EE128 Lab6 Writeup

Chris Mitchell, Shangge Liu

November 5th, 2018

Contents

1 Purpose

In the prior lab, we placed the poles of our physical system's state space's A matrix in order to make the system stable. In this lab, we address the issue that our state space, while observable, does not output all of our state space variables. Instead of using numerical approximations as done in last lab, we create a state space model $\hat{x}$ designed to approximate the entire state space. This approximation, beyond simply being useful for observation, supplies the necessary feedback in order for our pole placement model from the prior lab to work.

2 Prelab

2.1 Controllability and Observability

$rank(ctrb(A, B)) = 4$ and $rank(obsv(A, C)) = 4$, so the system is controllable and observable.

2.2 Observer Design

2.2.1 Dimensions of L

A is of dimension 4x4 and C is dimension 2x4, so L is of dimension 4x2.

2.2.2 Pole Placement of L

Since the eigenvalues of M are the same as the eigenvalues of M^T, so the eigenvalues of $A - LC$ are the same eigenvalues of $(A - LC)^T = A^T - C^T L^T$, so we can use the place command to find L^T, so $L = (place(A^T, C^T))^T$. From this,

$$L = \begin{bmatrix} 16.1619 & -2.3767 \\ 255.0069 & -5.4879 \\ 15.7116 & 21.0281 \\ 180.7793 & 378.2040 \end{bmatrix} \tag{1}$$

2.3 Simulation

2.3.1 Matlab Implementation

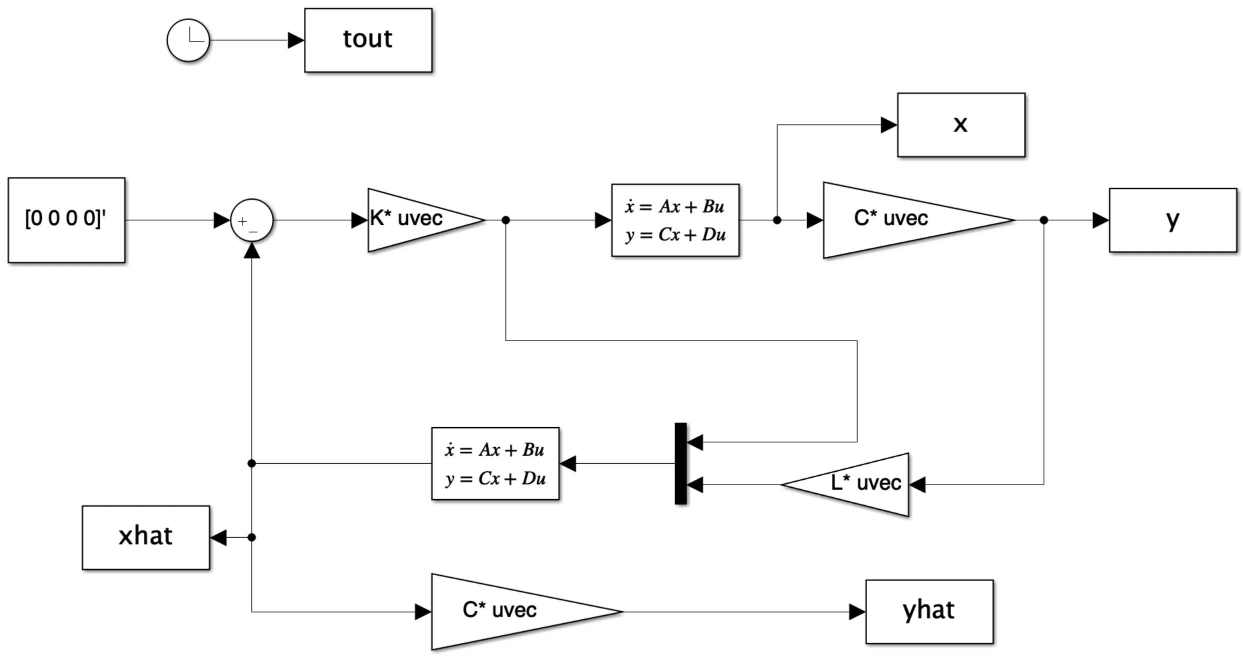

Figure 1: Simulink Model With Simulated Physical System

In this figure, the upper state space block is x and the lower state space block is $\hat{x}$. For the matrices for x, A and B are the same A and B from the previous labs (without the K modification as that is done beforehand to the input), with $D = 0$ and $C = I$. This way, the output is x, allowing us to view that variable, and a simple multiplication by C from the previous lab allows us to view y.

