.1.2 Design G_{22}

Following the same procedure that is used for designing g_{11}, g_{22} can also be designed.

Here also, the required performance specification is the crossover frequency of 2 rad/sec.

The controller type is also restricted to $\frac{K}{s}$.

Tuning the second controller to the required performance specification, we obtain the gain value of K_2 = 1.7299.

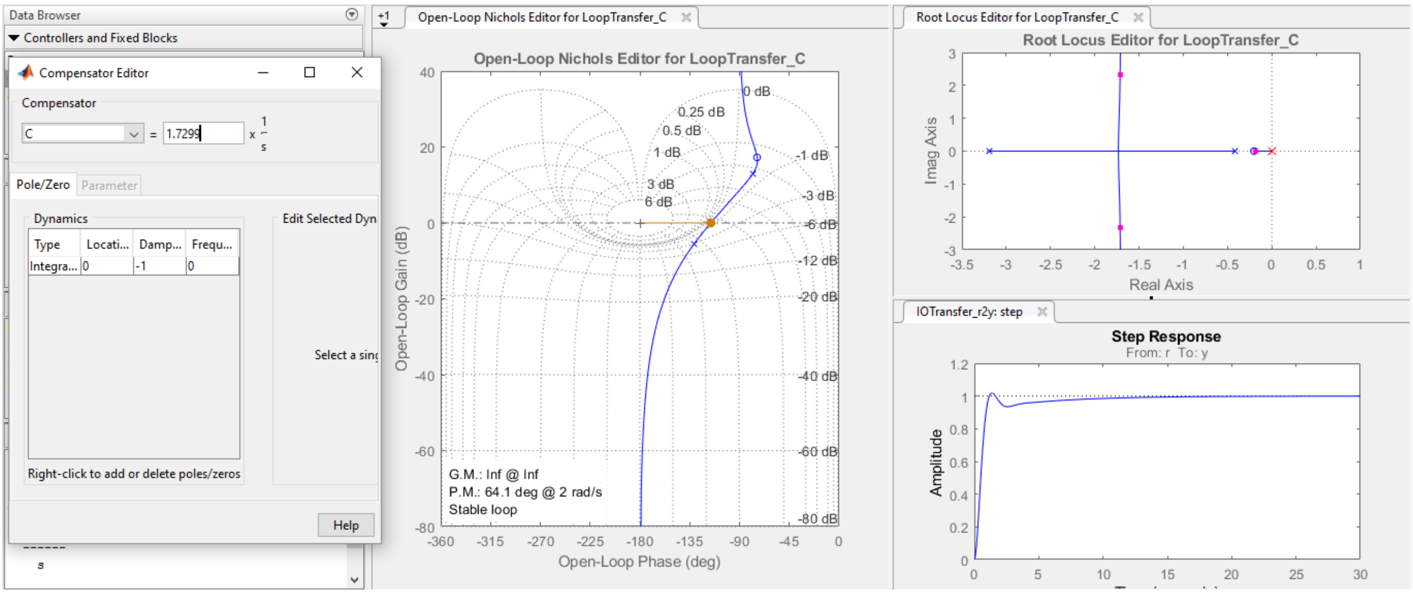


figure 3 - Controller design for Loop 2

Below is the transfer function used for controller design using loop 2.

```
G =

        3.3 s + 0.66
  ----------------------
  0.7464 s^2 + 2.694 s + 1

Continuous-time transfer function.
```

3.1.3 MATLAB code and Simulink model for Diagonally Dominant System

```
close all
clear all
clc

% Plant transfer functions
P11 = tf(0.76*[1/0.31 1],conv([1/0.42 1],[1/3.23 1]));
P12 = tf(0.53,conv([1/0.42 1],[1/2.64 1]));
P21 = tf(0.66,conv([1/0.42 1],[1/2.93 1]));
P22 = tf(0.66*[1/0.20 1],conv([1/0.42 1],[1/3.19 1]));
Plant = [P11 P12;P21 P22]

figure(1)
bode(Plant)
grid on
title('Bode plot for diagonally dominant CFTAH system')

% The amplitude of the off-diagonal terms is less than the amplitude
% of the diagonal term. Therefore, the given system is diagonally dominant.

% Defining G11
G11 = tf(2.308,[1 0])
% Defining G22
G22 = tf(1.7299,[1 0])
% Overall controller matrix
Controller = [G11 0;0 G22]

% Loop 1 design
L1 = G11*P11
figure(2);
nichols(L1);
hold on
grid on
% Loop 2 design
L2 = G22*P22

nichols(L2);
grid on
hold off
```

```
%% Problem 1
%
tfinal = 30;
h = 0.01;
time = 0:h:tfinal;
sim problem1_simulink
figure(3)
subplot(4,1,1)
plot(time,Qsys,'linewidth',2)
title('SAWANT: MECE 7362 - Final Project - Problem 1 Simulink Output')
ylim([0 6])
grid on
ylabel('Systemic flow')
subplot(4,1,2)
plot(time,Qpul,'linewidth',2)
ylim([0 8])
grid on
ylabel('Pulmonary flow')
subplot(4,1,3)
plot(time,VL,'linewidth',2)
ylim([0 6])
grid on
ylabel('Left Pump Voltage')
subplot(4,1,4)
plot(time,VR,'linewidth',2)
ylim([0 5])
grid on
ylabel('Right Pump Voltage')
xlabel('Time')
%
```

