MTRE 4610 Project
Design of a Controller for a DC Electromotor

Jarrett-Scott K. Jenny & Levi J. Vande Kerkhoff
Lab Professor: Prof. Meiling Sha

I. Scope of Assignment

This project focuses on the modeling and design of a position control system for an electromotor and its associated load system, with electromotors being widely utilized as industrial actuators in robotic systems. The system depicted in the accompanying Figure 1 involves key parameter values (1). The primary objectives include deriving the transfer function and state space model using the phase variable method with voltage (e_a) as the input and the load output shaft (θ_L) as the output, presenting detailed calculations. Additionally, the project entails controller design through the pole placement method to achieve a specific performance criterion of 25% overshoot and a settling time of 1 second. Further components involve the design of an observer with specific pole placements and the implementation of integral control to eradicate steady-state error with defined overshoot and settling time targets. Finally, the Ackermann method is employed to obtain observer gains for the phase variable state space model, facilitating the use of the observer for state variable estimation in conjunction with the controller.

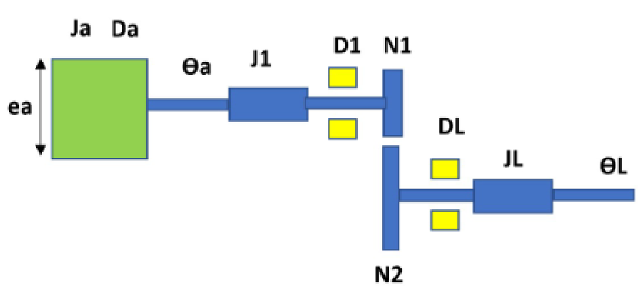

Fig. 1. Schematic of the electromotor.

$$
\begin{gathered}
J_a = 0.6\ kg \cdot m^2 \\
D_a = 0.1\ N \cdot m \cdot sec \\
\frac{K_t}{R_a} = 8\ \mathrm{N \cdot m}/_{A \cdot \Omega} \\
K_b = 0.2\ {rad}/_{V \cdot s} \\
J_1 = 1.4\ kg \cdot m^2 \\
D_1 = 0.4\ N \cdot m \cdot sec \\
N_1 = 50 \\
N_2 = 1250 \\
J_L = 1300\ kg \cdot m^2 \\
D_L = 140\ N \cdot m \cdot sec
\end{gathered}
\tag{1}
$$

Throughout the course of our simulations, we utilized Simulink scopes to create informative plots for our reports, showcasing the block diagram model and all its subblocks within the Simulink environment. We maintained a consistent 2-second period, employing a unit step as the input and adjusting the step time to zero. Each simulation presentation included three key figures: the Simulink block diagram, state variables, and the position of the load shaft. In total, we conducted four simulations, and each one featured three figures (Simulink diagram, State variables, shaft output), resulting in a comprehensive set of 12 figures.

It's worth noting that, in instances where an observer was integrated, we displayed both the observer's estimated state variables and output alongside the actual ones in the same plot. These figures were organized as follows:

- Fig. 1: Simulink block diagram of the "Name of simulation"
- Fig. 2: State variables based on a step of voltage.
- Fig. 3: Load shaft position based on a step of voltage.

The simulations encompassed the following scenarios:

- Simulating the system without integral control.
- Simulating the system with integral control.
- Simulating the response of the system observer, considering the initial condition of the observer model as [2, 2].
- Simulating the response of the system, incorporating both the observer and integral control.

II. Question 1

Obtain the transfer function and the state space model of the system (must use phase variable method based on transfer function) considering that the input is voltage (e_a) and output is load output shaft (θ_L). You must show detail of your calculations.

$$J_T = J_1 + J_a + \frac{N1^2}{N2} \cdot J_L$$
$$= 1.4 + 0.6 + \frac{1}{625}(1300)$$
$$= \frac{102}{25}$$

$$D = D_1 + D_a + \frac{N1^2}{N2} \cdot D_L$$
$$= 0.4 + 0.1 + \frac{1}{625}(140)$$
$$= \frac{181}{250}$$

(2)

$$\varpi_a = \frac{d\theta_a}{dt} \; and \; e_b = k_b \varpi_a$$

$$i_a = \frac{e_a - e_b}{R_a} \rightarrow \frac{e_a - k_b \varpi_a}{R_a}$$

(3)

Calculations for J_T and D_T can be found in data (2). From there we can work on i_a as shown in data set (3). Figure 2 helps show the electrical part of the system. It is from here we can acquire Torque, as seen in data (4).

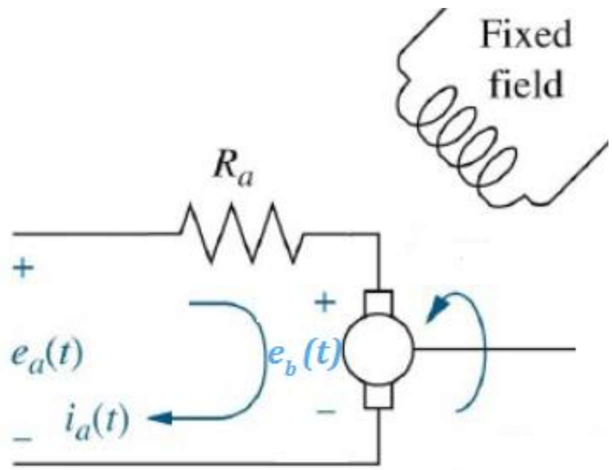

Fig. 2. Electromotor.

$$k_T i_a = Torque \; \rightarrow \; k_T \frac{e_a - k_b \varpi_a}{R_a}$$
$$\rightarrow \frac{k_T}{R_a} e_a - \frac{k_t - k_b}{R_a} \varpi_a$$

$$\Rightarrow 8e_a - 1.6\varpi_a = \frac{102}{25}\frac{d\varpi_a}{dt} + \frac{181}{250}\varpi_a$$
$$\rightarrow \frac{102}{25}\frac{d\varpi_a}{dt} = 8e_a - \frac{581}{250}\varpi_a$$

$$\Rightarrow \frac{d\varpi_a}{dt} = 1.9608 e_a - 0.5696 \varpi_a$$

(4)

With those we were able to derive the state space model of the system and transfer function. This is seen in equations (5) and (6).

$$\frac{d\theta_a}{dt} = \varpi_a$$

$$\frac{d\varpi_a}{dt} = 1.9608e_a - 0.5696\varpi_a$$

$$\theta_L = \frac{1}{25}\theta_a$$

$$\theta_a = x_1, \qquad \varpi_a = x_2, \qquad \theta_L = y$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -0.5696 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1.9608 \end{bmatrix} e_a$$
$$y = [0.04 \quad 0]\, x + [0] \tag{5}$$

$$(1.9608)(0.04) = 0.07843$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -0.5696 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e_a$$
$$y = [0.07843 \quad 0]\, x + [0]$$
$$TF = \frac{0.07843}{s^2 + 0.5696s} \tag{6}$$

III. Question 2

Design a controller using pole placement method for 25% overshoot and 1 second setting time (T_s).

$$\zeta = \frac{-\ln\left(\frac{PO}{100}\right)}{\sqrt{\pi^2 + \ln^2\left(\frac{PO}{100}\right)}} \tag{7}$$

We first must correctly calculate our damping ratio and frequency. Using equation (7) we will get our said damping ratio in data (8), with PO being the Percent Overshoot. From this we can use equation (10) to get the frequency for 1 second setting time.

$$\zeta = \frac{-\ln\left(\frac{25}{100}\right)}{\sqrt{\pi^2 + ln^2\left(\frac{25}{100}\right)}} = 0.4037 \tag{8}$$

$$w_n = \frac{4}{\zeta T_s} = \frac{4}{(0.4037)(1)} = 9.9083 \tag{9}$$

With this information we can get our first characteristic equation using formula (10). This calculation is then compared to our second characteristic equation. To get the second characteristic equation we harness equation (11) which utilizes our state space model in (6).

$$CE_1 = s^2 + 2\zeta w_n + w_n^2$$
$$\Rightarrow s^2 + 8s + 98.1754 \tag{10}$$

$$CE_2 = \det\left(SI - (A_C)\right)$$
$$\to \det\left(SI - (A - Bk)\right)$$

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & -0.5696 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} [k_1 \quad k_2]$$
$$\to \begin{bmatrix} 0 & 1 \\ -k_1 & -0.5696 - k_2 \end{bmatrix}$$

$$SI - A_c = \begin{bmatrix} s & -1 \\ k_1 & s + 0.5696 + k_2 \end{bmatrix}$$

$$\det(SI - A_c) = s^2 + (0.5696 + k_2)s + k_1 \tag{11}$$

Finally, we compare the first and second characteristic equations. From this we get our K and ultimately the controller. Data (12) shows this information.

$$0.5696 + k_2 = 8 \rightarrow k_2 = 7.4304$$
$$k_1 = 98.1754$$
$$k = [98.1754 \quad 7.4304]$$
$$A_c = \begin{bmatrix} 0 & 1 \\ -98.1754 & -8 \end{bmatrix} \tag{12}$$

IV. Question 3

Design an observer with its poles 10 times further away from imaginary axis (You much use observer canonical format here).

$$\dot{x} = \begin{bmatrix} -0.5696 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0.07843 \end{bmatrix} e_a$$
$$y = [1 \quad 0]\, x + [0]$$
$$P_1, P_2 = \zeta w_n \pm w_n\sqrt{1-\zeta^2}$$
$$\longrightarrow -4 \pm j9.065$$
$$\times\, 10 = -40 \pm j90.65 \tag{13}$$

First, we will form an observer canonical model and calculate the poles with equations (13), then multiple it by a factor of ten to satisfy the inquiry.

$$CE_1 = (s+40)^2 - (90.65j^2 \Rightarrow s^2 + 80s + 9817.4225 \tag{14}$$

Using these new pole positions we will form our first characteristic equation shown in data (14). We will then follow formula (15) and form our second characteristic equation to compare the first too.

$$CE_2 = \det(SI - (A_{ob})) \rightarrow \det(SI - (A_o - LC_o))$$
$$A_{ob} = \begin{bmatrix} -0.5696 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} [1 \quad 0] \rightarrow \begin{bmatrix} -(0.5696 - L_1) & 1 \\ -(L_2) & 0 \end{bmatrix}$$
$$SI - A_{ob} = \begin{bmatrix} s + (0.5696 + L_1) & -1 \\ L_2 & s \end{bmatrix}$$
$$\det(SI - A_c) = s^2 + (0.5696 + L_1)s + L_2 \tag{15}$$

We can see in data (16) this comparison and the resulting L value along with the observer matrix.

$$0.5696 + L_1 = 80 \rightarrow L_1 = 79.4304$$
$$L_2 = 9817.4225$$
$$L = \begin{bmatrix} 79.4304 \\ 9817.4225 \end{bmatrix}$$
$$A_{ob} = \begin{bmatrix} -80 & 1 \\ -9817.4225 & 0 \end{bmatrix} \tag{16}$$

V. Question 4

Design an integral control to eliminate the steady state error with the overshoot of 25%, and the setting time of one second.