For $\hat{x}$, $A_L = \begin{bmatrix} A - LC & 0 \end{bmatrix}$ and $B_L = \begin{bmatrix} B & I \end{bmatrix}$. This way, with the input $\begin{bmatrix} -K(\hat{x} - ref) \\ L * y \end{bmatrix}$, we get the equation:

$$\dot{\hat{x}} = (A - LC)\hat{x} - BK(\hat{x} + ref) \tag{2}$$

$$\dot{\hat{x}} = (A - BK)\hat{x} + BK * ref + L(y - \hat{y}) \tag{3}$$

$$\dot{\hat{x}} = A_K\hat{x} + B_K * ref + L(y - \hat{y}) \tag{4}$$

which is the desired state space equation. $C = I$ and $D = 0$, so the output is $\hat{x}$ so we can view said variable. For this reason, we also multiply the output by C to get $\hat{y}$.

2.3.2 Simulation with Initial Perturbation

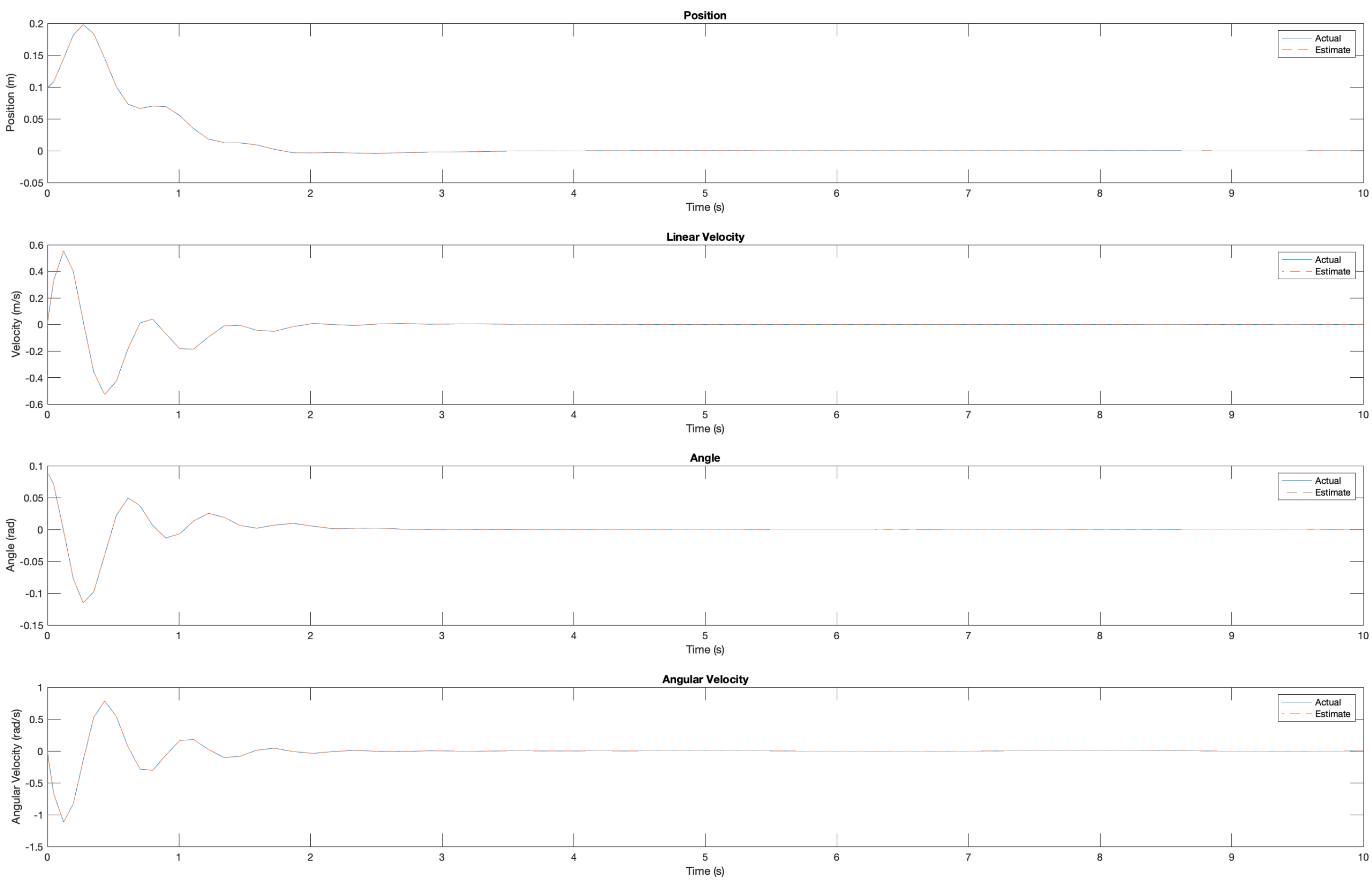

Figure 2: Comparisons of x with $\hat{x}$

2.3.3 Error Plot of $e = \hat{x} - x$

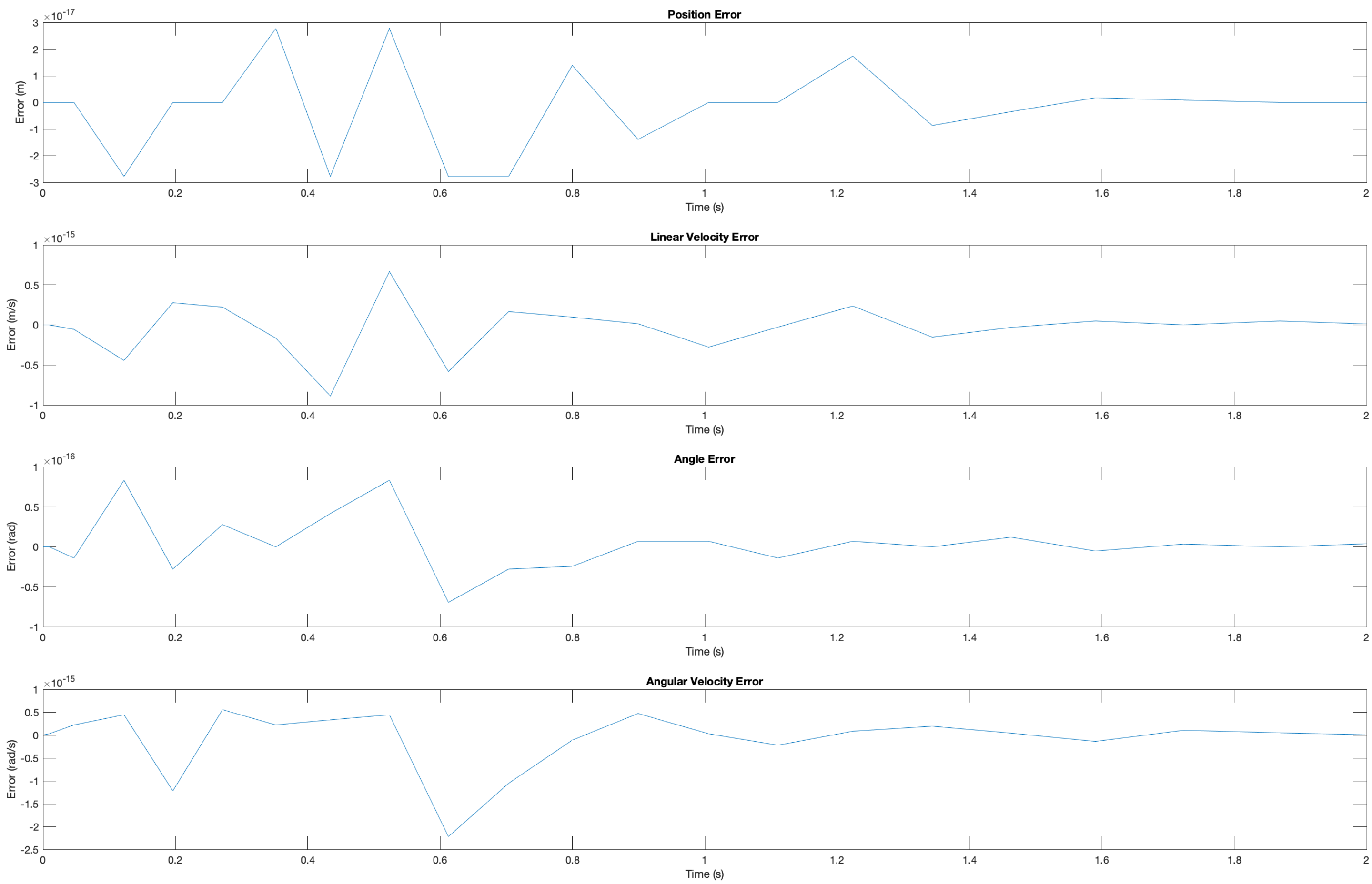

Figure 3: Error Plot of $e = \hat{x} - x$

From the graphs we can see that all the error plots converge to zero within the first second or two.

3 Lab Results and Analysis

3.1 Hardware Implementation

We added the Luenberger observer on the simulink model.The values of parameteres are all the same as the prelab. Whether the input signal is sine wave or zero depends on the requirements in the lab sheet.