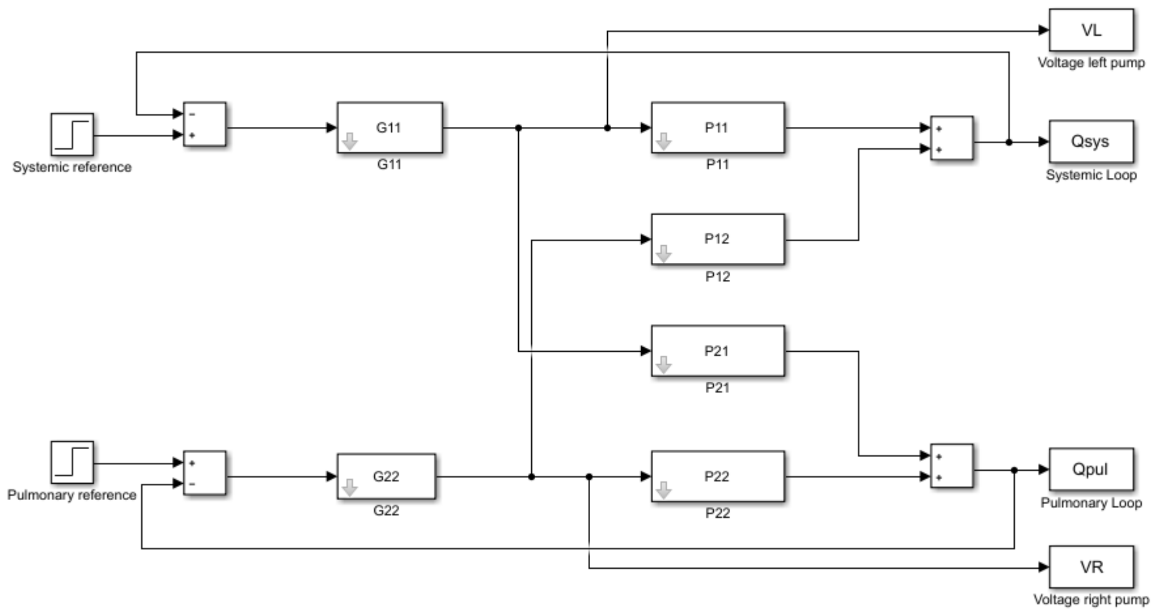


figure 4 - Simulink Model

3.1.4 Results of Diagonally Dominant Multivariable Control

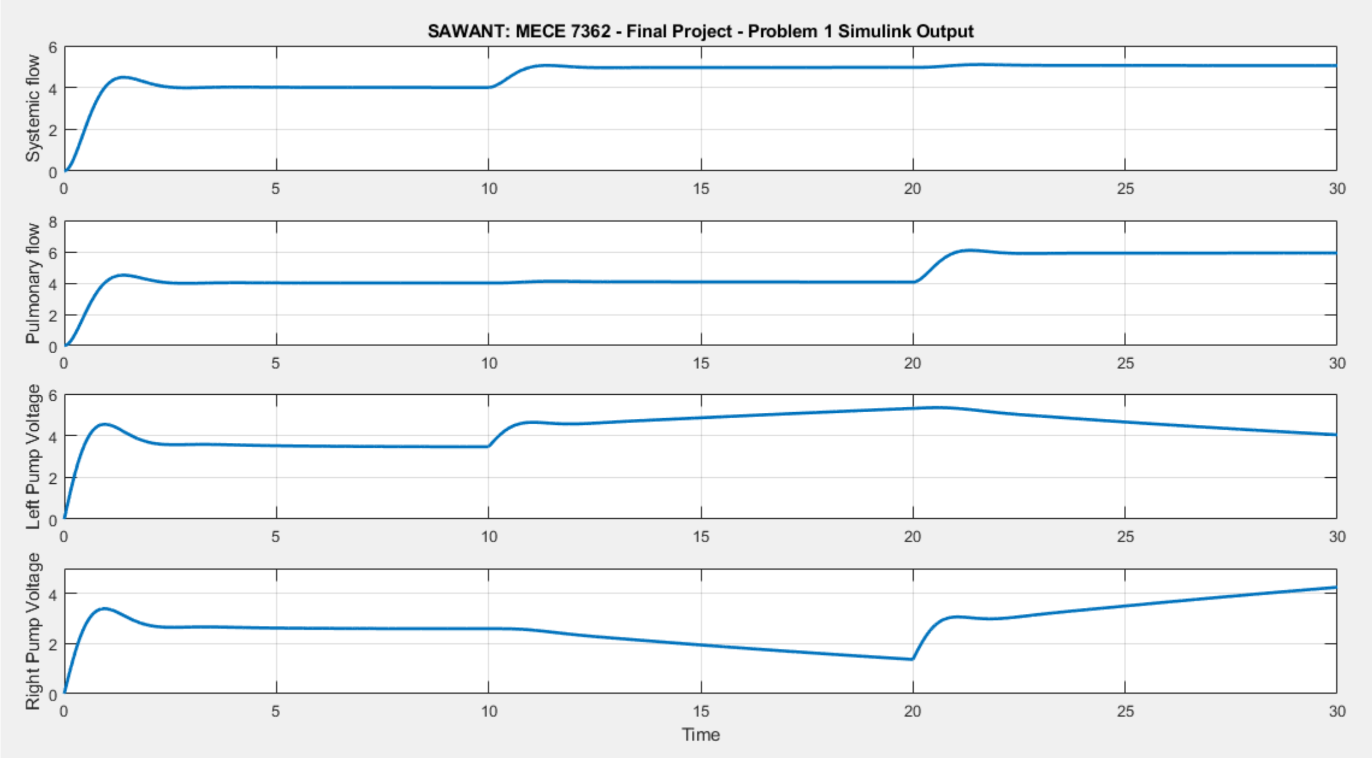


figure 5 - Transient response for Diagonally Dominant Multivariable Control

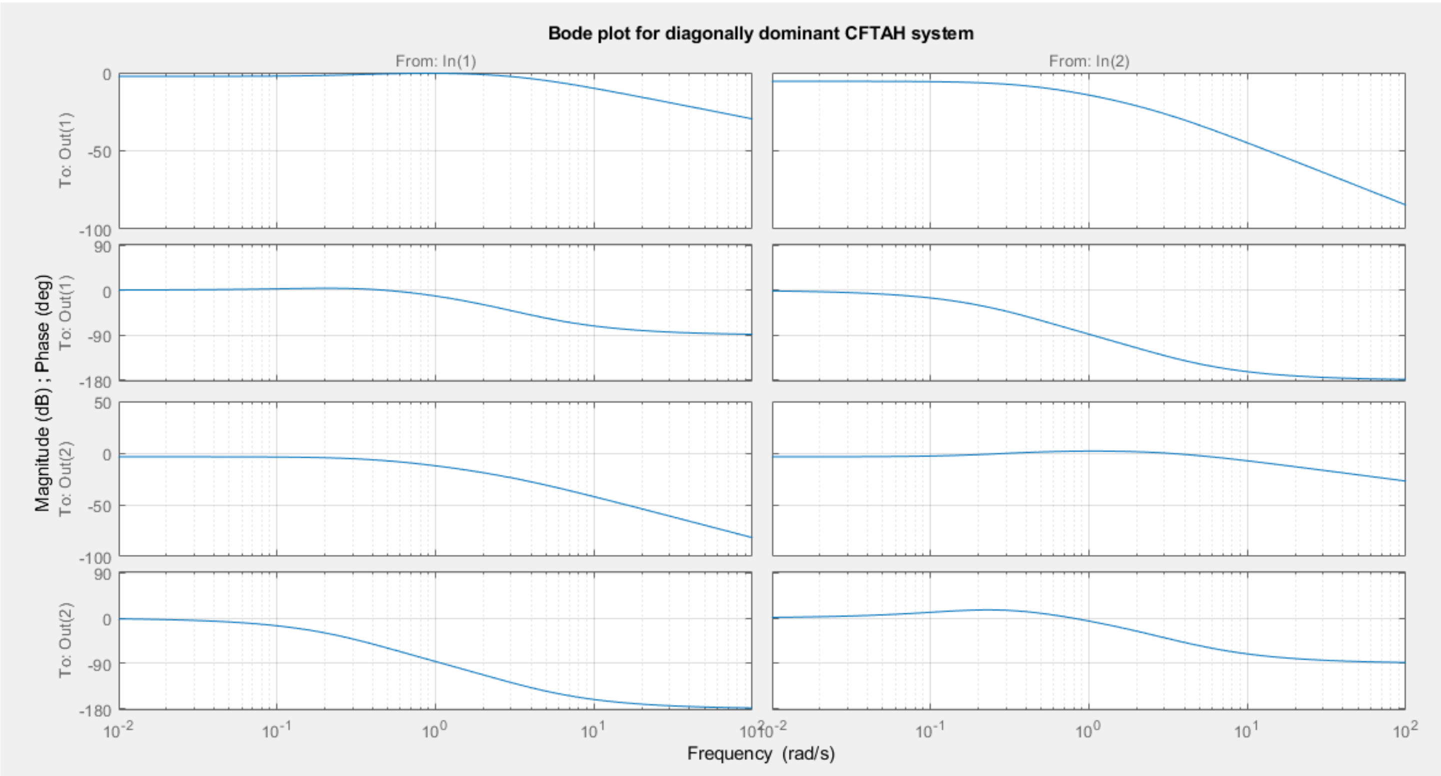


figure 6 – Bode plot for diagonally dominant CFTAH system

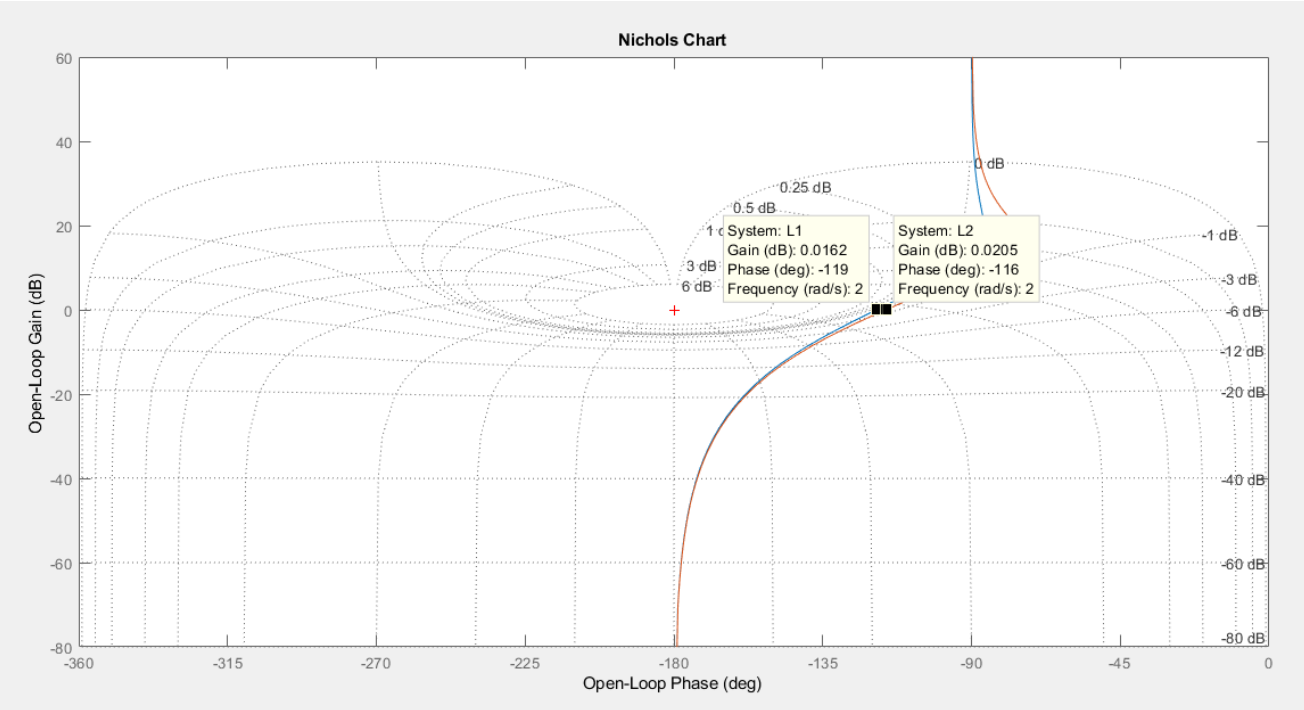


figure 7 - Nichols chart response for loops L1 and L2

3.2 Statically Decoupled (DC Decoupled) Diagonal Multivariable Control

In this type of design, one must take loop interactions into account. The basic idea behind this type of controllers is we decouple the system statically, i.e., put s = 0 in the plant transfer function. After solving, we get a constant matrix. To find the DC gains that need to be applied to the model, we need to invert the constant matrix obtained.

In this example, the matrix that we get by putting s = 0 is,

$$P = \begin{bmatrix} 0.76 & 0.53 \\ 0.66 & 0.66 \end{bmatrix}$$

Now, if we take an inverse of this matrix, we get,

$$\hat{P} = \begin{bmatrix} 4.3478 & -3.4914 \\ -4.3478 & 5.0066 \end{bmatrix}$$

This is also called the gain matrix (A).

To get the new plant dynamics transfer matrix, we need to pre-multiply this gain matrix by old plant matrix. After obtaining the resulting new plant transfer matrix, we can design the controllers using the method we followed for Diagonally Dominant Multivariable Control design.

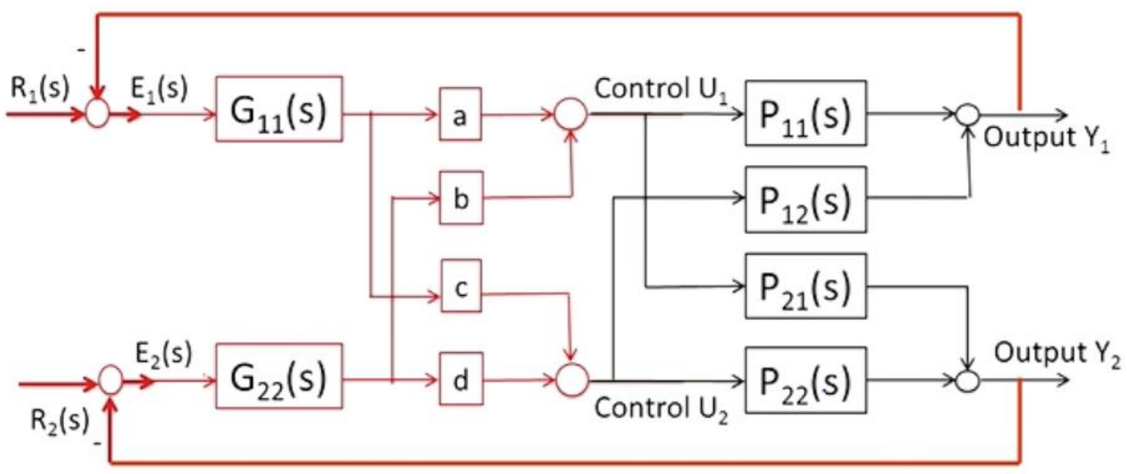


figure 8 – Block diagram for Statically Decoupled Diagonal Multivariable Control

Static gain matrix,

```
0.7600    0.5300
0.6600    0.6600
```

Here is the pre-compensated matrix,

```
 4.3478   -3.4914
-4.3478    5.0066
```

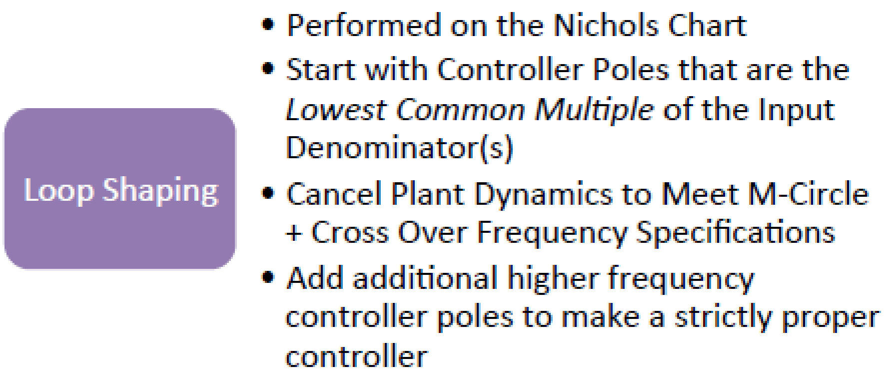
Here is the new plant obtained by pre-multiplying precompensated matrix with old plant matrix,

```
Plant_new =

  From input 1 to output...
                9.613 s^3 + 30.7 s^2 + 13.58 s + 1
   1:  ----------------------------------------------
       0.6648 s^4 + 4.461 s^3 + 9.064 s^2 + 5.45 s + 1

                 -11.66 s^3 - 39.25 s^2 - 14.43 s
   2:  -----------------------------------------------
       0.6065 s^4 + 4.221 s^3 + 8.894 s^2 + 5.417 s + 1

  From input 2 to output...
            -7.72 s^3 - 24.06 s^2 - 8.743 s - 4.441e-16
   1:  ----------------------------------------------
       0.6648 s^4 + 4.461 s^3 + 9.064 s^2 + 5.45 s + 1

                13.43 s^3 + 45.94 s^2 + 19.31 s + 1
   2:  -----------------------------------------------
       0.6065 s^4 + 4.221 s^3 + 8.894 s^2 + 5.417 s + 1

Continuous-time transfer function.
```

To design controllers in this method, we need plant matrix to find transfer function for loop 1 and 2. Transfer functions Pnew11 and Pnew22 will be used in SISOTOOL to design the controller.

