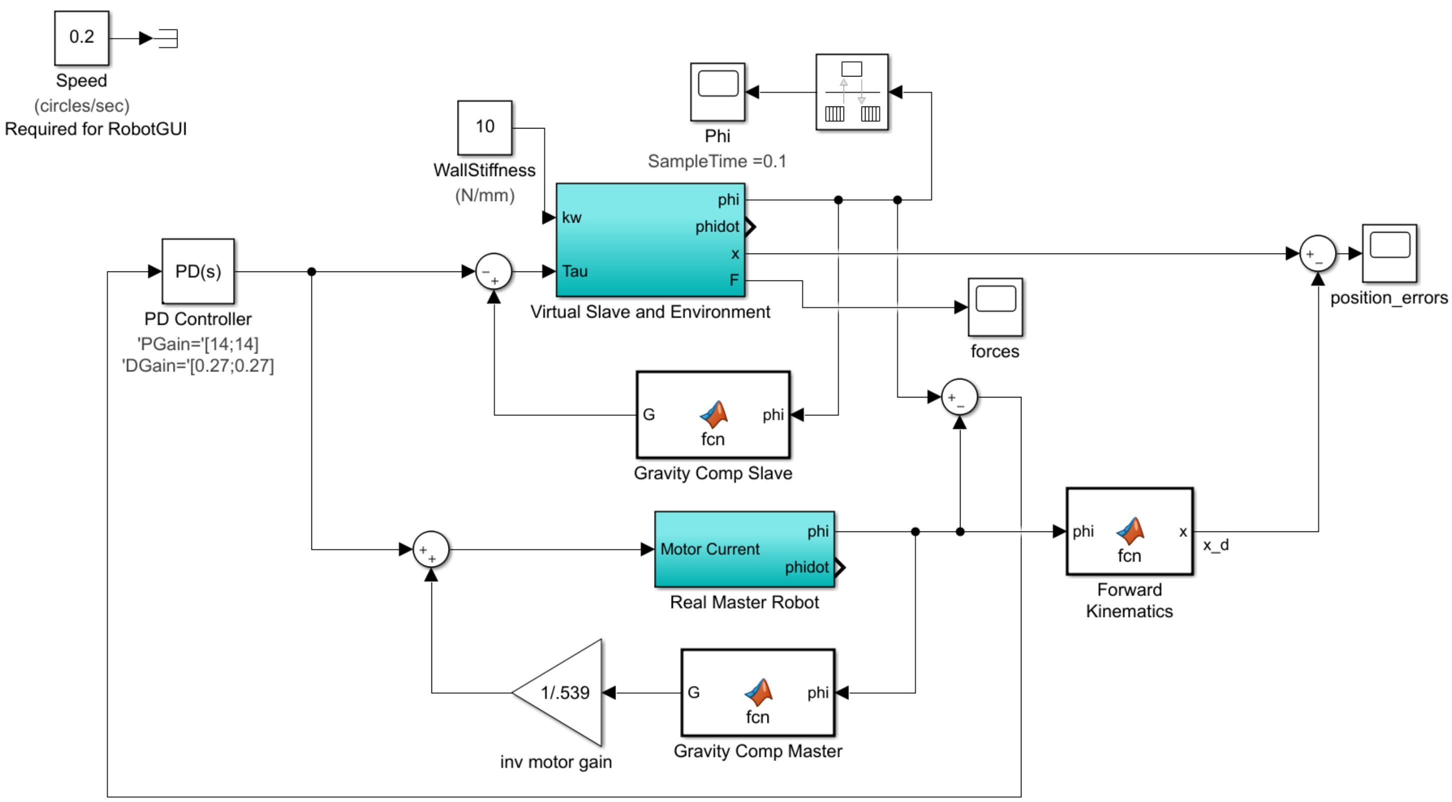
ME EN 6230
Lab 6
Ryan Dalby

Note that lab station 3 was used for this lab.