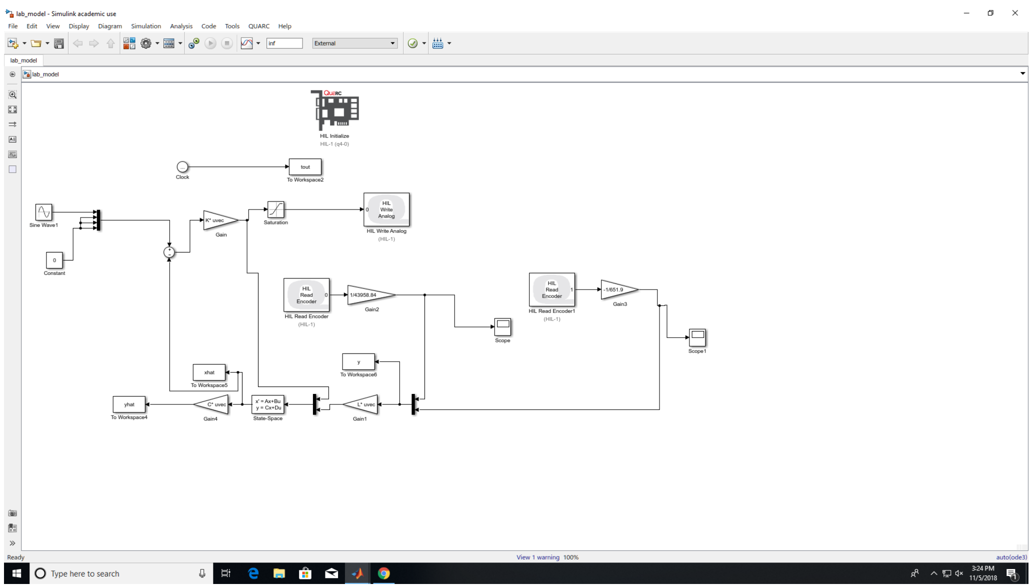

Figure 4: Simulink Model of Lab6b

In this part, after we built, connected and ran the model with the stick initially straight up, we oberved that the stick kept vertically up. Then we tapped it lightly and got the data within about 10 seconds. After that, we plot the y and yhat and compared them together.

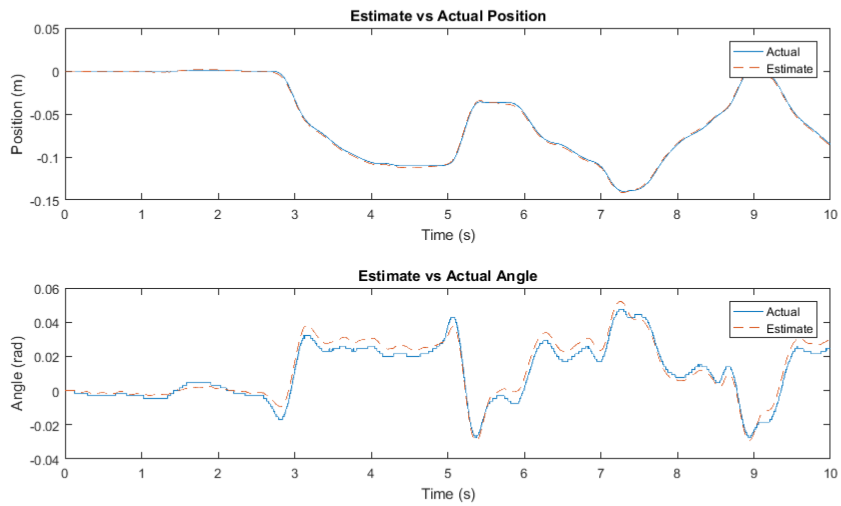

Figure 5: The Esitmated and Actual Output signals

From the plot, we knew that the observer extimated the system pretty well.