3.2.1 Design G_{11}

Once we obtain the pre-compensated plant dynamics, we can use SISOTOOL so design the g_{11}. As of to start every controller design, we can start off by adding an integrator to the system. The response obtained by using only an integrator was quite different than the one professor got. The response contained overshoot at the beginning. To reduce the unwanted overshoot, we can use loop shaping method.



To make the controller strictly proper, I keep the power of denominator higher than numerator, i.e., number of poles are more than number of zeros in controller transfer function.

First, I added an integrator in the design for steady state response.

Second, I cancelled out the slowest poles of the loop 1 by keeping similar zeros in the numerator of controller transfer function. Since all poles are stable, we can cancel the dynamics directly. Instead of cancelling all poles of the loop 1 transfer function, I considered only slowest and dominant poles.

Third, I tuned the gain value to adjust the Nichols plot such that crossover frequency occurs at 2 rad/sec. Here is the response of loop 1 on Nichols chart using SISOTOOL.

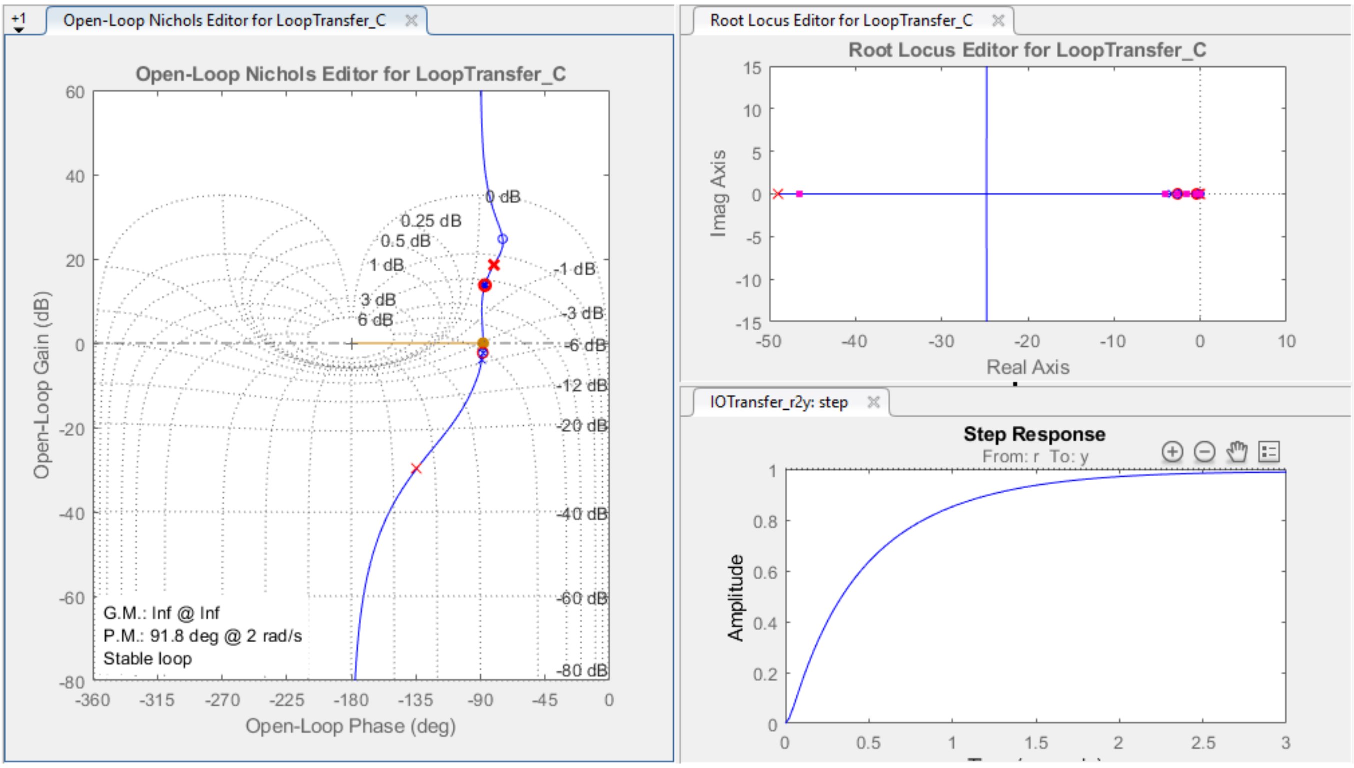


figure 9 – Nichols plot for loop 1 on SISOTOOL

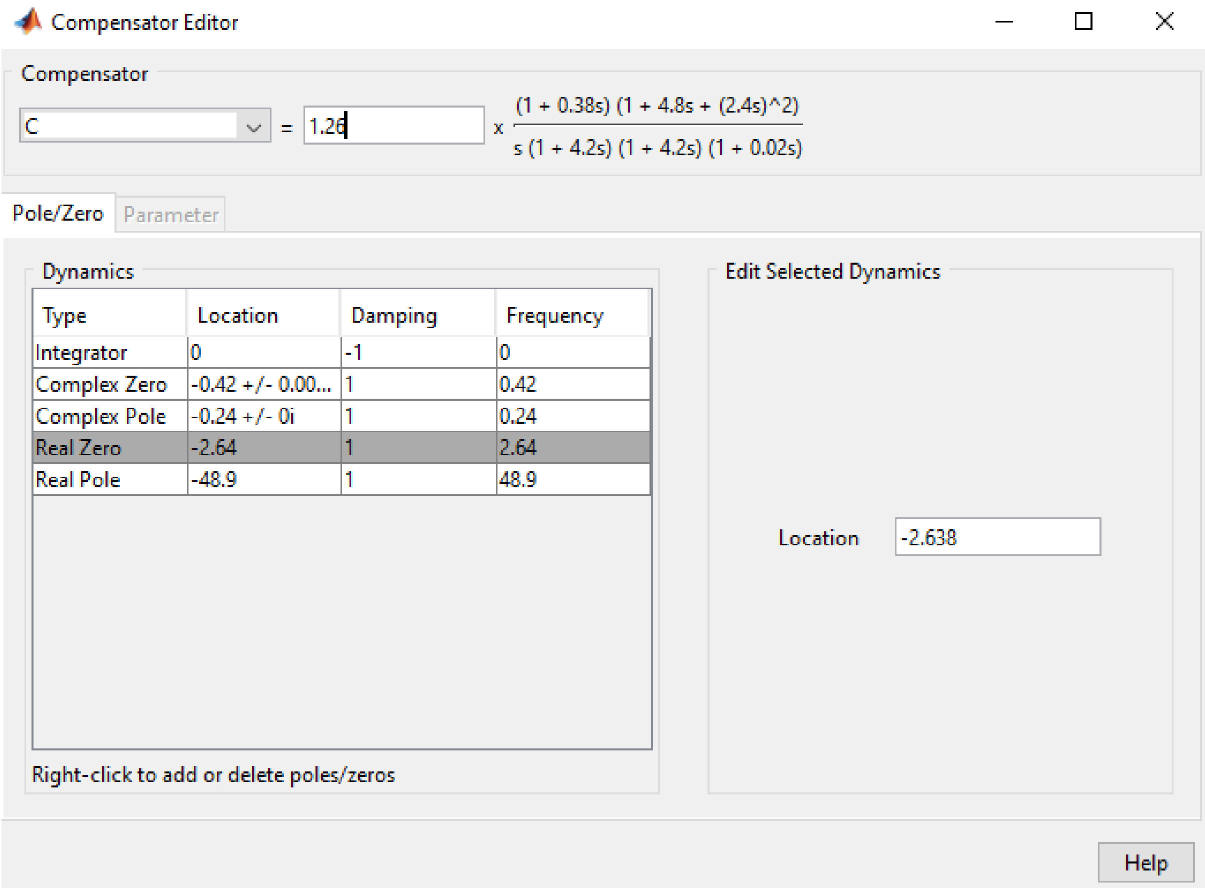


figure 10 – G11 controller equation

Below is the transfer function used for controller design using loop 1.

```
Plant_newl1 =

        9.613 s^3 + 30.7 s^2 + 13.58 s + 1
  -----------------------------------------------
  0.6648 s^4 + 4.461 s^3 + 9.064 s^2 + 5.45 s + 1

Continuous-time transfer function.
```

3.2.1 Design G_{22}

Like the previous design, I designed controller G22 for loop 2 using loop shaping technique on SISOTOOL. Here are the results of controller G22.