Once again starting with our damping ratio and natural frequency found in (8) and (9), respectfully, we form our first characteristic equation, but this time add a third pole a distance away from the others, at -40. Data (17) shows this breakdown and the resulting equation.

$$CE_1 = (s+40)(s^2 + 8s + 98.1754) \Rightarrow s^3 + 48s^2 + 418.1754s + 3927.016 \tag{17}$$

Forming out integral controller is a process shown in equations (18). This leads to our second characteristic equation.

$$CE_2 = \det(SI - (A_{IC})) \rightarrow \det\left(SI - \left(\begin{bmatrix} (A - Bk & BK_e \\ -C & 0 \end{bmatrix}\right)\right)$$

$$A_{IC} = \begin{bmatrix} A - Bk & BK_e \\ -C & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & 0 \\ -k_1 & -0.5696 - k_2 & K_e \\ -0.07483 & 0 & 0 \end{bmatrix}$$

$$SI - A_{IC} = \begin{bmatrix} s & -1 & 0 \\ k_1 & s + (0.5696 + k_2) & -K_e \\ -0.07483 & 0 & s \end{bmatrix}$$

$$\det(SI - A_{IC}) = s^3 + (k_2 + 0.5696)s^2 + k_1 s + 0.07843 K_e$$

(18)

$$k_2 + 0.5696 = 48 \rightarrow k_2 = 47.4304$$

$$k_1 = 418.1754$$

$$0.07483 K_e = 3927.016 \rightarrow K_e = 52479.16611$$

$$K = [418.1754 \quad 47.4304]$$

$$K_e = 52479.16611$$

$$A_{IC} = \begin{bmatrix} 0 & 1 & 0 \\ -418.1754 & -48 & 52479.16611 \\ -0.07483 & 0 & 0 \end{bmatrix}$$

(19)

Comparing the two equations yields new K values and a K_e, which in turn eliminates our steady state error, as well as our resulting integral controller, seen in (19). We can check this with formula (20).

$$e_{ss} = 1 - C_{IC} A_{IC}^{-1} B_{IC} \Rightarrow 1 - [0.07483 \quad 0 \quad 0] \begin{bmatrix} 0 & 1 & 0 \\ -418.1754 & -48 & 52479.16611 \\ -0.07483 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$$

(20)

VI. Question 5

Using Ackermann method obtain the observer gains for the phase variable state space model. Then use the observer to estimate the state variables for the controller.

Our first characteristic equation can be found in data point (10). To get our second we need to have our form be in cascade. (21) shows us how we converted our transfer function to this form and the matrix we got from it.

$$\frac{0.07843}{s^2 + 0.5696a} \rightarrow \frac{1}{s + 0} \cdot \frac{1}{s + 0.5696} \cdot 0.07843 \rightarrow \frac{0.07843}{(s + 0)(s + 0.5696)}$$

$$\dot{x} = \begin{bmatrix} -0.5696 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e_a$$

$$y = [0.07843 \quad 0]\, x + [0]$$

(21)

From her we followed the steps outlined in (22) to achieve our second CE. We then compared the two. As seen in (23). These yields use our cascade control matrix as well as our cascade K values. It is with these we then are able to derive our cascade L values.

$$CE_2 = \det(SI - (A_X)) \rightarrow \det(SI - (A_{ca} - B_{ca}k))$$

$$A_Z = \begin{bmatrix} -0.5696 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \rightarrow \begin{bmatrix} -0.5696 & 1 \\ -(k_1) & -(k_2) \end{bmatrix}$$

$$SI - A_Z = \begin{bmatrix} s + 0.5696 & 1 \\ k_1 & k_2 \end{bmatrix}$$

$$\det(SI - A_Z) = s^2 + (0.5696 + k_2)s + 0.5696k_2 + k_1$$

(22)

$$0.5696 + k_2 = 8 \rightarrow k_2 = 7.4304$$

$$0.5696k_2 + k_1 = 93.943$$

$$K_Z = \begin{bmatrix} 93.943 & 7.4304 \end{bmatrix}$$

$$A_Z = \begin{bmatrix} -0.5696 & 1 \\ -93.943 & -7.4304 \end{bmatrix}$$

(23)

Ackermann's formula stands as a control system design approach devised by Jürgen Ackermann to address the pole allocation problem for invariant-time systems. In control system design, a key challenge involves crafting controllers capable of altering a system's dynamics by manipulating the eigenvalues of the matrix that characterizes the closed-loop system's dynamics. This process is tantamount to modifying the poles of the associated transfer function, particularly when there is no cancellation of poles and zeros.

With the acquisition of K_Z we can obtain $L_Z$by following the equations in (24). Going through these conversion and algebraic steps we are able to derive L_Z from our previously obtained systems. Ideally, we could have done the same with K_Z but took the “long way” to show proof of concept.

$$L_Z = PL$$

$$P = C_{MZ}C_{MV}^{-1}$$

$$C_{MZ} = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$C_{MV} = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -0.5696 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 \\ 0.5696 & 1 \end{bmatrix}$$

$$P_{LV} = \begin{bmatrix} 79.4304 \\ 9862.66605 \end{bmatrix} = L_Z$$

(24)

VII. Simulink Figures: Simulation 1

The following section shows the results of the MATLAB Simulink simulations for the system with only controller gains. The figures below either contain the block diagram of the system itself or a scope output. The simulation time for each is 2 seconds.