1. Bilateral Servo in Joint Space

1.1. Model

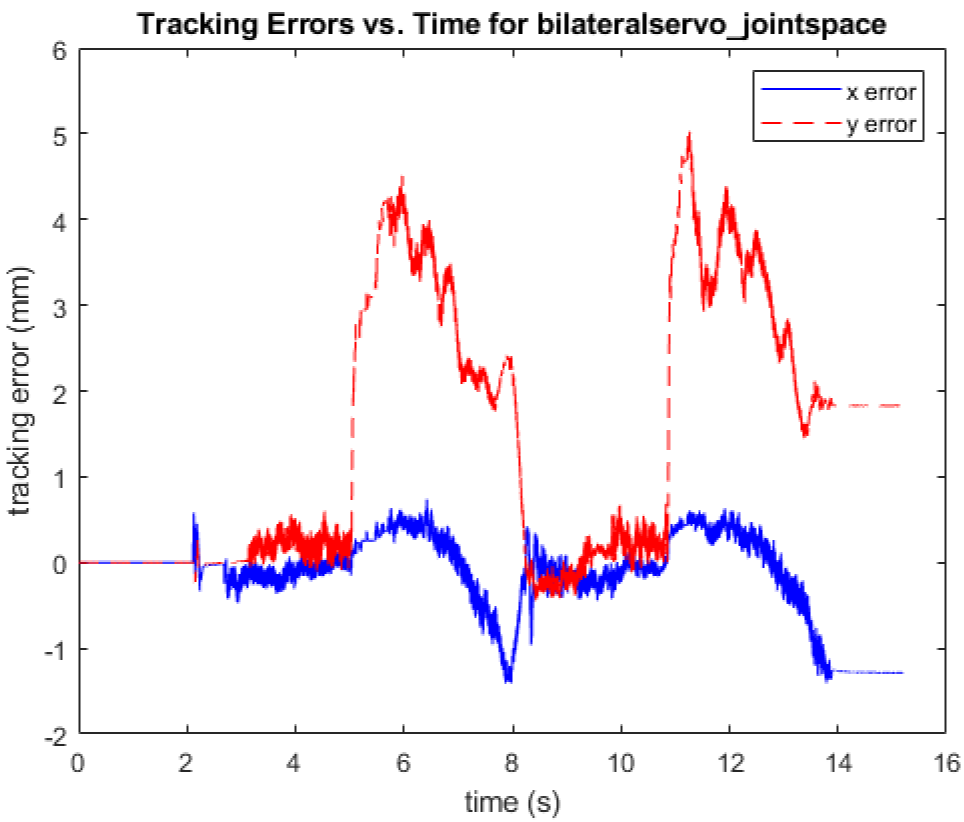


1.2. Code

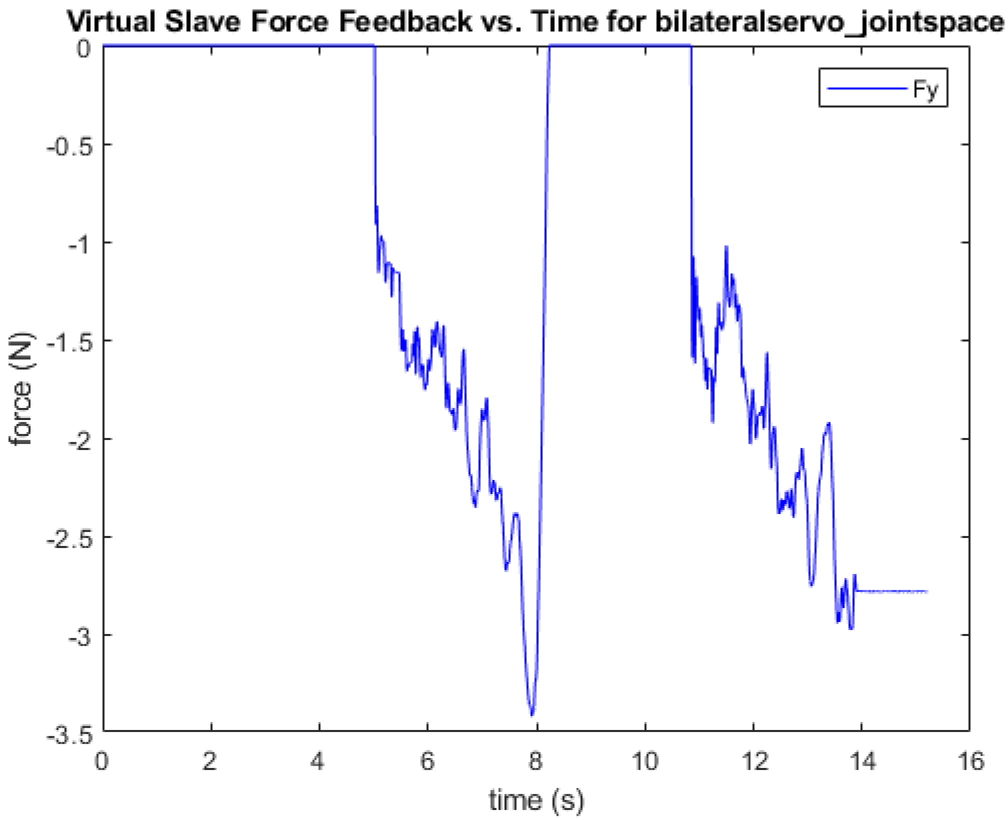
```
Forward Kinematics block:
function x = fcn(phi)
% This block supports an embeddable subset of the MATLAB language.See the
help menu for details.
a1=0.15;
a2=0.15;
x = [0;0];
x(1) = a1*cos(phi(1))+a2*cos(phi(2));
x(2) = a1*sin(phi(1))+a2*sin(phi(2));
```