3.2 Controller Comparison

3.2.1 Zero Reference

Here is the simulink model in lab6a without the observer:

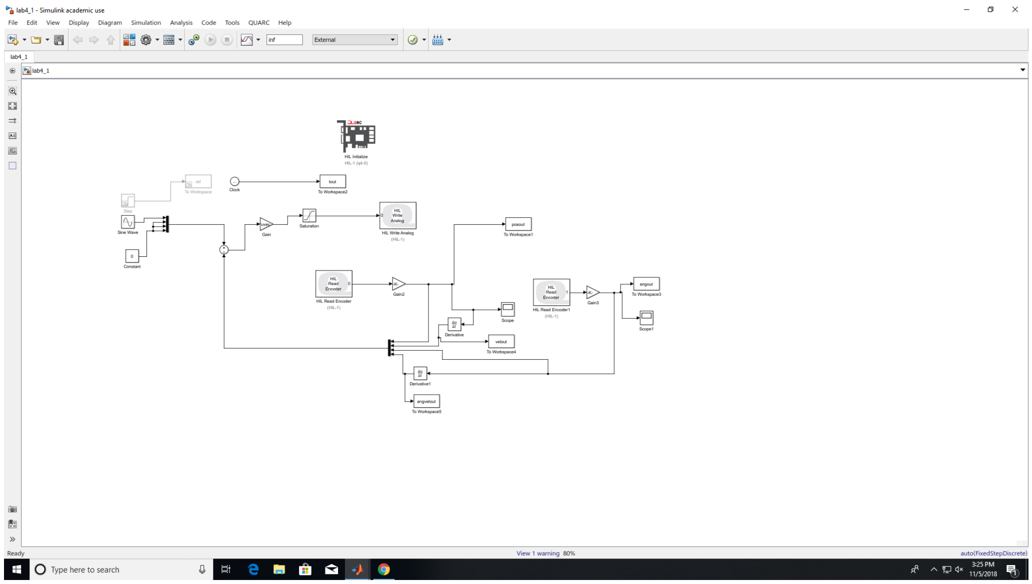

Figure 6: Simulink Model without the Observer

After simulating both models with the stick initially vertically up, we plotted the output signals and compared both the position and the angle of these two models with different input. Here is the plot of the output with zero reference.

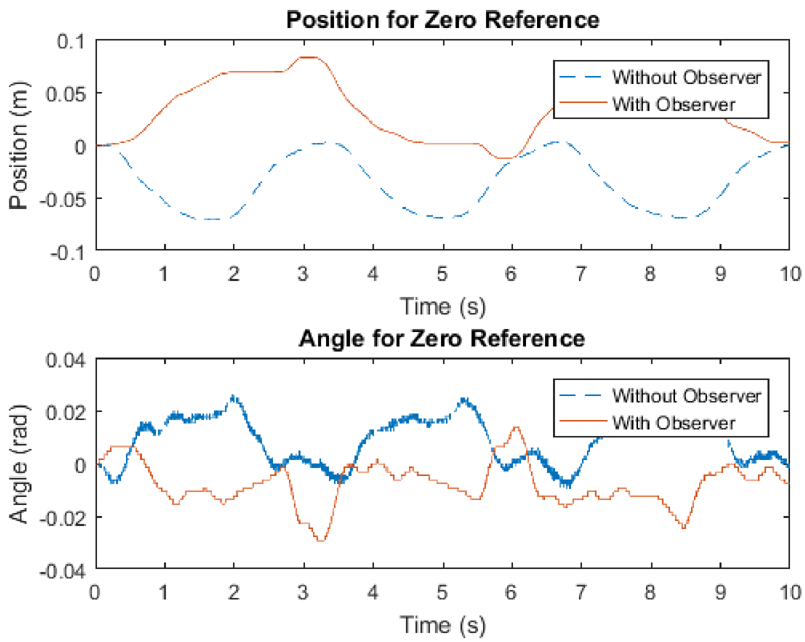

Figure 7: The output signals of the two system with zero reference

From the plot, we know that both of the systems work well because the position and angle are really close to zero, though there is still some sine wave disturbance. However, the system without the observer have much louder noise from the gears than the one with the observer. So we can say that the system with the observer works better.