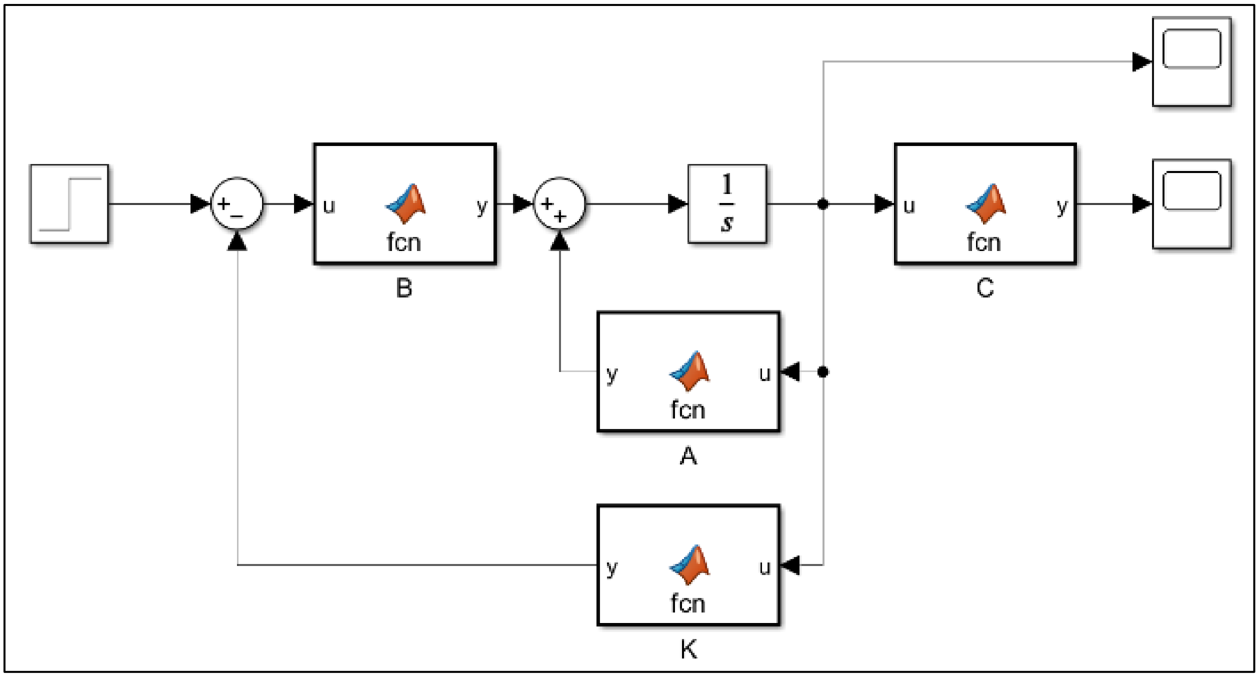

Fig. 3. Simulink Block Model

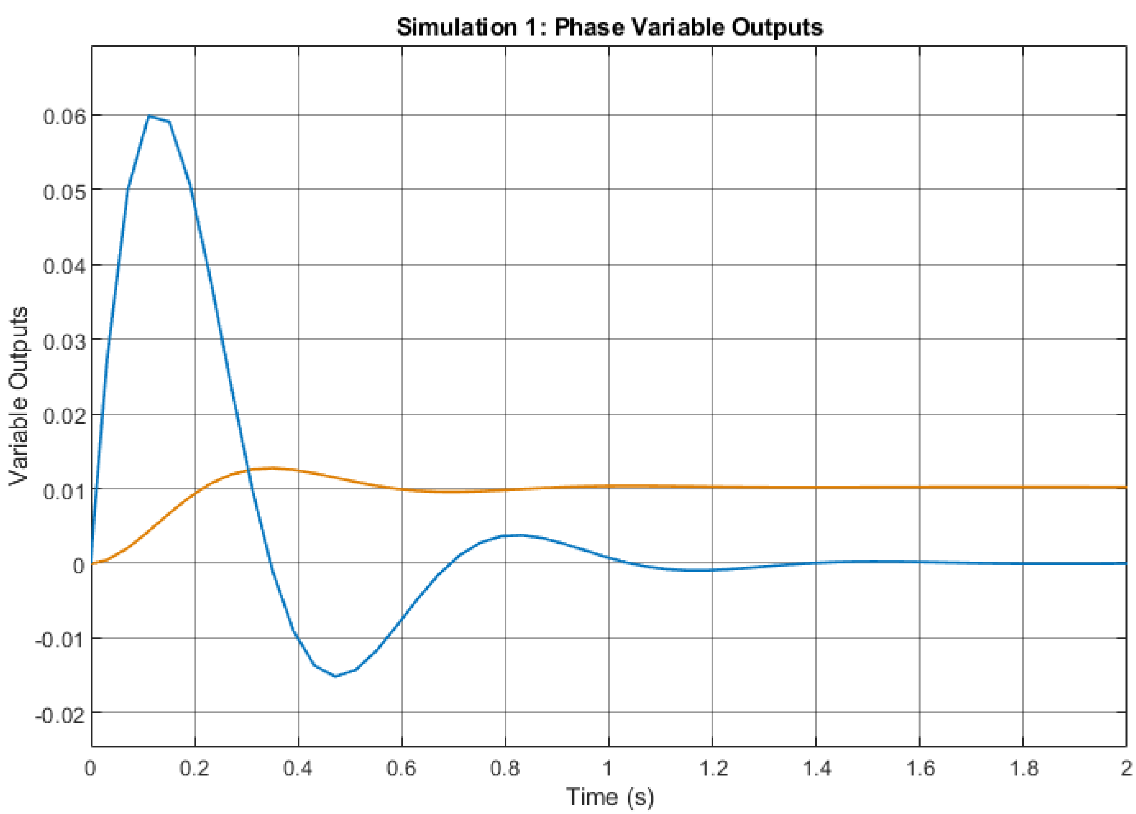

Fig. 4. Phase Variable Outputs

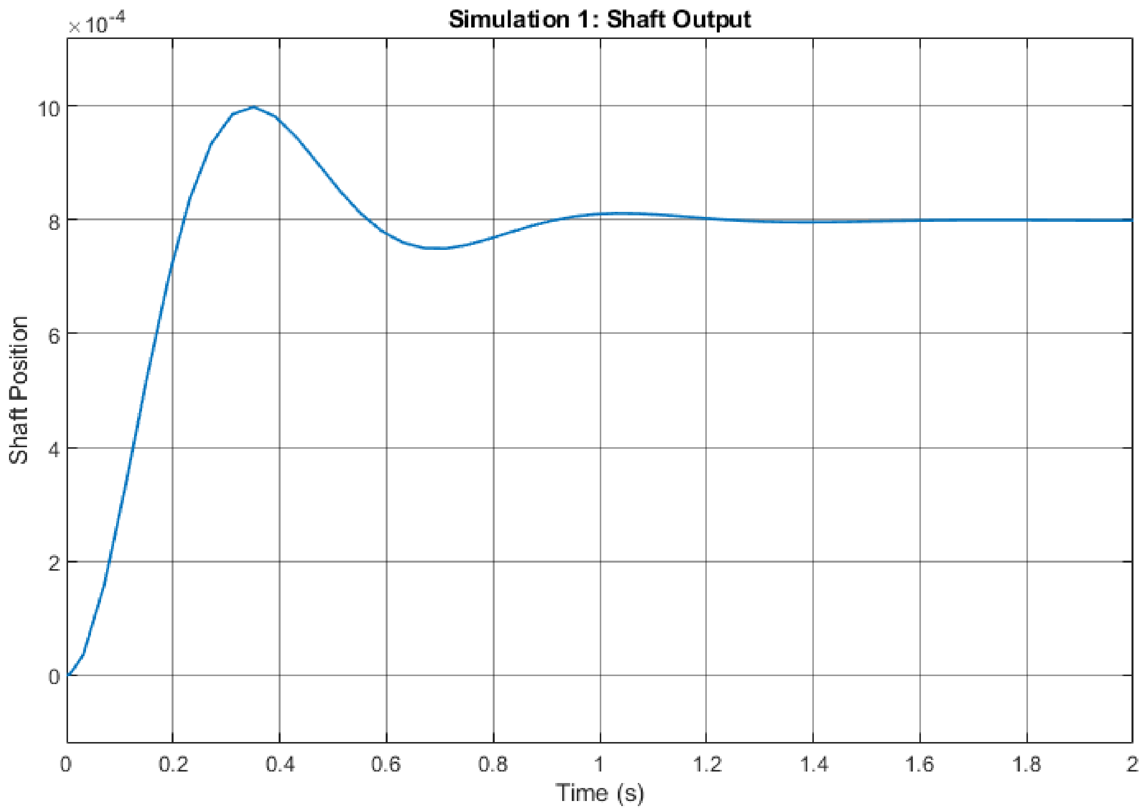

Fig. 5. Plant Shaft Position

IIX. Simulink Figures: Simulation 2

The following section shows the results of the MATLAB Simulink simulations for the system with controller gains and integral control. The figures below either contain the block diagram of the system itself or a scope output. The simulation time for each is 2 seconds.