1.3. Soft Wall (k_w = 1 N/mm)

1.3.1. Position Tracking Error

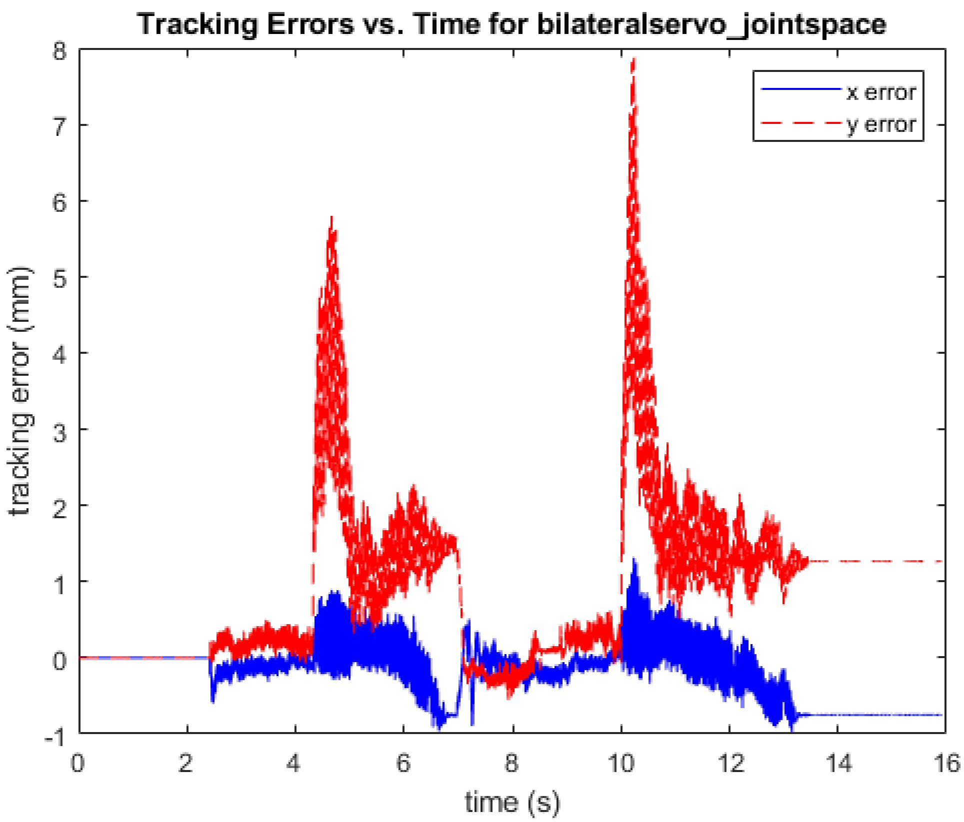


1.3.2. Force Feedback

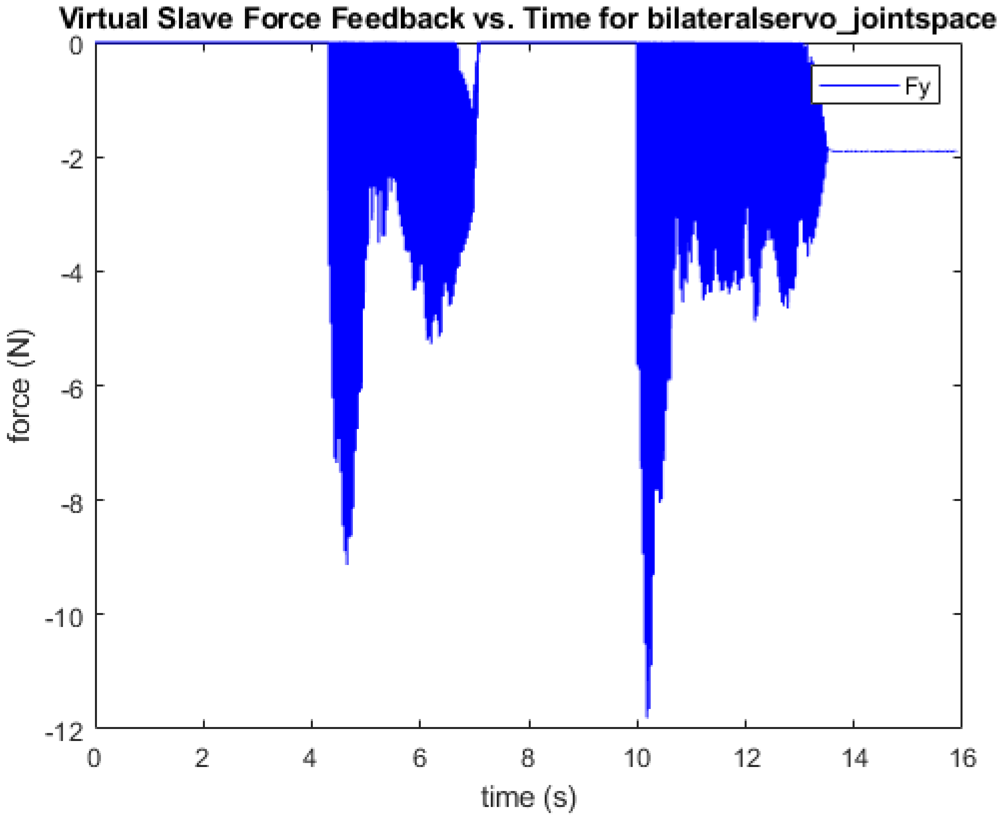


1.4. Hard Wall (k_w = 10 N/mm)

1.4.1. Position Tracking Error



1.4.2. Force Feedback

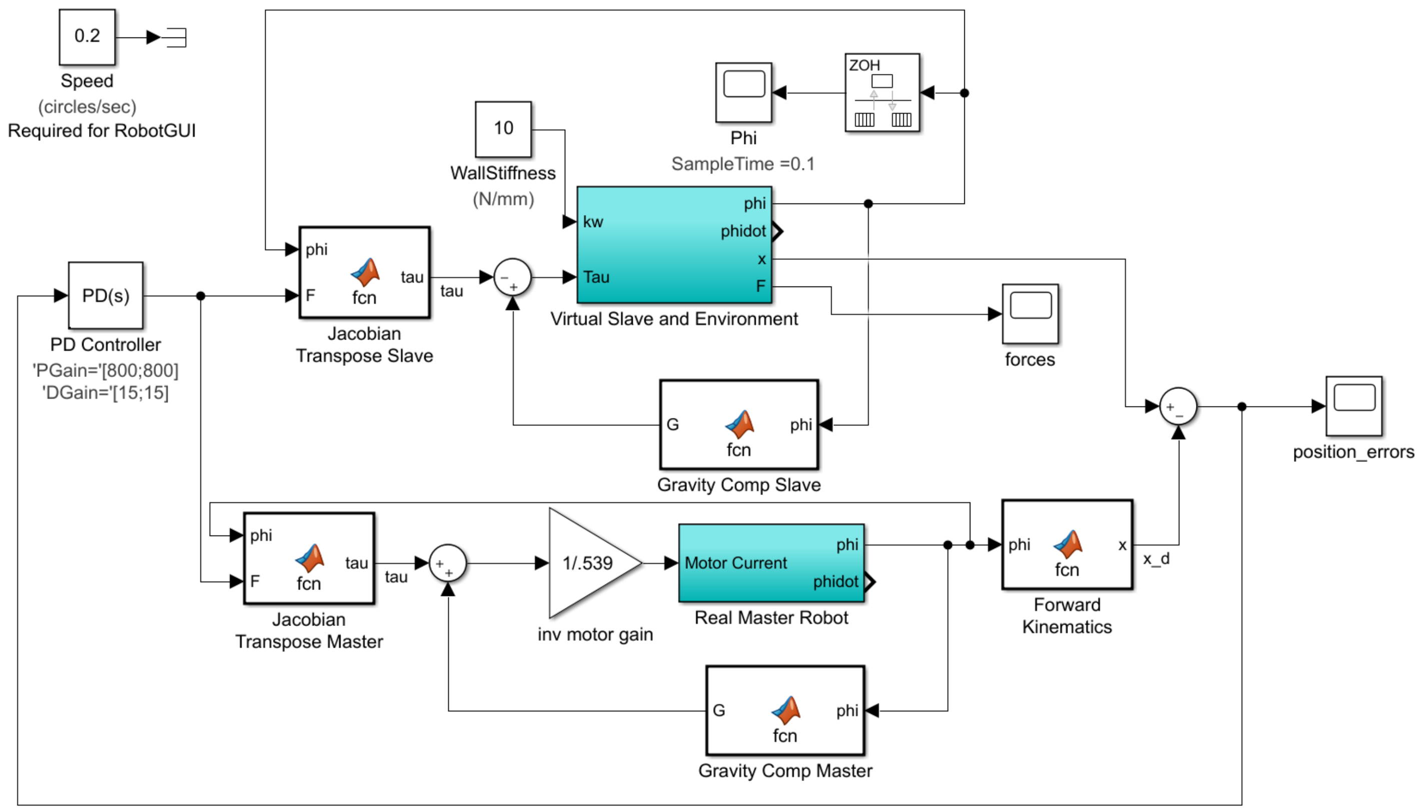


1.5. Analysis

The bilateral servo controller in joint space manages to achieve stable control and also have good tracking performance for both soft and hard virtual walls. I was able to perceive the contact with the virtual wall but not in the way I would expect if I had made contact with a physical wall. When I pulled down on the master robot at the contact point of the virtual wall it did not have the same “hard”/”sudden” stop I would expect if I had just hit the surface of the table. This observation remained consistent even with the hard virtual wall.