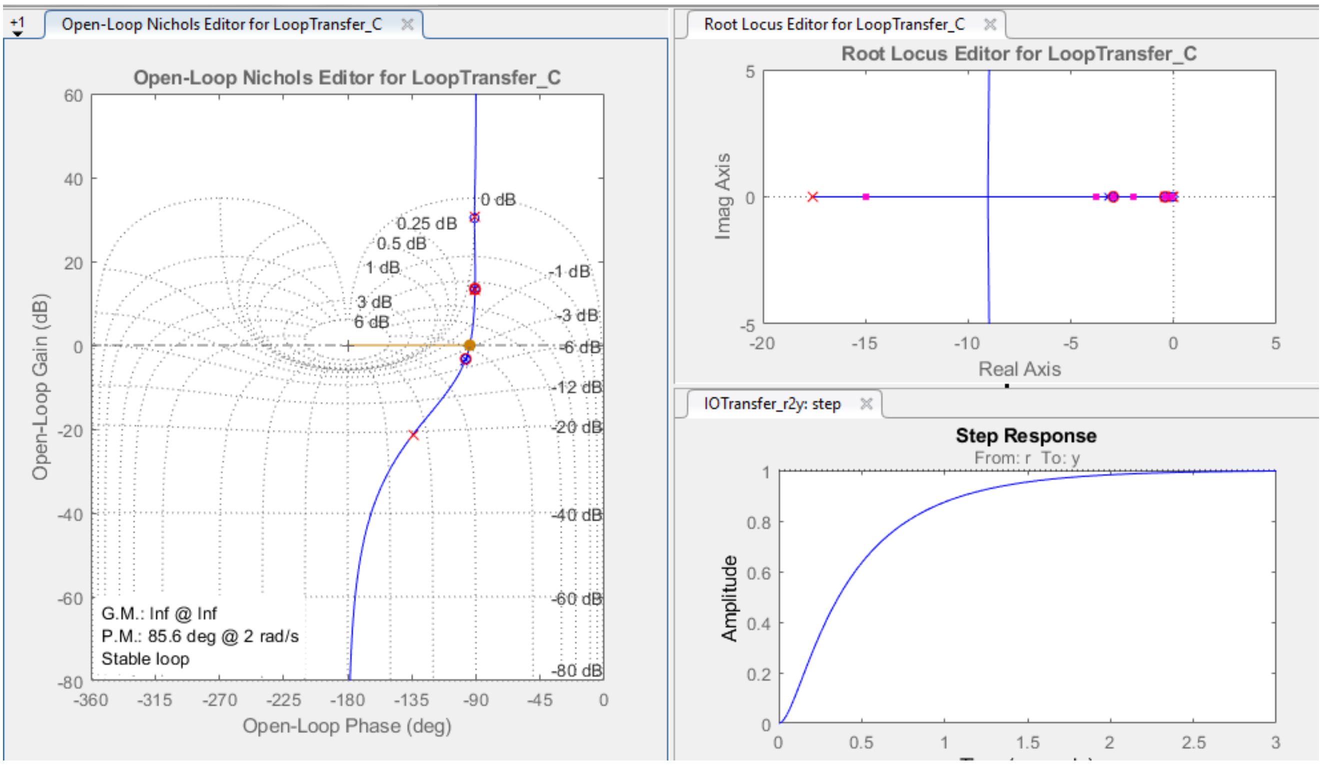


figure 11 – Nichols plot for loop 2 on SISOTOOL

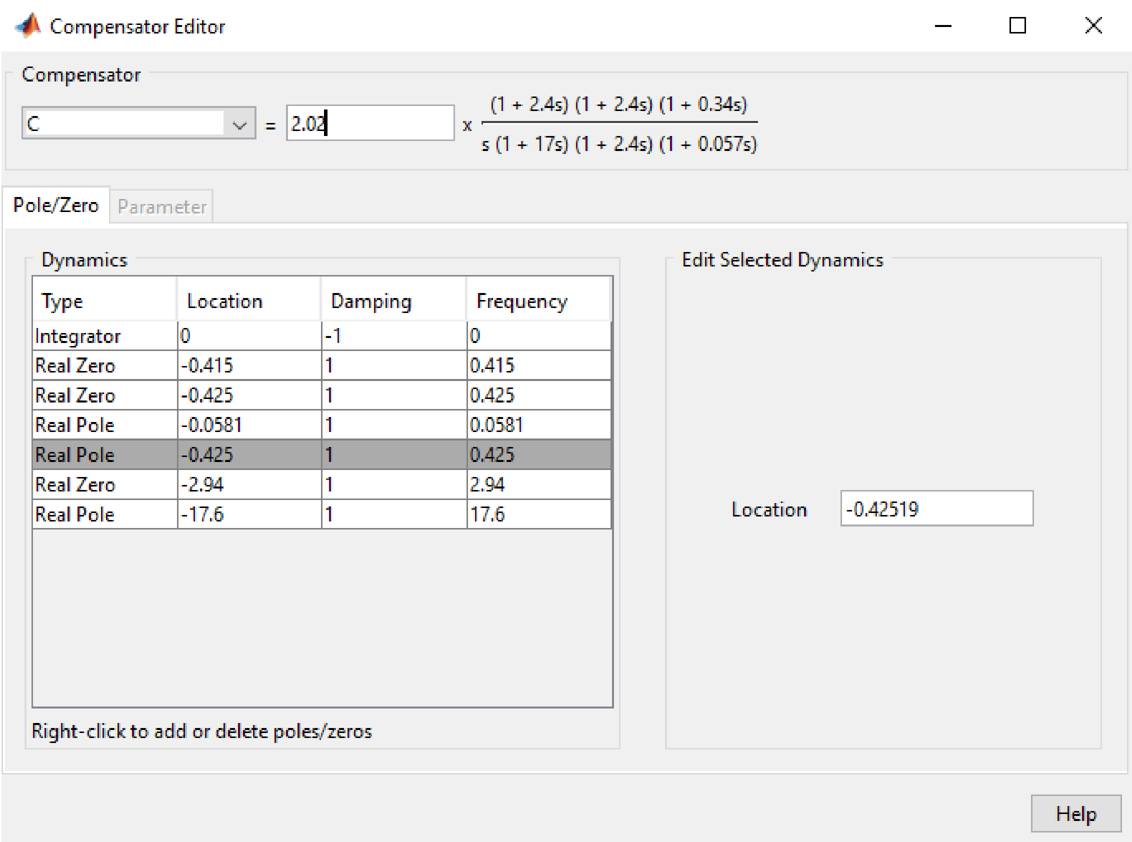


figure 12 – G22 controller equation

Below is the transfer function used for controller design using loop 2.

```
Plant_new22 =

        13.43 s^3 + 45.94 s^2 + 19.31 s + 1
  ------------------------------------------------
  0.6065 s^4 + 4.221 s^3 + 8.894 s^2 + 5.417 s + 1

Continuous-time transfer function.
```

3.2.3 MATLAB code and Simulink model for DC Decoupled Control

```
close all
clear all
clc

% Plant transfer functions
P11 = tf(0.76*[1/0.31 1],conv([1/0.42 1],[1/3.23 1]));
P12 = tf(0.53,conv([1/0.42 1],[1/2.64 1]));
P21 = tf(0.66,conv([1/0.42 1],[1/2.93 1]));
P22 = tf(0.66*[1/0.20 1],conv([1/0.42 1],[1/3.19 1]));
Plant = [P11 P12;P21 P22]

figure(1)
bode(Plant)
grid on
title('Bode plot for diagonally dominant CFTAH system')
```

```
% Defining G11
num1 = conv([0.38 1],[5.76 4.8 1])
den1 = conv(conv(conv([1 0],[4.2 1]),[4.2 1]),[0.02 1])
G11 = tf(1.26*num1,den1)
% Defining G22
num2 = conv(conv([2.4 1],[2.4 1]),[0.34 1])
den2 = conv(conv(conv([1 0],[17 1]),[2.4 1]),[0.057 1])
G22 = tf(2.02*num2,den2)
% Overall controller matrix
Controller = [G11 0;0 G22]

%% Decoupling
% Static gain matrix
A = [0.76 0.53; 0.66 0.66]
% New plant
Plant_new = Plant*inv(A)
Plant_new11 = Plant_new(1,1)
Plant_new12 = Plant_new(1,2)
Plant_new21 = Plant_new(2,1)
Plant_new22 = Plant_new(2,2)

% Loop 1 design
L1 = G11*Plant_new11
figure(2);
nichols(L1);
hold on
grid on
% Loop 2 design
L2 = G22*Plant_new22
nichols(L2);
grid on
hold off

%% Problem 2
%
tfinal = 30;
h = 0.01;
time = 0:h:tfinal;
sim problem2_simulink
figure(3)
subplot(4,1,1)
plot(time,Qsys,'linewidth',2)
title('SAWANT: MECE 7362 - Final Project - Problem 2 Simulink Output')
ylim([0 6])
grid on
ylabel('Systemic flow')
subplot(4,1,2)
plot(time,Qpul,'linewidth',2)
ylim([-2 8])
grid on
ylabel('Pulmonary flow')
subplot(4,1,3)
plot(time,VL,'linewidth',2)
ylim([0 6])
grid on
ylabel('Left Pump Voltage')
subplot(4,1,4)
plot(time,VR,'linewidth',2)
```

```
ylim([-2 6])
grid on
ylabel('Right Pump Voltage')
xlabel('Time')
%
```