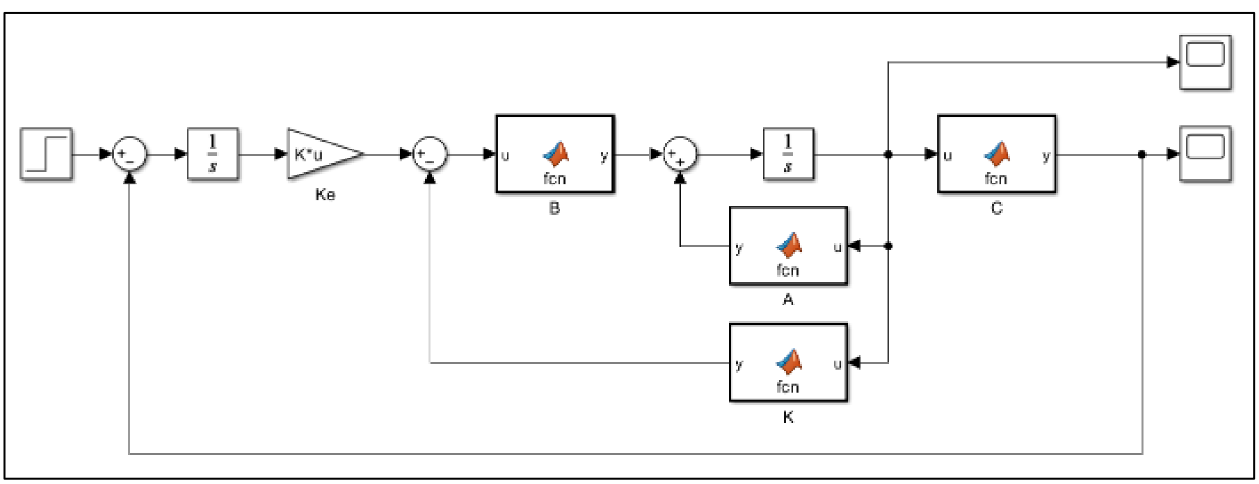

Fig. 6. Simulink Block Model

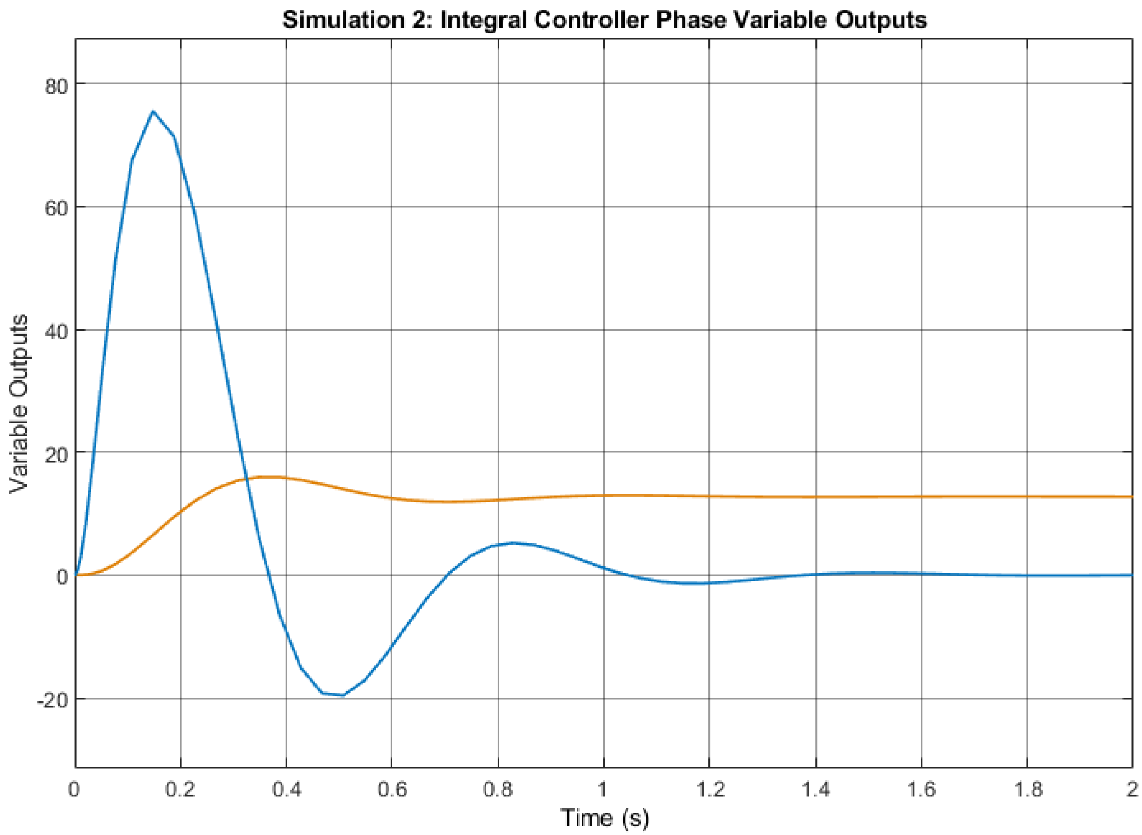

Fig. 7. Phase Variable Outputs

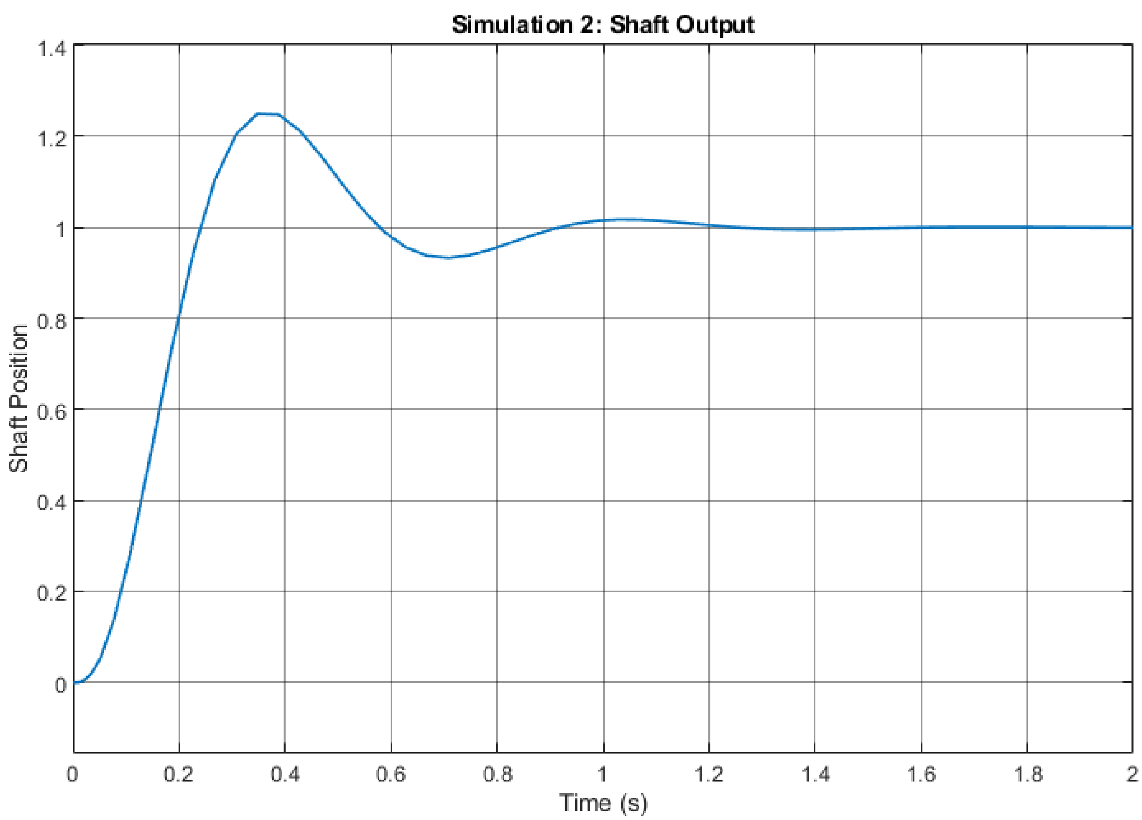

Fig. 8. Plant Shaft Position

IX. Simulink Figures: Simulation 3

The following section shows the results of the MATLAB Simulink simulations for the system with controller gains and an identical observer system with poles 10 times further from the imaginary axis. The figures below either contain the block diagram of the system itself or a scope output. The simulation time for each is 2 seconds. Figure 11 shows an experiment that was a bit beyond the scope of this project; the following question was raised: what happens when the observer system is *not* in the identical form of the plant (i.e., if the plant is in phase variable form and the observer is in canonical form)? The figures following Figure 11 show the output of the non-identical observer system compared to the original experiment. The simulation time for this experiment was increased to 15 seconds so that the steady state of the non-identical observer system could be observed. The results of this experiment will be discussed in the conclusion.