3.2.2 Zero Reference with Small Perturbations

Here is the plot of the output with small perturbations.

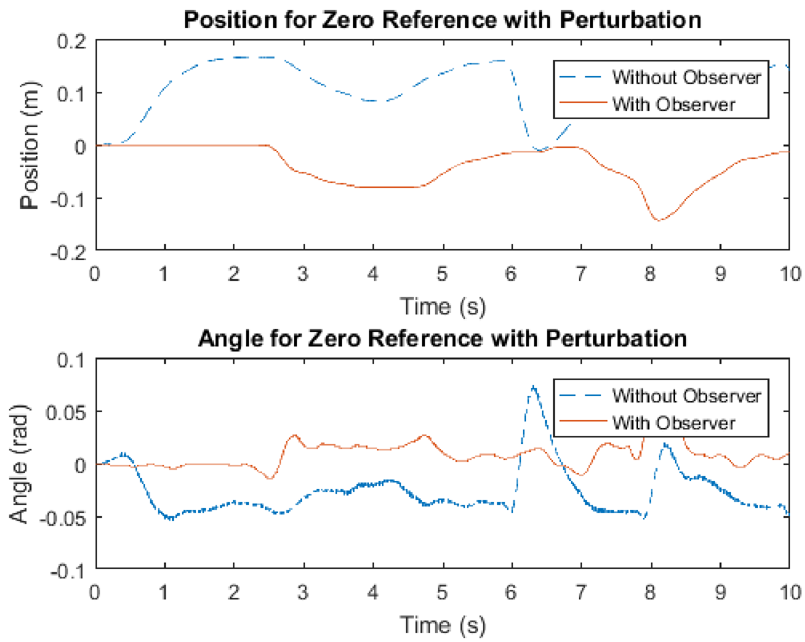

Figure 8: The output signals of the two system with small perturbations

The plots show that with small perturbations, the output signal of the system with observer is more close to zero, which means that the system with the observe is more stable and has a much stronger capacity of resisting disturbance than the first system.

3.2.3 Sinusoidal Reference Position

Here is the plot of the output with sinusoidal reference.

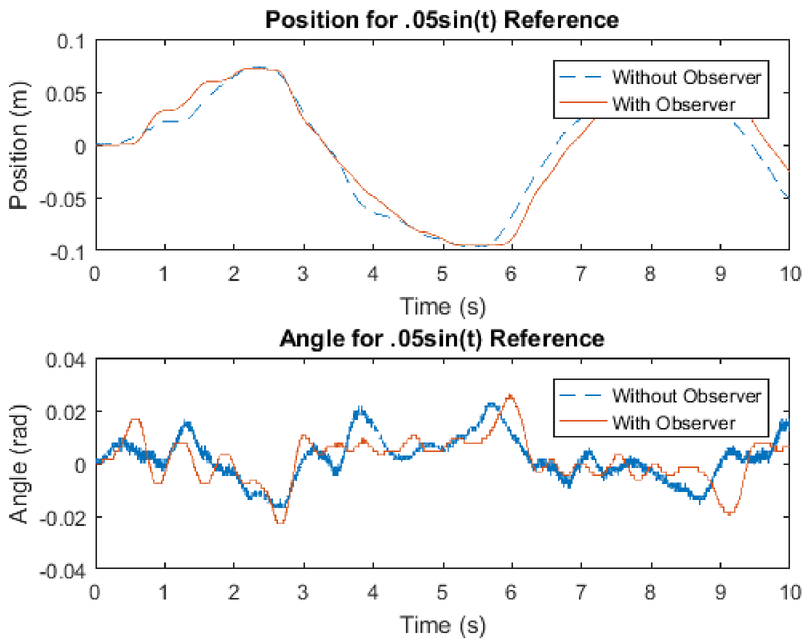

Figure 9: Caption

From the plot we know that both of the systems track the sinusoidal reference well. There is nearly no difference between the output of the two systems.