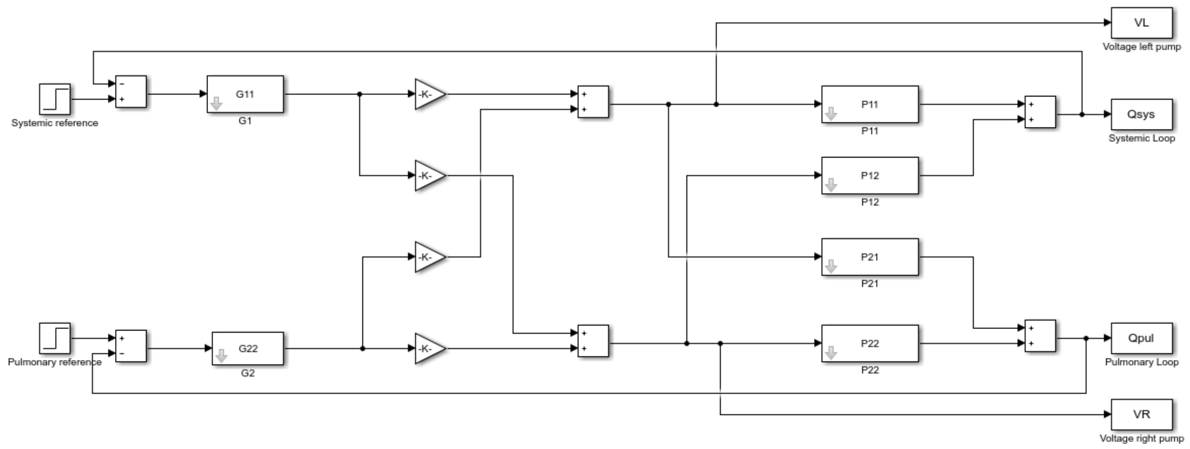


figure 12 – Simulink Model

3.2.4 Results of DC Decoupled Multivariable Control

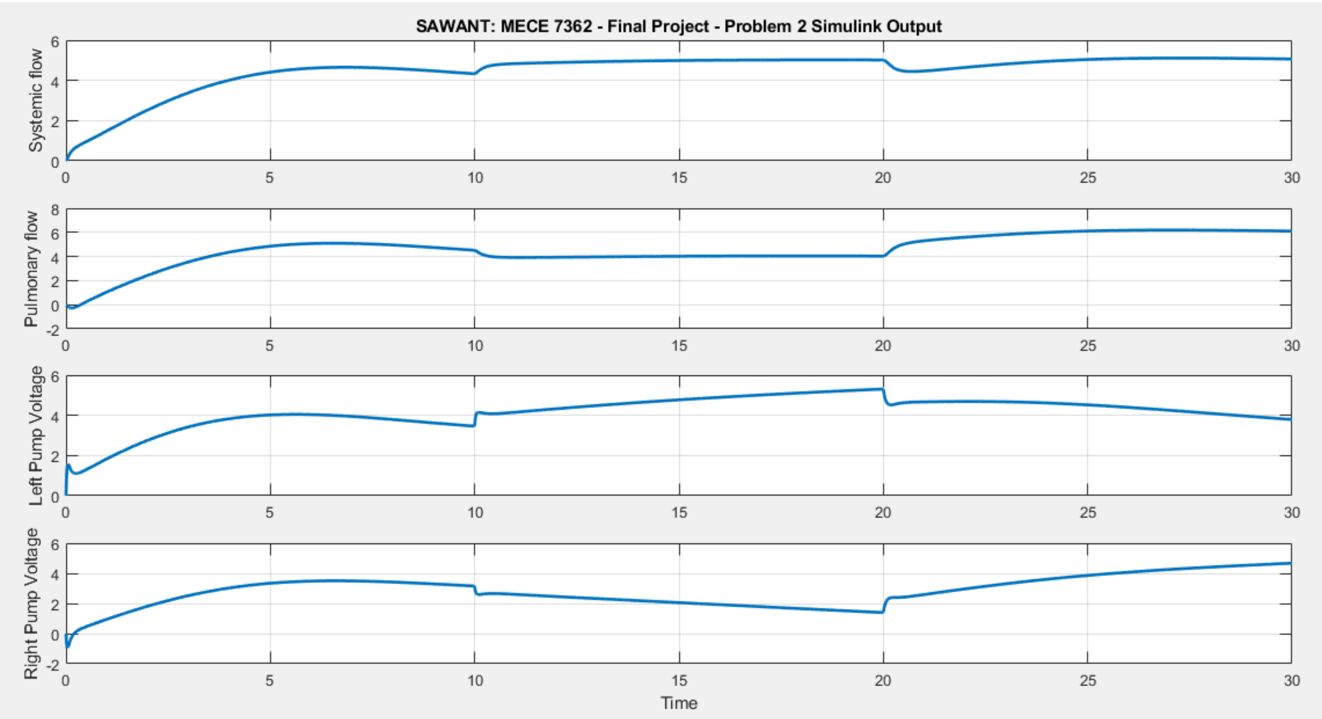


figure 13 – Transient response for DC Decoupled Multivariable Control

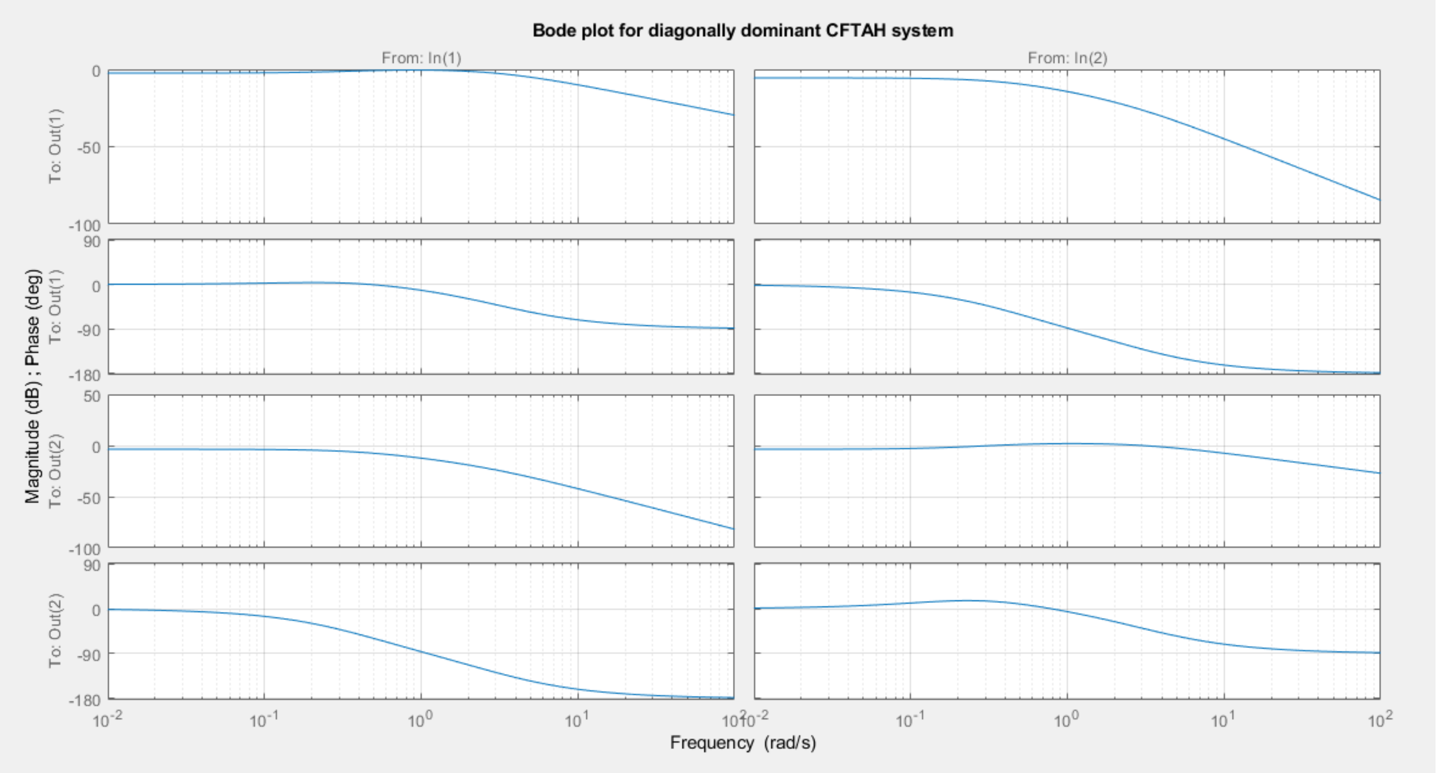


figure 14 – Bode plot for diagonally dominant CFTAH system

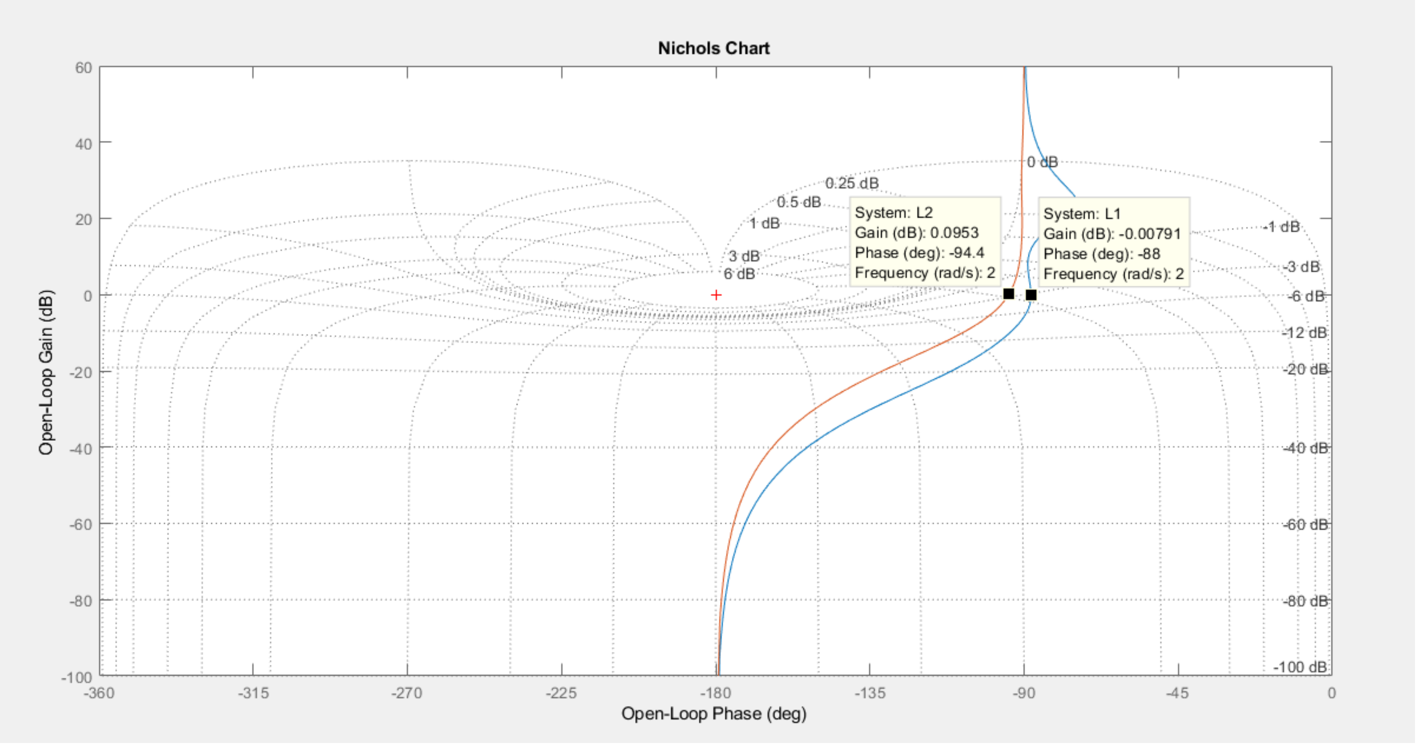


figure 15 – Nichols chart response for loops L1 and L2

3.3 Dynamically Decoupled diagonal multivariable control

In this type of control methodology, we cancel out the loop interactions dynamically, which turns out to be more efficient.

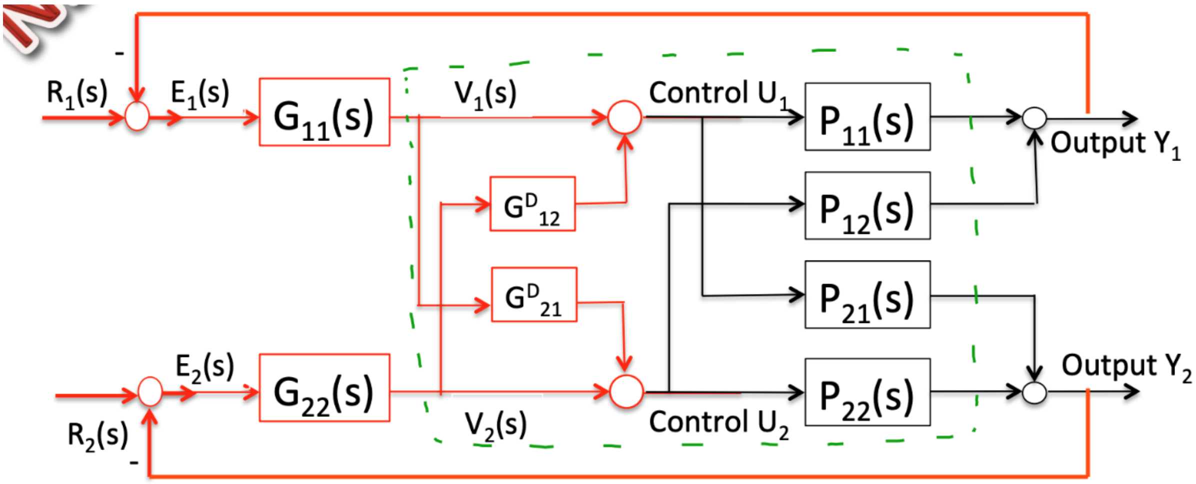


figure 16 – Block diagram for dynamically decoupled diagonal multivariable control

Writing the output equations for Y_1 and Y_2.

$$Y_1 = P_{11}(s)U_1(s) + P_{12}(s)U_2(s)$$

$$Y_2 = P_{21}(s)U_1(s) + P_{22}(s)U_2(s)$$

But $U_1 = V_1 + G_{12}^D V_2$ and $U_2 = V_2 + G_{21}^D V_1$

Therefore,

$$Y_1 = P_{11}(s)(V_1 + G_{12}^D V_2) + P_{12}(s)(V_2 + G_{21}^D V_1)$$

$$Y_2 = P_{21}(s)(V_1 + G_{12}^D V_2) + P_{22}(s)(V_2 + G_{21}^D V_1)$$

Here, to get decoupling, we need to make off diagonal terms zero. To achieve this,

$G_{12}^D = -\frac{P_{12}(s)}{P_{11}(s)}$ and $G_{21}^D = -\frac{P_{21}(s)}{P_{22}(s)}$

This will yield the new plant dynamics as,

$Y_1 = [P_{11}(s) - \frac{P_{12}(s)P_{21}(s)}{P_{22}(s)}]V_1(s)$ -------------------- 1 and

$$Y_2 = [P_{22}(s) - \frac{P_{21}(s)P_{12}(s)}{P_{11}(s)}]V_2(s) \text{ -------------------2}$$

whereas

$V_1(s) = G_{11}(s)E_1(s)$ and $V_2(s) = G_{22}(s)E_2(s)$

Now we need to design controllers using the terms in the brackets of the equations 1 and 2. As we decoupled the system, we can start designing G_{11} and G_{22} with the help of SISOTOOL in MATLAB.