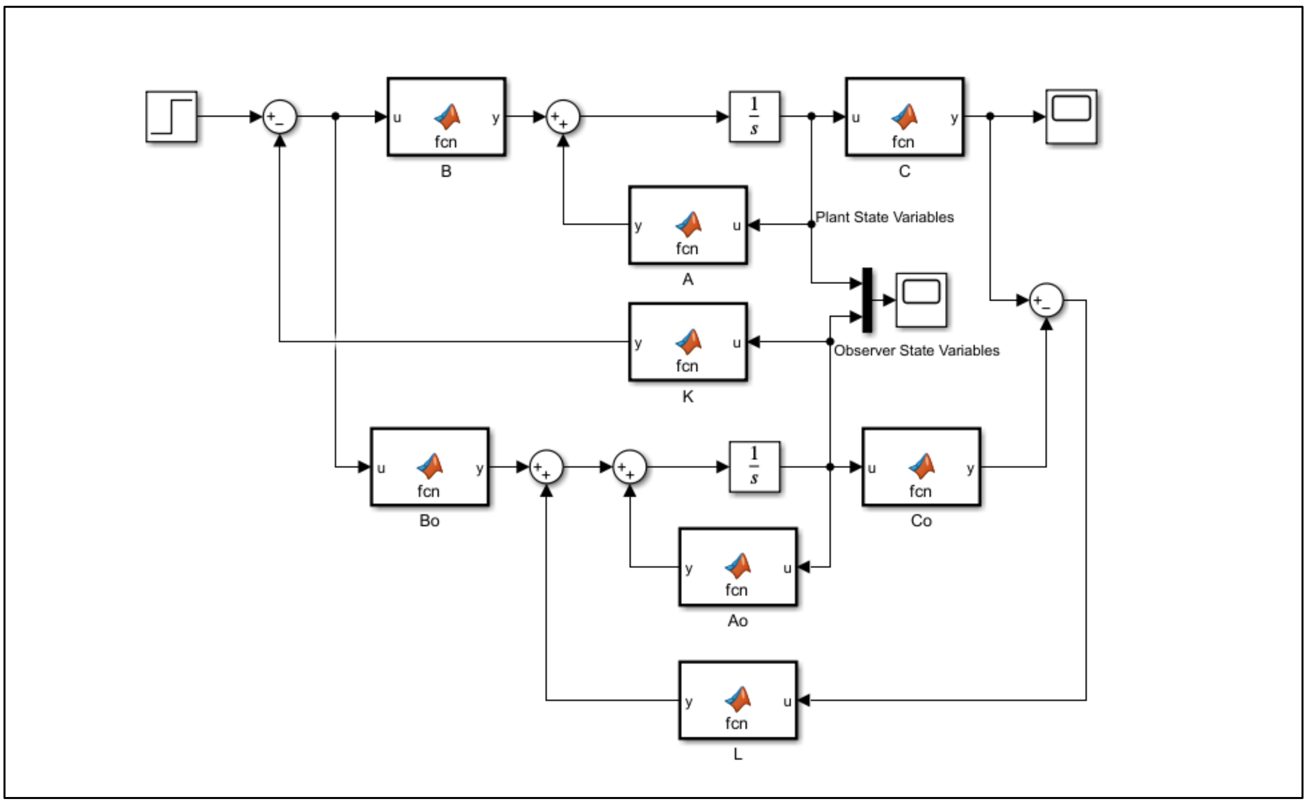

Fig. 9. Simulink Block Diagram

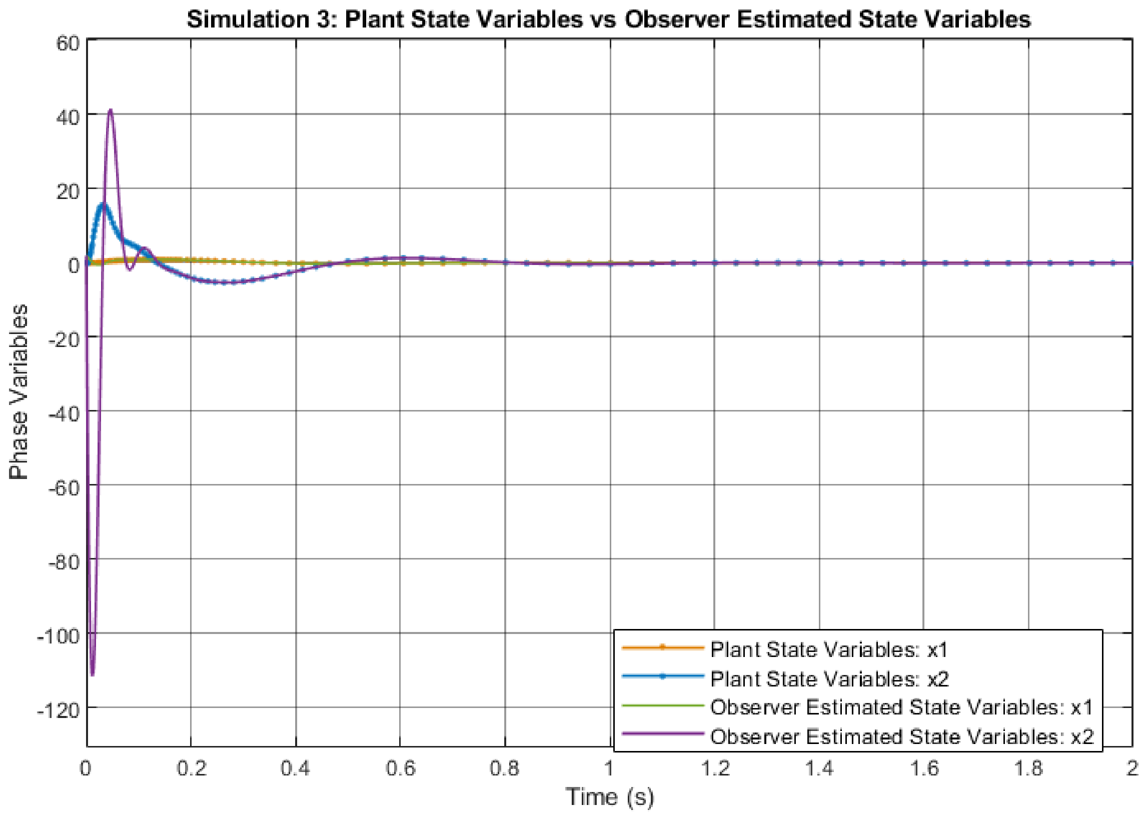

Fig. 10. Plant State Variables vs Observer Estimated State Variables

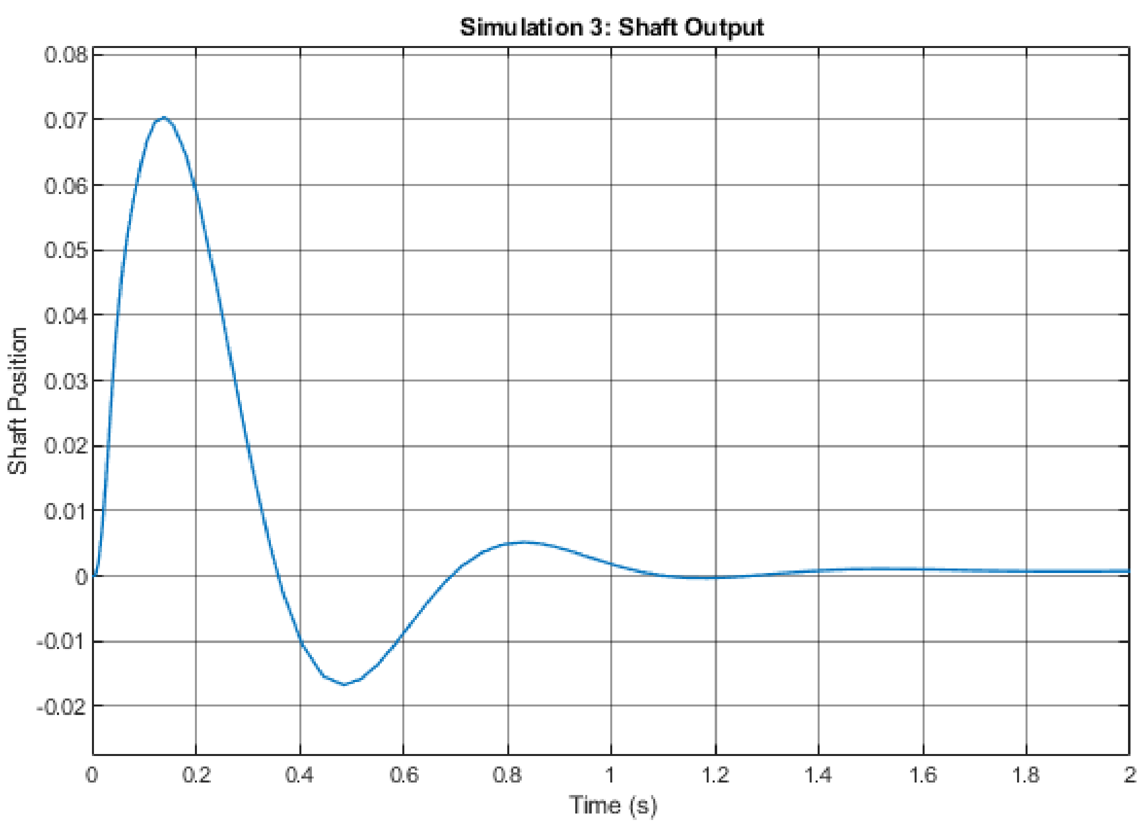

Fig. 11. Plant Shaft Position

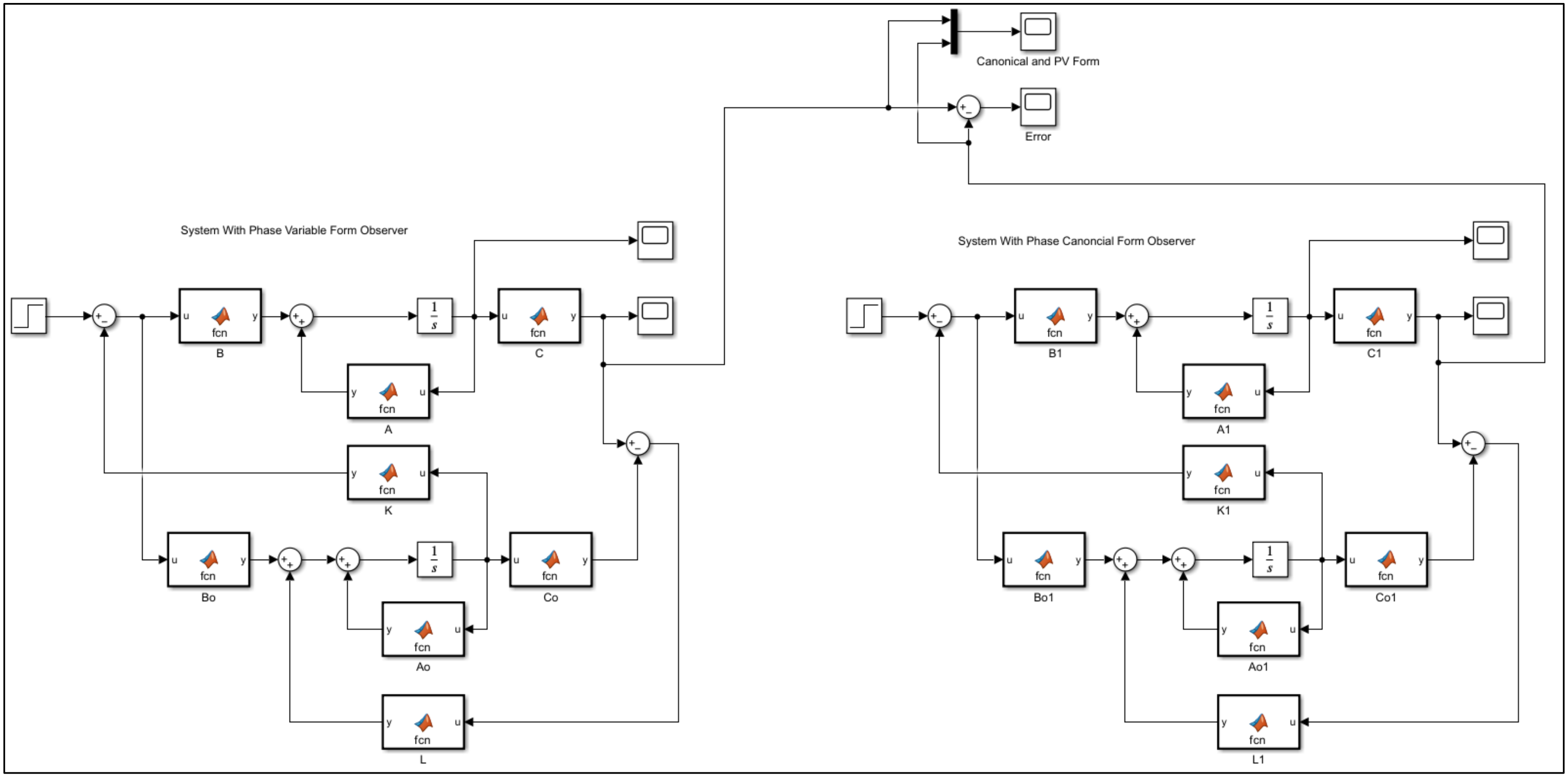

Fig. 12. Simulink Block Model of Compared Observer Forms

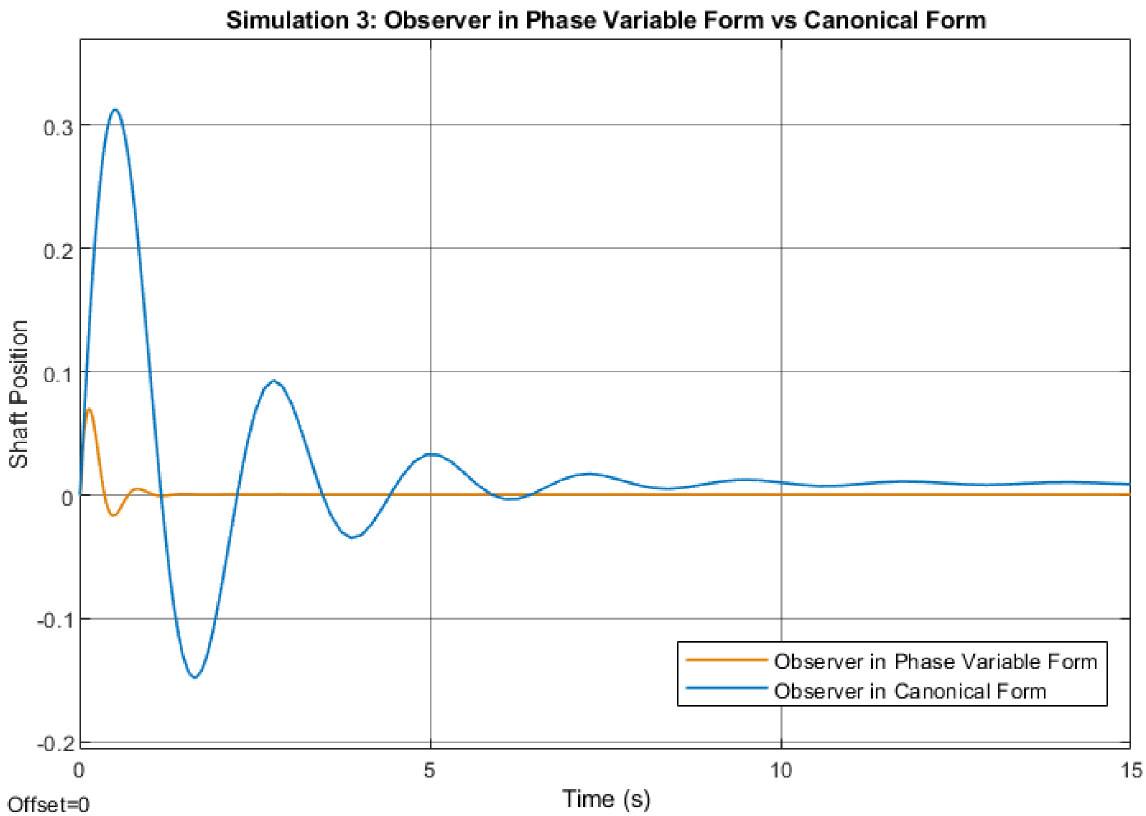

Fig. 13. Comparison of Observer Forms' Outputs

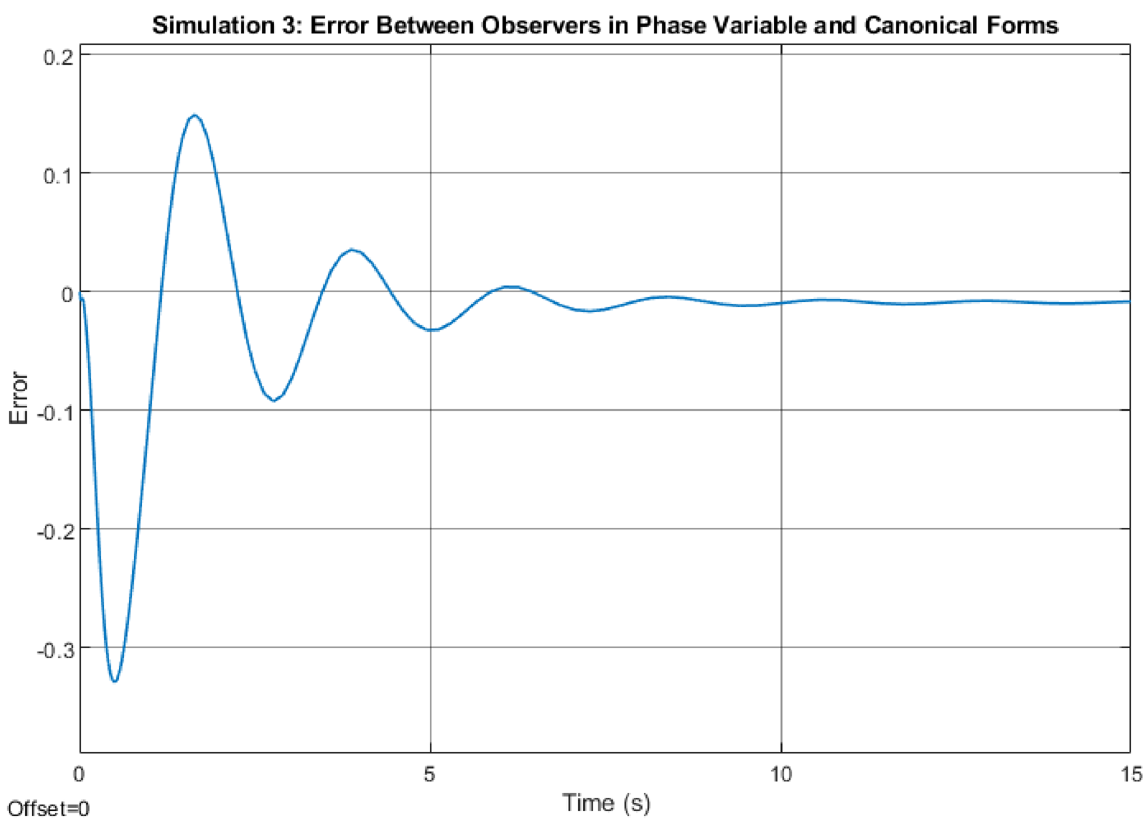

Fig. 14. Error Between Observers in Phase Variable Form and Canonical Form

X. Simulink Figures: Simulation 4

The following section shows the results of the MATLAB Simulink simulations for the system with controller gains, integral control, and an identical observer system with poles 10 times further from the imaginary axis. The figures below either contain the block diagram of the system itself or a scope output. The simulation time for each is 2 seconds. Figure 17 shows another experiment in kind to the last that was a bit beyond the scope of this project; the same question was raised: what happens when the observer system is *not* in the identical form of the plant? The figures following Figure 17 show the output of the non-identical observer system compared to the original experiment. The results of this experiment will be discussed in the conclusion.