3.3 Comparison of Velocity Estimates

Here is the plot of the velocity of the cart.

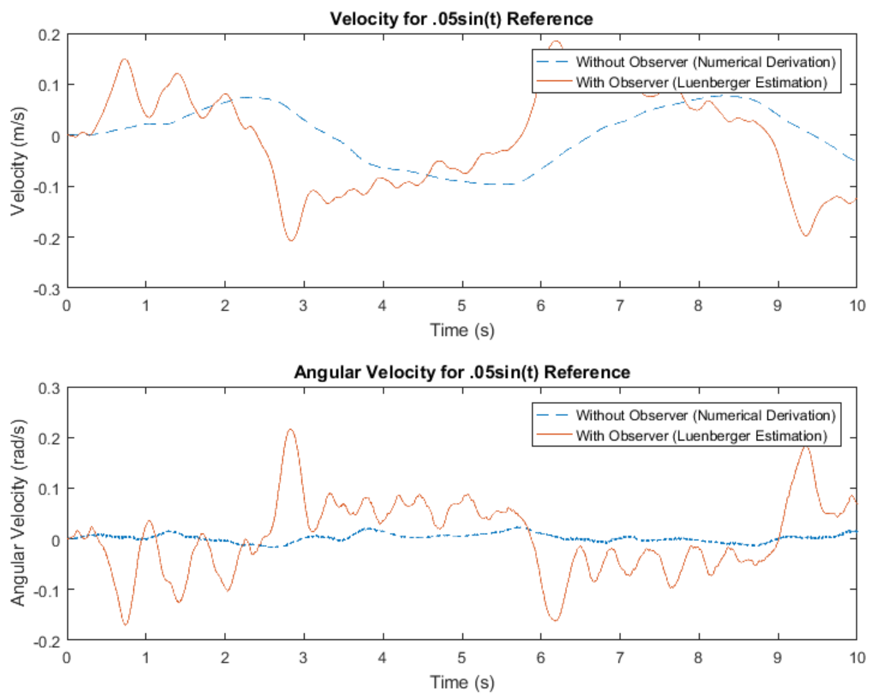

Figure 10: Caption

From the velocity plot, we know that the system with observer is more sensitive to the noise in the measurement.

3.4 Comparison of Schemes

Personally, the system with the observer gives better performance. It has much better resistance to the perturbations and much less noise in the gears. However, the velocity of the cart might not be stable in some cases.

Appendices

A Code for Prelab Calculations

```
a22 = -6.81;
a23 = -1.5;
a42 = 15.47;
a43 = 25.67;
```

```
b2 = 1.52;
b4 = -3.46;

A = [0 1 0 0; 0 a22 a23 0; 0 0 0 1; 0 a42 a43 0];

B = [0; b2; 0 ; b4];

C = [1 0 0 0; 0 0 1 0];
D = [0; 0];

sys = ss(A, B, C, D);
K = [-13.0659  -14.7536  -48.0527   -6.594];

r1 = rank(ctrb(sys));
r2 = rank(obsv(sys));

obs_poles = [-10 + 15j, -10 - 15j, -12 + 17j, -12 - 17j];
%L = place(A', C', obs_poles)';

L = [16.1619   -2.3767;
    255.0069   -5.4879;
    15.7116   21.0281;
    180.7793  378.2040];


Al = A - L*C;
Bl = [B eye(4)];
Cl = eye(4);
Dl = zeros([4 5]);
```

B Code for Prelab Estimations

```
prelab1;

sim("prelab_model")

clf

subplot(4, 1, 1)
plot(tout, x(:, 1), '-', tout, xhat(:, 1), '--')
legend("Actual", "Estimate")
title("Position")
xlabel("Time (s)")
```

```
ylabel("Position (m)")


subplot(4, 1, 2)
plot(tout, x(:, 2), '-', tout, xhat(:, 2), '--')
legend("Actual", "Estimate")
title("Linear Velocity")
xlabel("Time (s)")
ylabel("Velocity (m/s)")

subplot(4, 1, 3)
plot(tout, x(:, 3), '-', tout, xhat(:, 3), '--')
legend("Actual", "Estimate")
title("Angle")
xlabel("Time (s)")
ylabel("Angle (rad)")

subplot(4, 1, 4)
plot(tout, x(:, 4), '-', tout, xhat(:, 4), '--')
legend("Actual", "Estimate")
title("Angular Velocity")
xlabel("Time (s)")
ylabel("Angular Velocity (rad/s)")
```