3.3.1 Design G_{11}

We are going to use the same method I mentioned in Static Decoupled Multivariable Control to design controllers. Loop shaping helps achieve the robustness for the controller.

After following the same steps on SISOTOOL. Here are the results I got for controller G11 using loop 1.

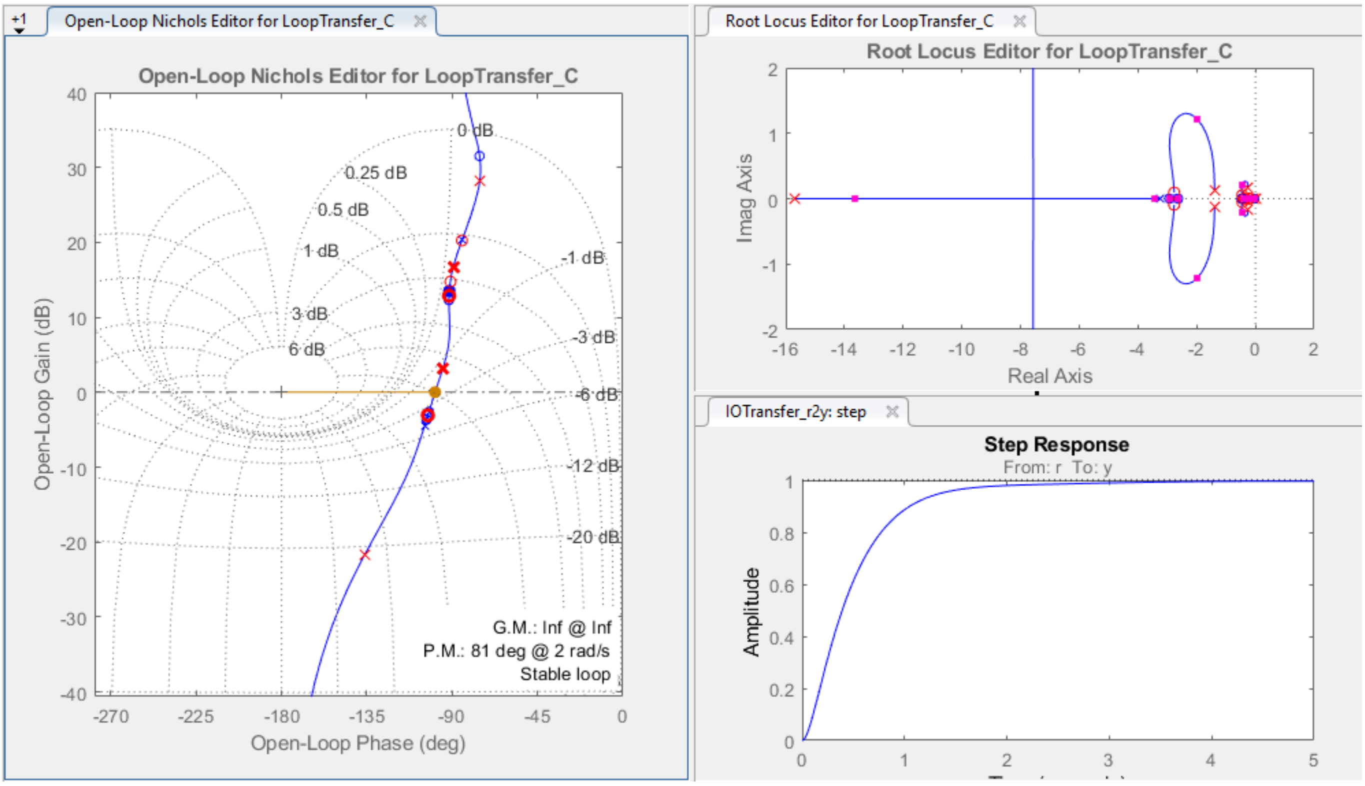


figure 16 – Nichols plot for loop 1 on SISOTOOL

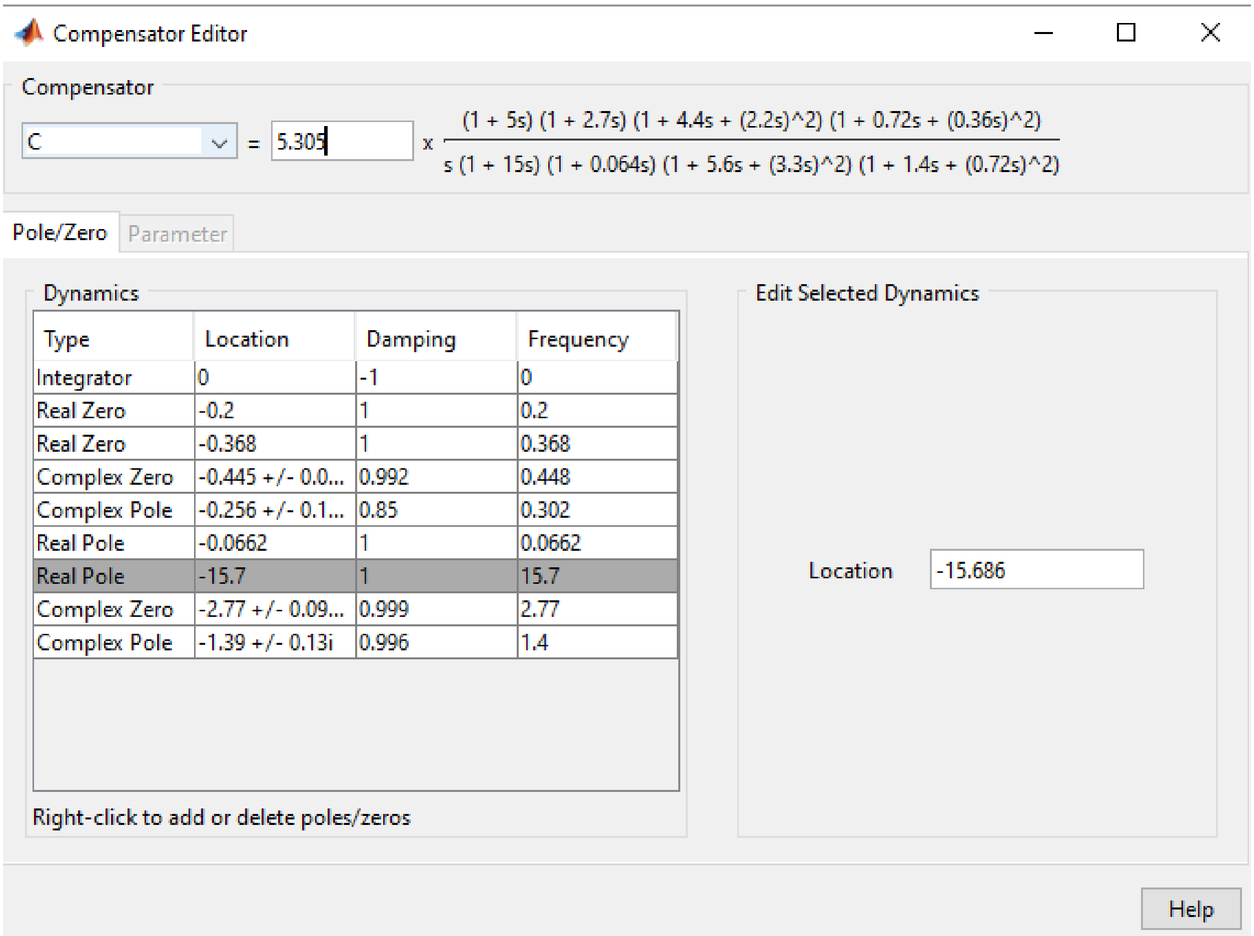


figure 17 – G11 controller equation

Here are the GD12 and GD21 transfer functions,

```
GD12 =

     -0.3907 s^2 - 1.426 s - 0.53
  --------------------------------------
  2.211 s^3 + 7.451 s^2 + 4.549 s + 0.76

Continuous-time transfer function.
```

```
GD21 =

     -0.4926 s^2 - 1.778 s - 0.66
  --------------------------------------
  2.682 s^3 + 9.52 s^2 + 5.097 s + 0.66

Continuous-time transfer function.
```

Below is the transfer function used for controller design using loop 1.

```
intermediate_L1 =

       5.929 s^6 + 41.03 s^5 + 94.21 s^4 + 83.38 s^3 + 32.28 s^2 + 4.992 s + 0.1518
  ---------------------------------------------------------------------------------------
  1.783 s^7 + 18.29 s^6 + 70.16 s^5 + 124.1 s^4 + 103.7 s^3 + 43.28 s^2 + 8.694 s + 0.66

Continuous-time transfer function.
```

3.3.2 Design G22

Similarly, we can design controller using loop 2 transfer function obtained from equation 2. Again, following the loop shaping method results a strictly proper controller G22. Here are the results I got for controller G22 using loop 2.