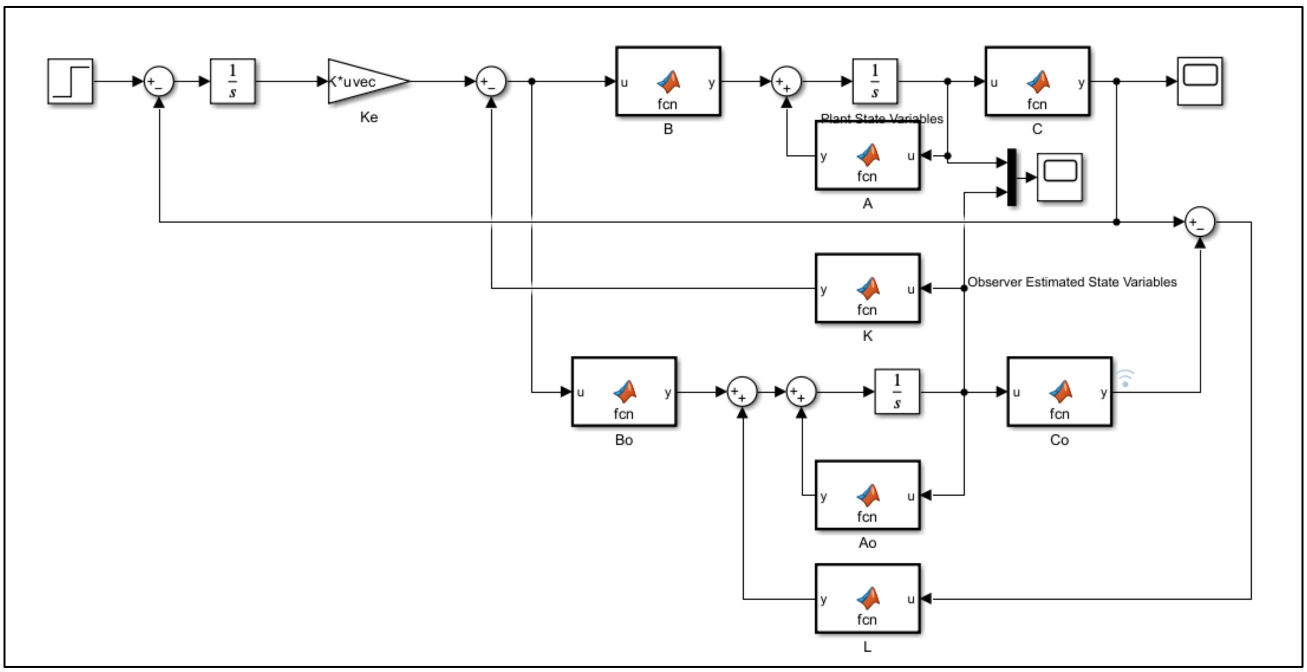

Fig. 15. Simulink Block Model of Integral Control and Observer

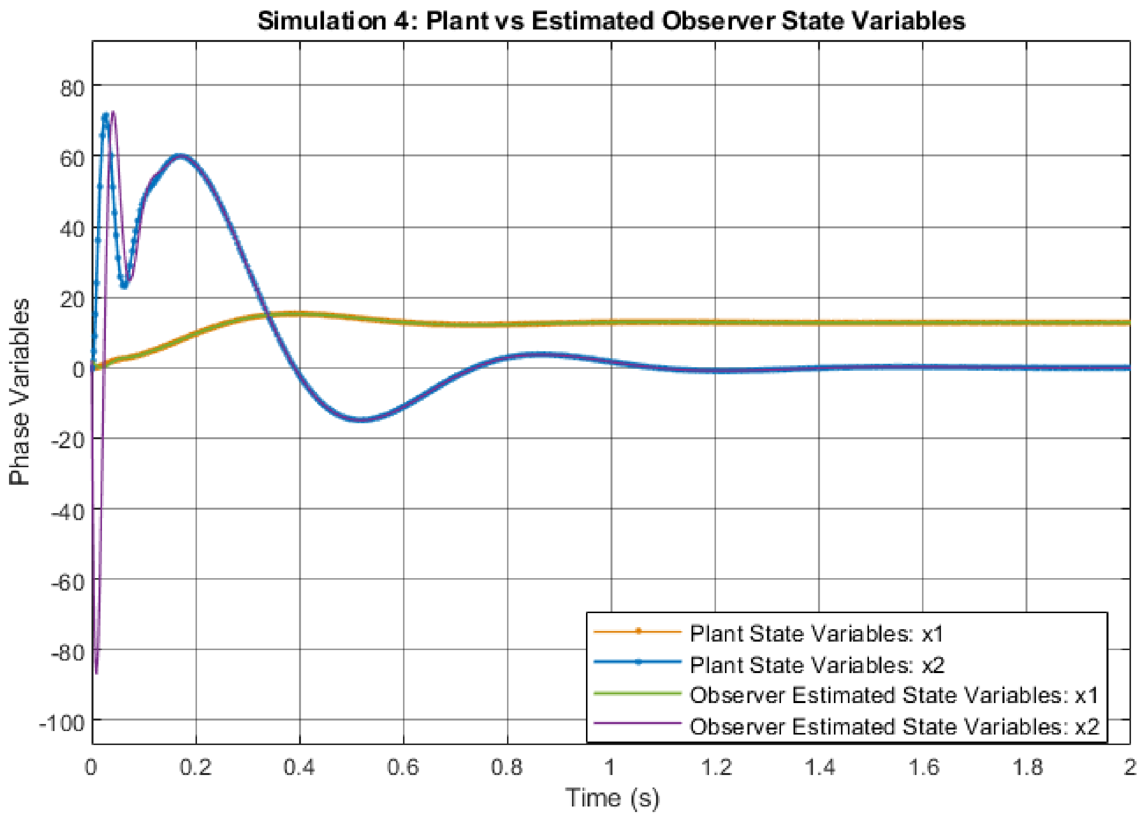

Fig. 16. Plant vs Estimated Observer State Variables

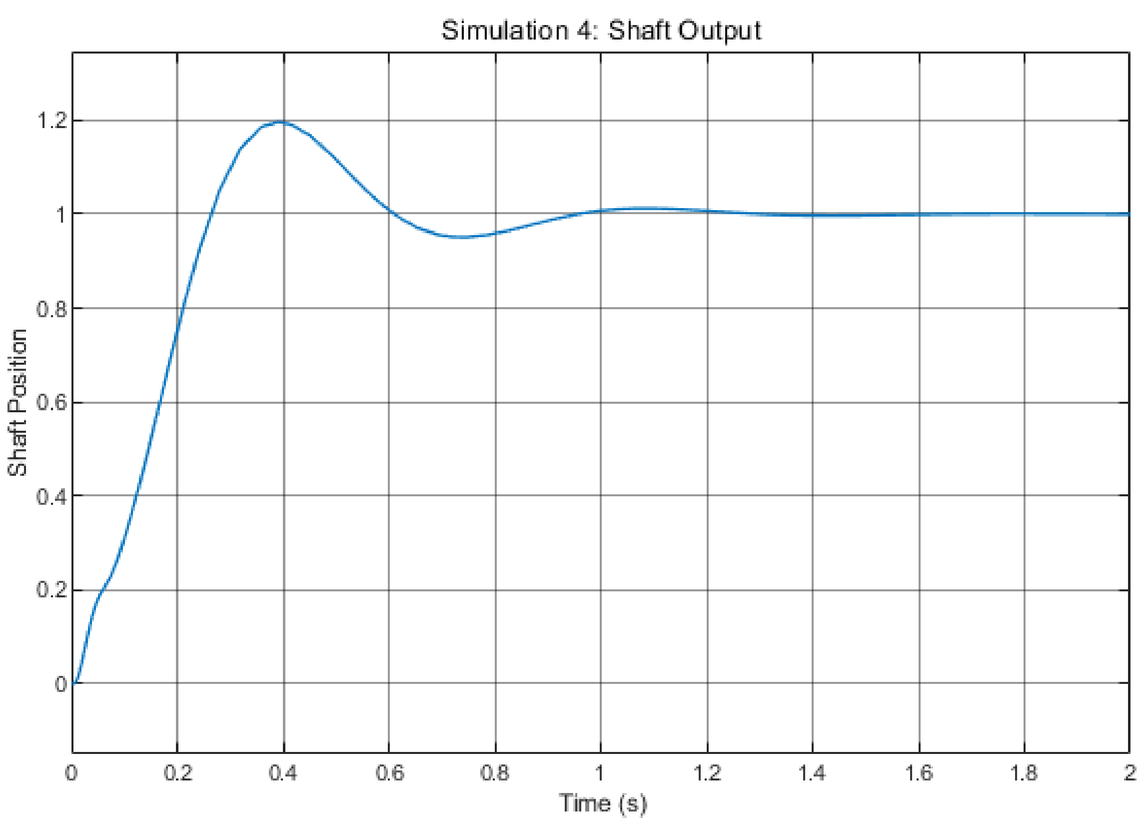

Fig. 17. Shaft Output

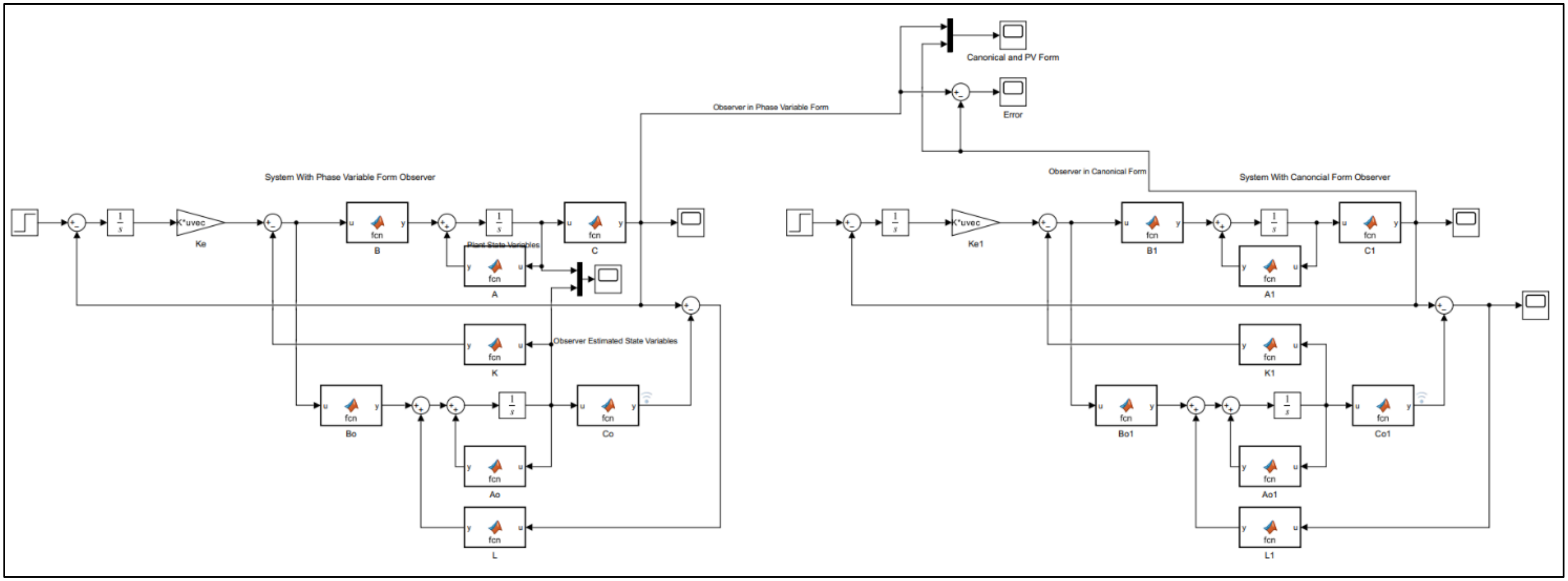

Fig. 18. Simulink Block Model of Compared Observer Forms

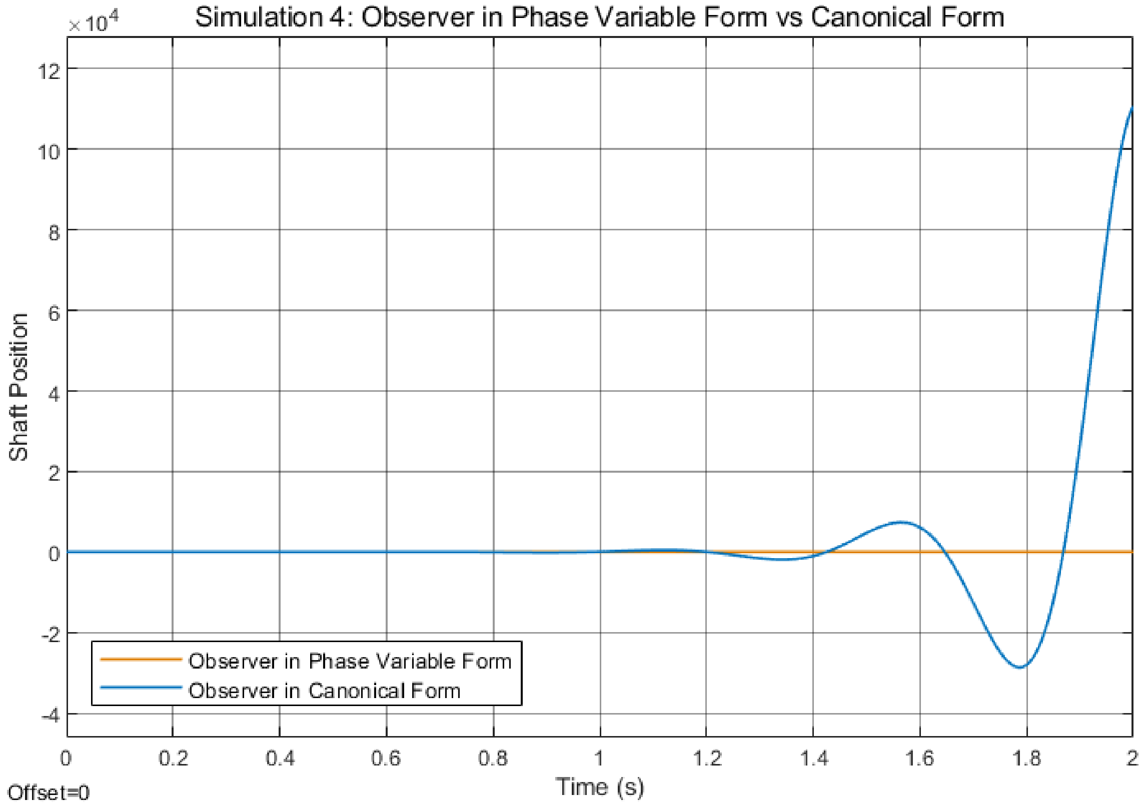

Fig. 19. Comparison of Observer Forms' Outputs

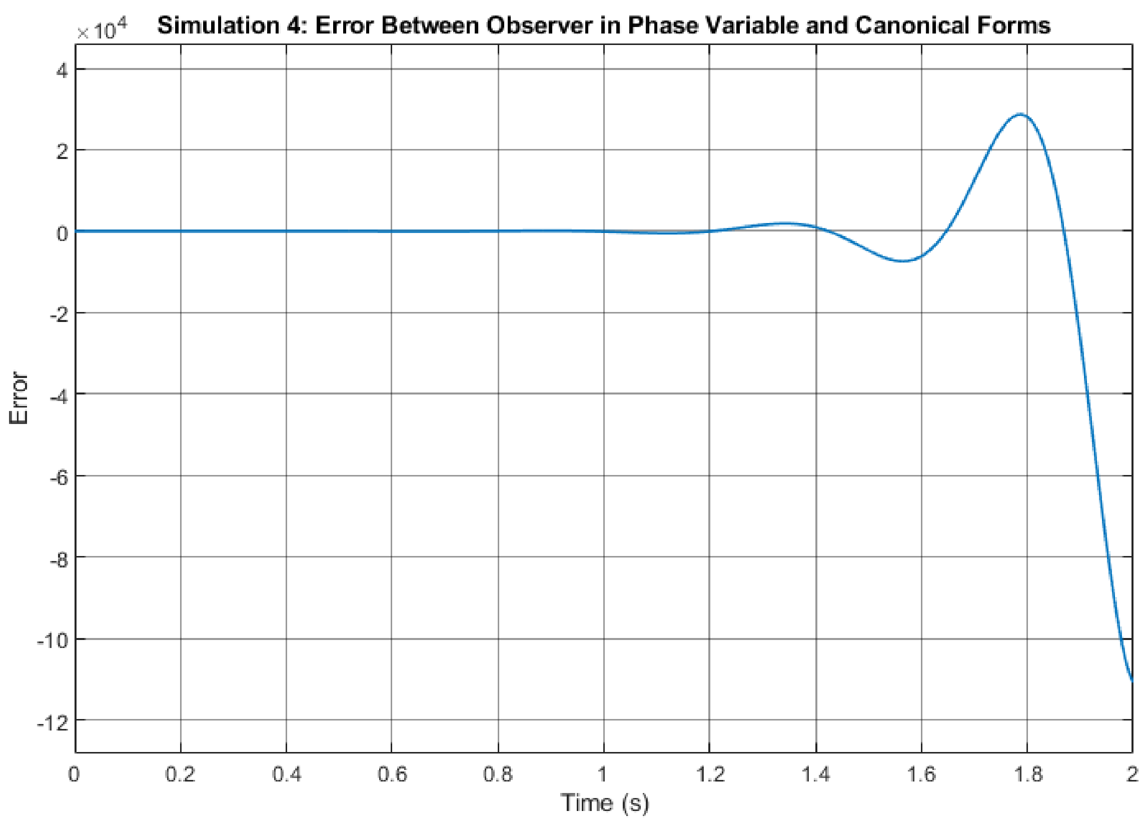

Fig. 20. Error of Canonical Form Compared to Phase Variable Form

VIII. Data Analysis & Conclusion

The stepinfo function in MATLAB was used to gather the data in the following tables for each simulation. This can be seen in the MATLAB code at the end of the report.

Simulation 1		Simulation 2	
Rise Time (s)	0.152	Rise Time (s)	0.157
Settling Time (s)	0.849	Settling Time (s)	0.874
Overshoot	24.99%	Overshoot	23.95%
Peak	9.98E-04	Peak	1.239
Peak Time (s)	0.351	Peak Time (s)	0.388
Steady State Error	99.90%	Steady State Error	0.01%

Simulation 3		Simulation 4	
Rise Time (s)	0.004	Rise Time (s)	0.205
Settling Time (s)	1.964	Settling Time (s)	0.878
Overshoot	9054.09%	Overshoot	19.66%
Peak	0.070	Peak	1.197
Peak Time (s)	0.139	Peak Time (s)	0.391
Steady State Error	99.90%	Steady State Error	0.00%

The results of the first simulation show that for this particular system, there is a large steady state error when an integral controller is absent from the system. The steady state error calculated from the simulation data was found to be 99.9%, or alternatively stated as close to zero; the unit step input, or the desired output, was 1 while the actual output was 7.98×10^{-4}. The desired overshoot was acquired and was estimated from the simulation to be 25%.

The results of the second simulation when compared to the first and third show that integral control is absolutely necessary for this system to have an actual output that is accurate to your desired output. A third pole at -40, exactly ten times further from the imaginary axis than the other two poles' real components, yielded an overshoot of 23.9% and a steady state error of 0%. A more accurate third pole placement was calculated numerically in MATLAB and was exactly 10.389 times further form the imaginary axis than the real components of the other two poles. This yielded an overshoot of exactly 25% and a steady state error of 0% as well.

The results of the third simulation showed the helpfulness of adding an observer to a multiple input multiple output system. This simulation also showed that an observer is not a replacement for integral control but is only included to monitor the dynamics of such a system; this was proven by the faster peak time of 0.139s compared to 0.351s and the faster rise time of 0.004s compared to 0.152s. The error in this system was much like that of the first simulation. The steady state error was 99.9%, or instead stated as close to zero; the desired output was 1 and the actual output was 7.7×10^{-4}.

The fourth simulation combined the effectiveness of the integral controller and the observer and yielded the following results. The steady state error of the system was 0% with an overshoot of roughly 20%. This combination allows the controls engineer to observe the condition of otherwise unknown state variables and to mitigate the steady state error of the system. The addition of the integral controller also greatly reduced the overshoot from over 9000% to the aforementioned 20%.