C Code for Prelab Error Calculation

```
prelab1;

sim("prelab_model")

clf

subplot(4, 1, 1)
plot(tout, xhat(:, 1) - x(:, 1), '-')
title("Position Error")
xlabel("Time (s)")
ylabel("Error (m)")
xlim([0 2])


subplot(4, 1, 2)
plot(tout, xhat(:, 2) - x(:, 2), '-')
title("Linear Velocity Error")
xlabel("Time (s)")
ylabel("Error (m/s)")
```

```
xlim([0 2])

subplot(4, 1, 3)
plot(tout, xhat(:,3) - x(:, 3), '-')
title("Angle Error")
xlabel("Time (s)")
ylabel("Error (rad)")
xlim([0 2])

subplot(4, 1, 4)
plot(tout, xhat(:, 4) - x(:, 4), '-')
title("Angular Velocity Error")
xlabel("Time (s)")
ylabel("Error (rad/s)")
xlim([0 2])
```

D Code for Lab Model Setup

```
a22 = -6.81;
a23 = -1.5;
a42 = 15.47;
a43 = 25.67;

b2 = 1.52;
b4 = -3.46;

A = [0 1 0 0; 0 a22 a23 0; 0 0 0 1; 0 a42 a43 0];

B = [0; b2; 0 ; b4];

C = [1 0 0 0; 0 0 1 0];


K = [-13.0659  -14.7536  -48.0527   -6.594];


L = [16.1619   -2.3767;
    255.0069   -5.4879;
    15.7116   21.0281;
    180.7793  378.2040];


Al = A - L*C;
Bl = [B eye(4)];
Cl = eye(4);
```

```
Dl = zeros([4 5]);
```

E Code for Lab Section 1

```
lab_model_setup;

clf;
subplot(2, 1, 1);
plot(tout, y(:, 1), '-', tout, yhat(:, 1), '--')
legend('Actual', 'Estimate')
title('Estimate vs Actual Position')
xlabel('Time (s)')
ylabel('Position (m)')
xlim([0 10])

subplot(2, 1, 2);
plot(tout, y(:, 2), '-', tout, yhat(:, 2), '--')
legend('Actual', 'Estimate')
title('Estimate vs Actual Angle')
xlabel('Time (s)')
ylabel('Angle (rad)')
xlim([0 10])
```

F Code for Lab Section 2

```
clf;

len = min([length(tout), length(y(:, 1)), length(posout)]);

tout = tout(1:len);
y = y(1:len, :);
posout = posout(1:len);
angout = angout(1:len);

subplot(2, 1, 1);
plot(tout, posout, '--', tout, y(:, 1), '-');
title('Position for .05sin(t) Reference')
xlabel('Time (s)')
ylabel('Position (m)')
legend('Without Observer', 'With Observer')
xlim([0 10])

subplot(2, 1, 2);
plot(tout, angout, '--', tout, y(:, 2), '-');
```

```
title('Angle for .05sin(t) Reference')
xlabel('Time (s)')
ylabel('Angle (rad)')
legend('Without Observer', 'With Observer')
xlim([0 10])
```

G Code for Lab Section 3

```
len = min([length(tout), length(xhat(:, 2)), length(velout)]);

tout = tout(1:len);
xhat = xhat(1:len, :);
velout = posout(1:len);
angvelout = angout(1:len);

subplot(2, 1, 1);
plot(tout, velout, '--', tout, xhat(:, 2), '-');
title('Velocity for .05sin(t) Reference')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
legend('Without Observer (Numerical Derivation)', 'With Observer (Luenberger Estimation)')
xlim([0 10])

subplot(2, 1, 2);
plot(tout, angout, '--', tout, xhat(:, 4), '-');
title('Angular Velocity for .05sin(t) Reference')
xlabel('Time (s)')
ylabel('Angular Velocity (rad/s)')
legend('Without Observer (Numerical Derivation)', 'With Observer (Luenberger Estimation)')
xlim([0 10])
```