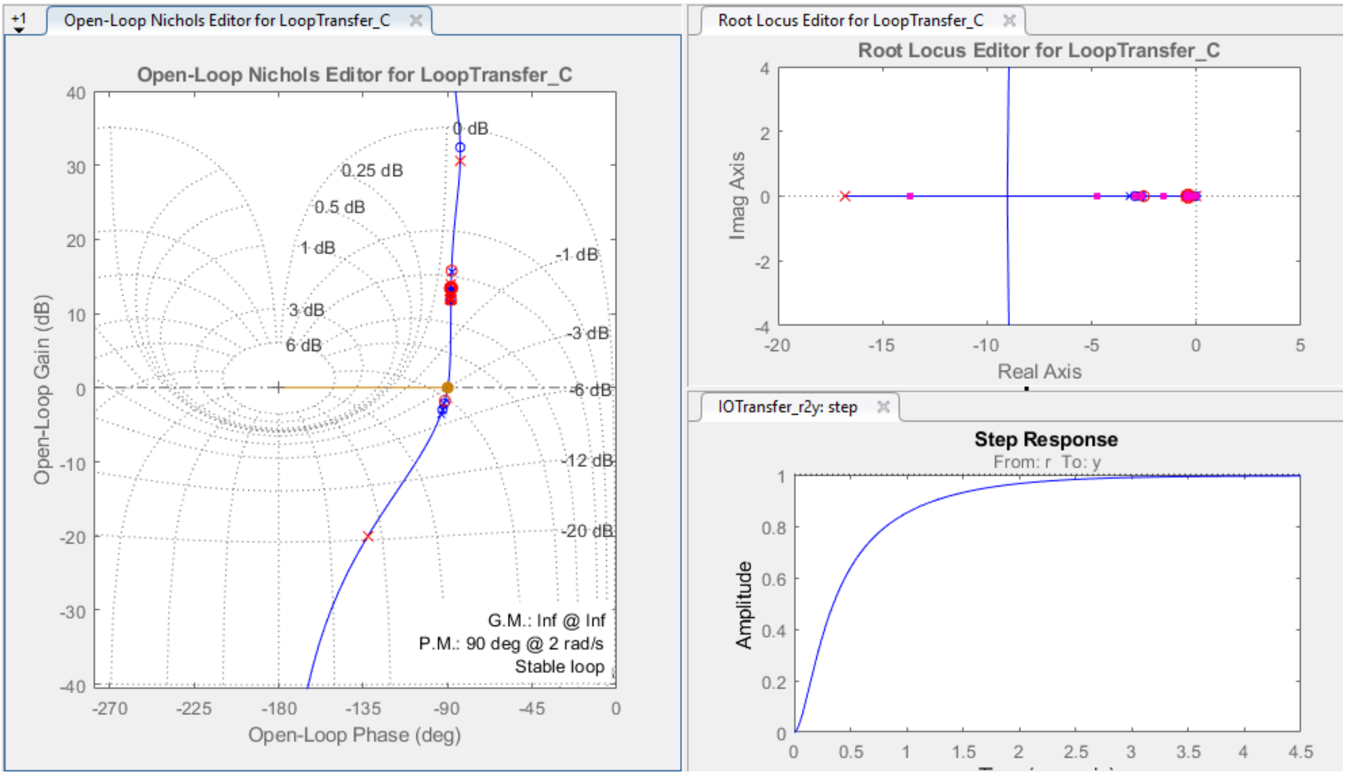


figure 18 – Nichols plot for loop 1 on SISOTOOL

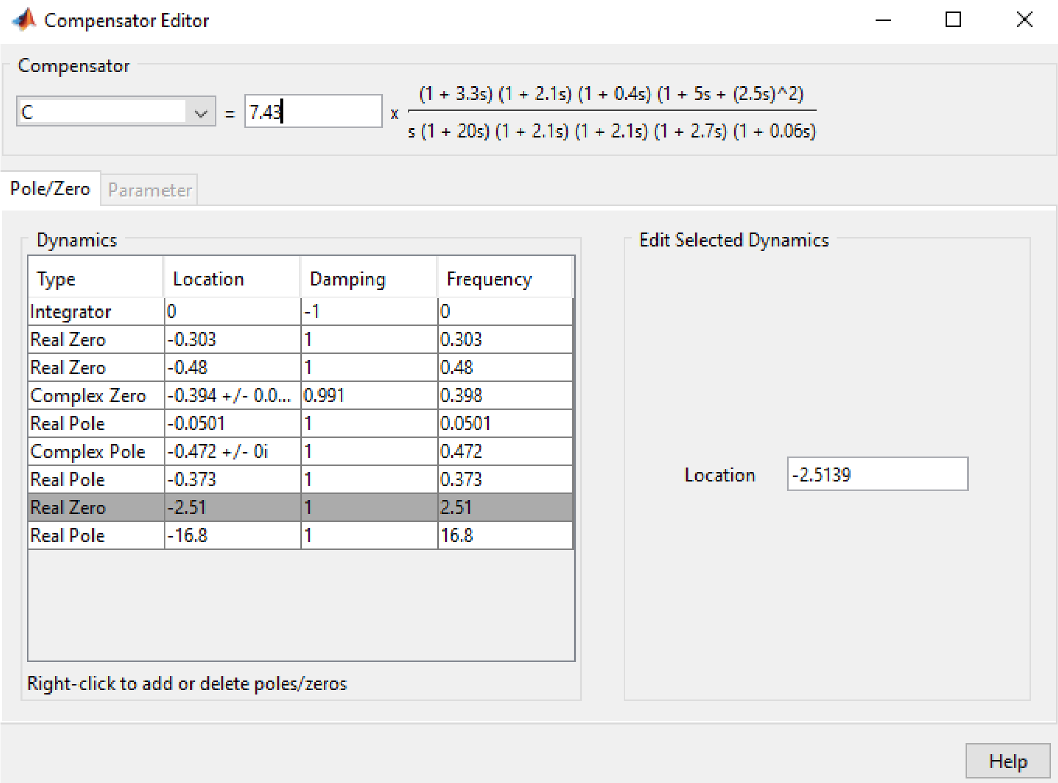


figure 19 – G22 controller equation

Below is the transfer function used for controller design using loop 2.

```
intermediate_L2 =

      5.929 s^6 + 41.03 s^5 + 94.21 s^4 + 83.38 s^3 + 32.28 s^2 + 4.992 s + 0.1518
  ---------------------------------------------------------------------------------
  1.341 s^7 + 13.85 s^6 + 53.88 s^5 + 97.91 s^4 + 86.24 s^3 + 38.85 s^2 + 8.666 s + 0.76

Continuous-time transfer function.
```

3.3.3 MATLAB code and Simulink model for Dynamically Decoupled Control

```
close all
clear all
clc

% Plant transfer functions
P11 = tf(0.76*[1/0.31 1],conv([1/0.42 1],[1/3.23 1]));
P12 = tf(0.53,conv([1/0.42 1],[1/2.64 1]));
P21 = tf(0.66,conv([1/0.42 1],[1/2.93 1]));
P22 = tf(0.66*[1/0.20 1],conv([1/0.42 1],[1/3.19 1]));
Plant = [P11 P12;P21 P22]

figure(1)
bode(Plant)
grid on
title('Bode plot for diagonally dominant CFTAH system')

% Defining G11
num1 = conv(conv(conv([5 1],[2.7 1]),[4.84 4.4 1]),[0.1296 0.72 1])
den1 = conv(conv(conv(conv([1 0],[15 1]),[0.064 1]),[10.89 5.6 1]),[0.5184 1.4 1])
G11 = tf(5.305*num1,den1)
% Defining G22
num2 = conv(conv(conv([3.3 1],[2.1 1]),[0.4 1]),[6.25 5 1])
den2 = conv(conv(conv(conv(conv([1 0],[20 1]),[2.1 1]),[2.1 1]),[2.7 1]),[0.06 1])
G22 = tf(7.43*num2,den2)
% Overall controller matrix
Controller = [G11 0;0 G22]

% Defining GD12
GD12 = -P12/P11
% Defining GD21
GD21 = -P21/P22

% Loop 1 design
intermediate_L1 = P11-(P12*P21/P22)
L1 = intermediate_L1*G11
figure(2);
nichols(L1);
hold on
grid on
% Loop 2 design
intermediate_L2 = P22-(P21*P12/P11)
L2 = intermediate_L2*G22
```

```
nichols(L2);
grid on
hold off

%% Problem 3
%
tfinal = 30;
h = 0.01;
time = 0:h:tfinal;
sim problem3_simulink
figure(3)
subplot(4,1,1)
plot(time,Qsys,'linewidth',2)
title('SAWANT: MECE 7362 - Final Project - Problem 3 Simulink Output')
ylim([0 6])
grid on
ylabel('Systemic flow')
subplot(4,1,2)
plot(time,Qpul,'linewidth',2)
ylim([0 6])
grid on
ylabel('Pulmonary flow')
subplot(4,1,3)
plot(time,VL,'linewidth',2)
ylim([0 6])
grid on
ylabel('Left Pump Voltage')
subplot(4,1,4)
plot(time,VR,'linewidth',2)
ylim([0 5])
grid on
ylabel('Right Pump Voltage')
xlabel('Time')
%
```