Although it was outside of the assigned scope of this project, the pairing of non-identical plant systems and observer systems were also evaluated. The plant system in each simulation was kept in phase variable form. The observer, however, was altered from a matching phase variable form to a canonical form. The gains in these experiments were kept the same. These experiments can be seen in Sections IX and X in Figures 12-14 and 18-20. Simulation 3 explored this pairing without an accompanying integral controller. This experiment proved that the observer in non-identical form provided a steady state output that was closer to the desired output but seemed to be much less stable than the original system. This system also required much more time to reach its steady state, further proving its more volatile nature. The hypothesis for the second experiment was that the addition of an integral controller would improve the behavior of the system solely on the basis of its ability to handle the accumulation of error over time. As can be seen in Figures 19 and 20, this was not the case. On the contrary, the system was unstable, and its output very quickly became unbounded. The results of these experiments prove that the form of the observer needs to be identical to that of the plant.

MATLAB Code Used to Verify Work Done By Hand

```
clc;
% original state space model
A = [0, 1; 0, -0.5696];
B = [0; 1];
C = [0.07843, 0];
D = 0;

% ------------------------- Calculating the Controller Gains ------------------------
PercentOvershoot = 0.25;
SettlingTime = 1;
DampingRatio = (-log(PercentOvershoot)/sqrt(3.14159^2+log(PercentOvershoot)^2))
NaturalFrequency = 4/(DampingRatio*SettlingTime)

syms k1 k2 k3 s
multiplier = 1;
pole_1 = multiplier*(-DampingRatio*NaturalFrequency) + 1i*NaturalFrequency*sqrt(1-
DampingRatio^2);
pole_2 = multiplier*(-DampingRatio*NaturalFrequency) - 1i*NaturalFrequency*sqrt(1-
DampingRatio^2);
CE_1 = ((s - pole_1)*(s - pole_2));
CE_1 = vpa(expand(CE_1),6)

Ac = A - B*[k1 k2]
CE_2 = det(s*eye(2) - Ac);
CE_2 = vpa(CE_2)

% compare CE's manually to yield:

K1 = 98.169
K2 = 8.0 - 0.5696

% ------------------------- Calculating the Observer Gains -------------------------
syms L1 L2
multiplier = 10;
pole_1 = multiplier*((-DampingRatio*NaturalFrequency) + 1i*NaturalFrequency*sqrt(1-
DampingRatio^2));
pole_2 = multiplier*((-DampingRatio*NaturalFrequency) - 1i*NaturalFrequency*sqrt(1-
DampingRatio^2));
CE_1 = (s - pole_1)*(s - pole_2);
CE_1 = vpa(expand(CE_1),8)

Ao = [0 1; 0 -0.5696];
Bo = [0; 1];
Co = [0.07843 0];
Do = 0;
L = [L1; L2];

CE_2 = det(s*eye(2) - (Ao - L*Co));
vpa(CE_2)

% compare CE's manually to yield:
L1 = (80-0.5696)/0.07843
L2 = 9816.9015/0.07843

% ------------------------- Calculating the Integrator Gains ------------------------
clear("k1", "k2");
syms k1 k2 ke
```

```
multiplier = 1;
pole_multiplier = 10.389;
pole_1 = multiplier*(-DampingRatio*NaturalFrequency) + 1i*NaturalFrequency*sqrt(1-
DampingRatio^2)
pole_2 = multiplier*(-DampingRatio*NaturalFrequency) - 1i*NaturalFrequency*sqrt(1-
DampingRatio^2)
pole_3 = pole_multiplier*real(pole_1) % this makes the third pole a function of the
other two
CE_1 = ((s - pole_1)*(s - pole_2))*(s - pole_3);
CE_1 = vpa(expand(CE_1),8)

K = [k1, k2];
Aic = [A-B*K B*ke; -C 0];
CE_2 = det(s*eye(3) - Aic);
CE_2 = vpa(CE_2, 8)

% compare CE's manually to yield:

ke = 4079.5116/0.07843
k1 = 418.169
k2 = 48 - 0.5696

%% Checking the percent overshoot of each simulation --------------------------
clc; clear;

% Each simulation was stored in variable for using the "To Workspace" block in
Simulink. Each simulation's data was saved in a .mat file and named accordingly. This
allowed users to load each simulation into the program without running the Simulink
file each time.

disp("Simulation 1")
load("simulation1.mat")
output = squeeze(out.simout.Data(1,2,:));
sim1Data = stepinfo(output, out.tout)
sim1Ess = (abs(1-output(end))/1)*100

disp("Simulation 2")
load("simulation2.mat")
output = squeeze(out.simout.Data(1,2,:));
sim2Data = stepinfo(output, out.tout)
sim2Ess = (abs(1-output(end))/1)*100

disp("Simulation 3")
load("simulation3.mat")
output = out.simout.Data;
sim3Data = stepinfo(output, out.tout)
sim3Ess = (abs(1-output(end))/1)*100

disp("Simulation 4")
load("simulation4.mat")
output = out.simout.Data;
sim4Data = stepinfo(output, out.tout)
sim4Ess = (abs(1-output(end))/1)*100
```