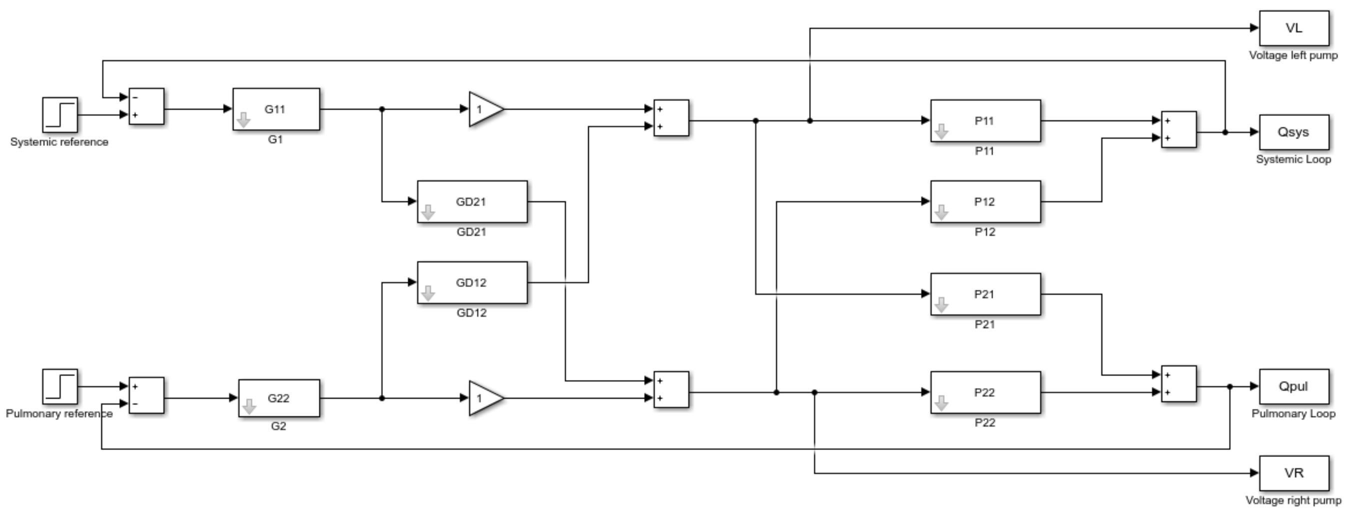


figure 20 – Simulink Model

3.3.4 Results of Dynamically Decoupled Multivariable Control

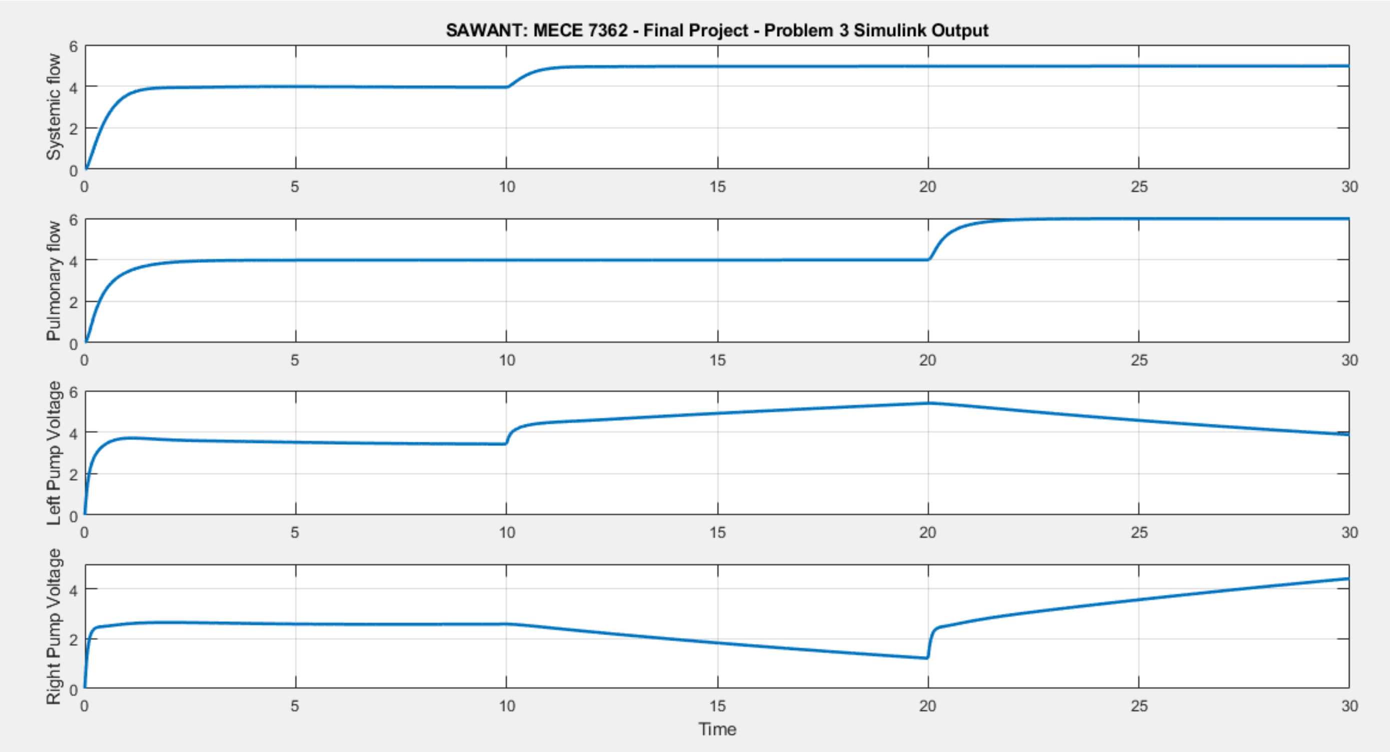


figure 20 – Transient response for DC Decoupled Multivariable Control

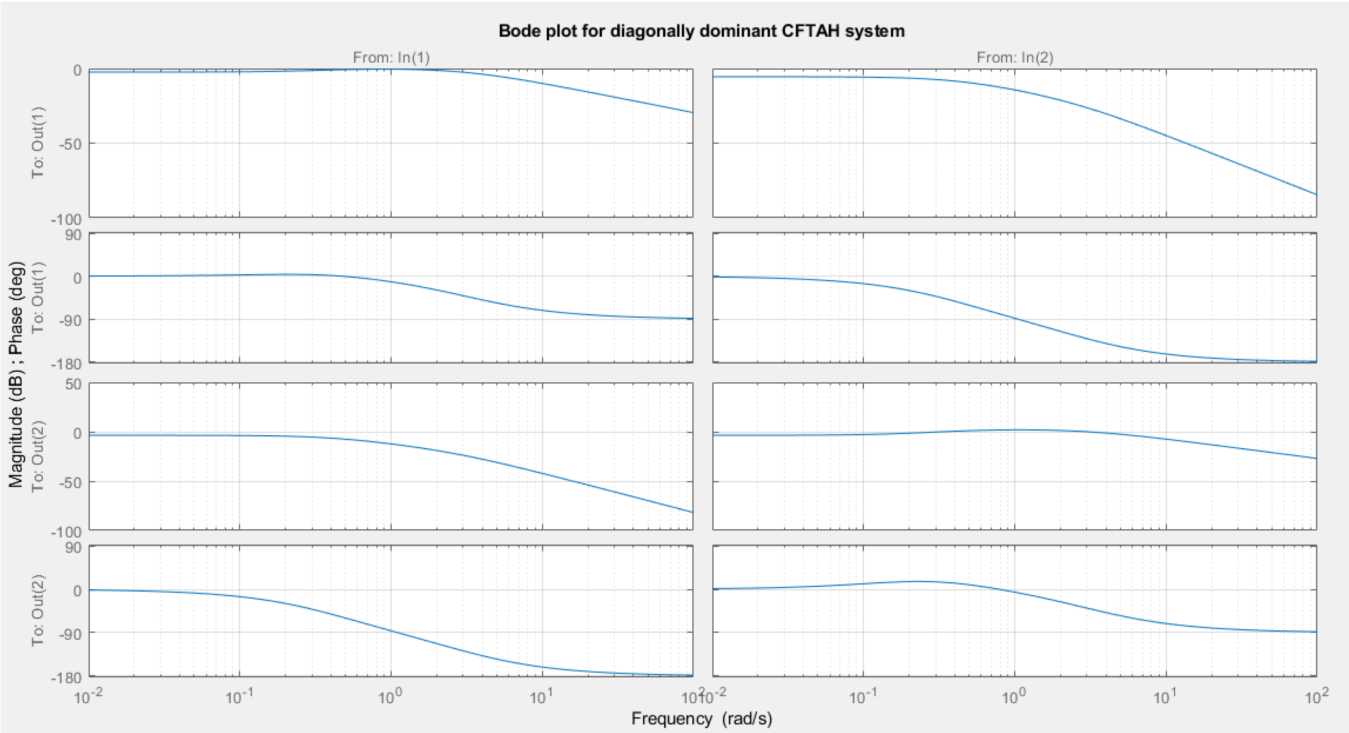


figure 21 – Bode plot for diagonally dominant CFTAH system

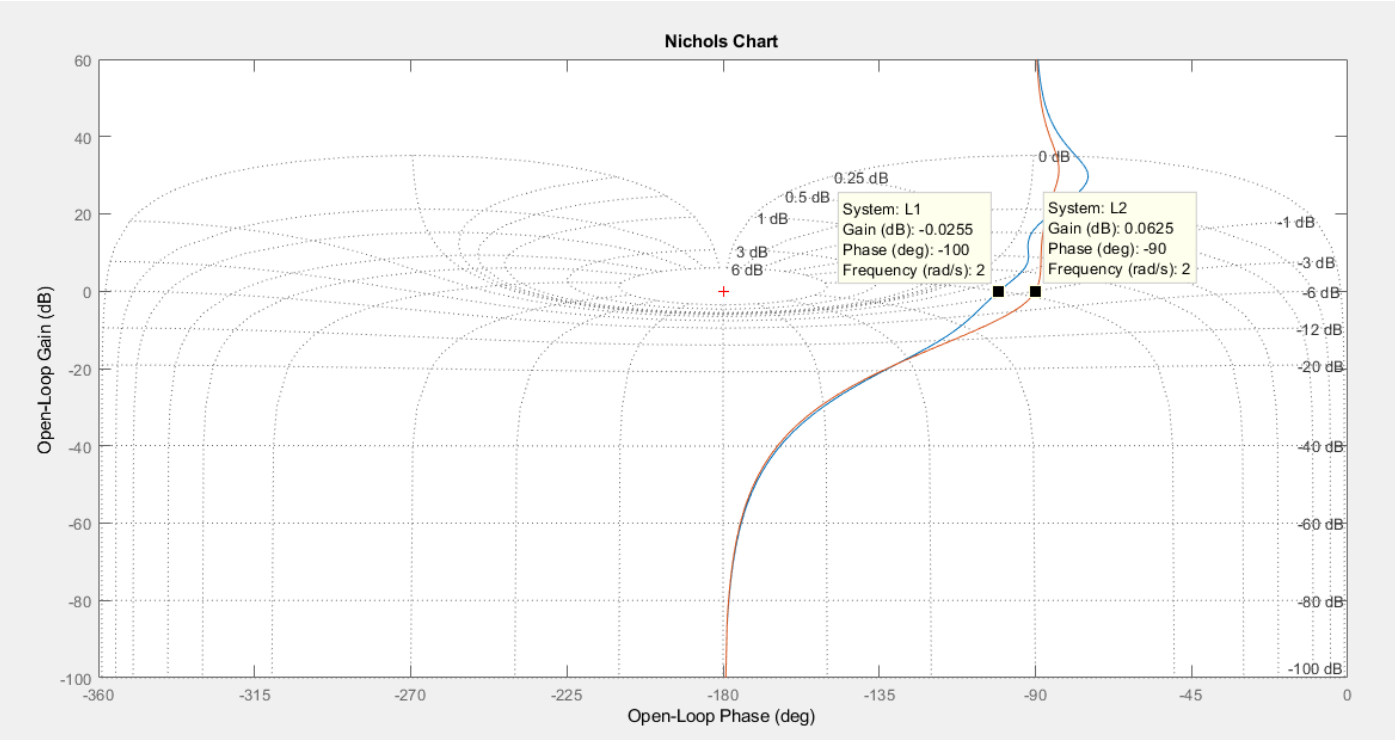


figure 22 – Nichols chart response for loops L1 and L2

4. Results and Discussion

In first problem, we have implemented Diagonally Dominant Multivariable Control. Figure 5 is the response of Simulink output for this method. If we look at the bode plot response for the given MIMO system, amplitude of off-diagonal terms is attenuating rapidly compare to the amplitude response for the diagonal terms. Hence, we can say that this system is diagonally dominant. Hence, to design controllers, we just ignore the off-diagonal terms and produce controllers using diagonal terms.

In second problem, we use static gain matrix obtained by substituting s = 0 in the transfer function to get new plant dynamics. This makes the system statically decoupled and avoid loop interaction. To design the controllers, new plant dynamics are used along with loop shaping technique to achieve strictly proper controller.

Comparing first 2 methods, diagonal controllers are acknowledged for their simplicity and robustness, whereas decoupled controllers may provide good performance but might be non-robust. Decouplers maybe particularly non-robust when applied to ill conditioned system/plant. As we can see in figure 5 and 13, both diagonal dominant and pre-compensated (DC decoupled) controllers provide equally good performance in some aspects but not all. First response goes back to tracking but second response gives little overshoot which is unwanted.

As seen in figure 5, the diagonally dominant controller makes sudden flow increase in the systemic and pulmonary flow. This sudden increase may cause an arterial hypertension. We didn't consider any loop interaction; hence we can see the plateau region after increase in the systemic and pulmonary flow. But instead of all these, diagonal controller is easy in design and more robust.

Whereas figure 13 shows that the considering loop interaction can affect the performances for both the loops i.e. systemic and pulmonary. Having pre-compensated, the blood flow in pulmonary and systemic loop increases gradually and similar events take place in the voltage increase. It turns out that statically decoupled system gives much better performance than diagonal dominant system. Decoupling is susceptible to erroneous or incomplete cross-coupling cancellation due to modelling errors. Hence, appropriate robustness of the design should be guaranteed.

The way statically decoupled system works is the interaction with another loop exist only till it reaches the steady state response, once the previous loop reaches steady state, the interaction does not exist.

5. Conclusion

I have successfully completed the project of designing robust controllers for MIMO systems using 3 different methods. All MATLAB files as well as SIMULINK model works perfectly. Talking about the methods we used, we can say that to achieve better performance we can select statically decoupled system but to achieve more robustness, we can use diagonally dominant controller. Among three control methodologies described above, the dynamically decoupled multivariable control is the most efficient one, because as the system is dynamically decoupled, the loop interactions get cancelled and the input follows the response for only one controller. If we see all three Simulink output plots, then Dynamically Decoupled Multivariable method is more robust and best for the applications like Artificial Heart.

6. Reference

[1] I. Horowitz, Quantitative Feedback Design Theory (QFT), *QFT Publications, Boulder, Colorado, 80303, 1993.*

[2] David Macku, The continuous flow total artificial heart in clinical practice, *Czech Technical University in Prague, Department of Cybernetics, Prague, Czech Republic,*

Na Homocle Hospital, Department of Biomedical Engineering, Prague, Czech Republic

[3] Robust Multivariable Control Lecture Notes by *Dr. Matthew A. Franchek*

[4] Mathworks, MATLAB, *https://www.mathworks.com/*