2. Bilateral Servo in Operational Space

2.1. Model



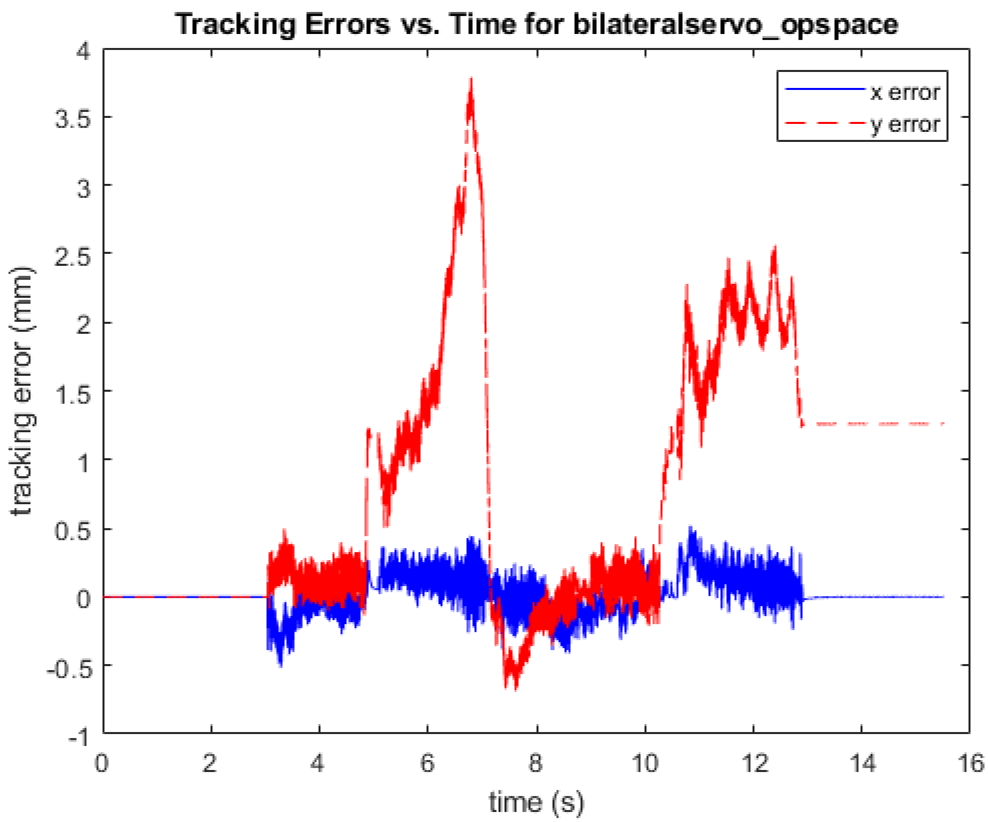
2.2. Code

The Jacobian Transpose Slave and Jacobian Transpose Master share the same code:

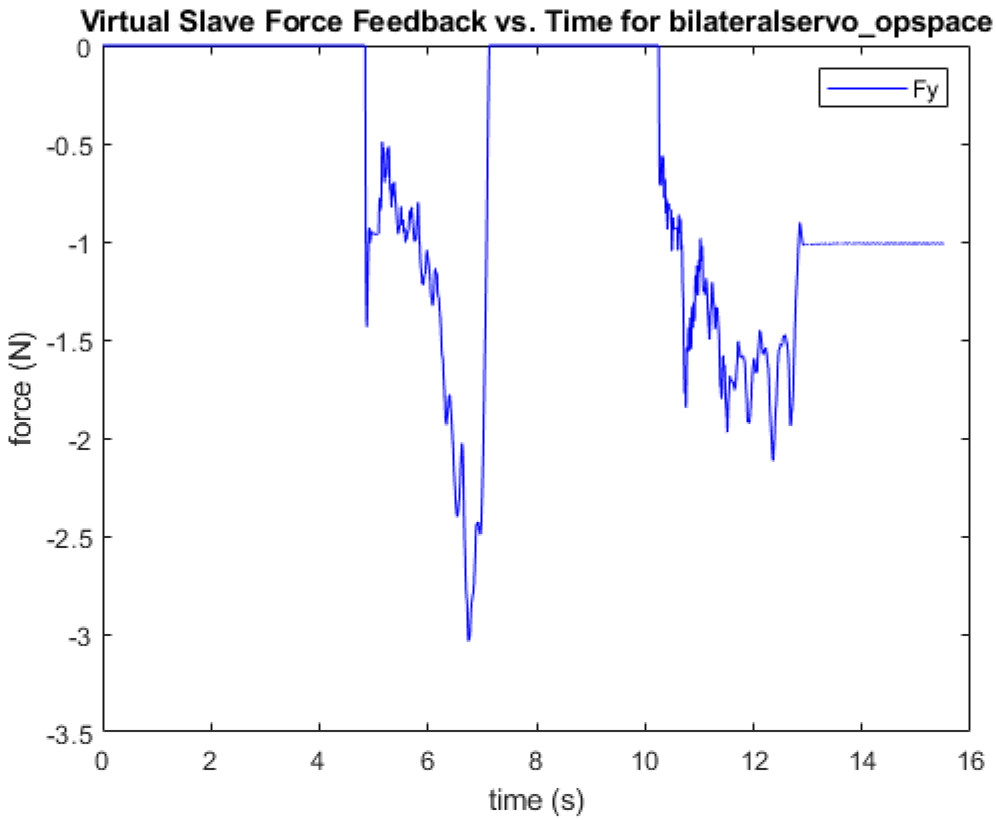
```
function tau = fcn(phi, F)
% This block supports an embeddable subset of the MATLAB language.
% See the help menu for details.
a1=0.15;
a2=0.15;
J11 = -a1*sin(phi(1));
J12 = -a2*sin(phi(2));
J21 = a1*cos(phi(1));
J22 = a2*cos(phi(2));
J = [J11 J12; J21 J22];
tau = J'*F;
```

2.3. Soft Wall (k_w = 1 N/mm)

2.3.1. Position Tracking Error

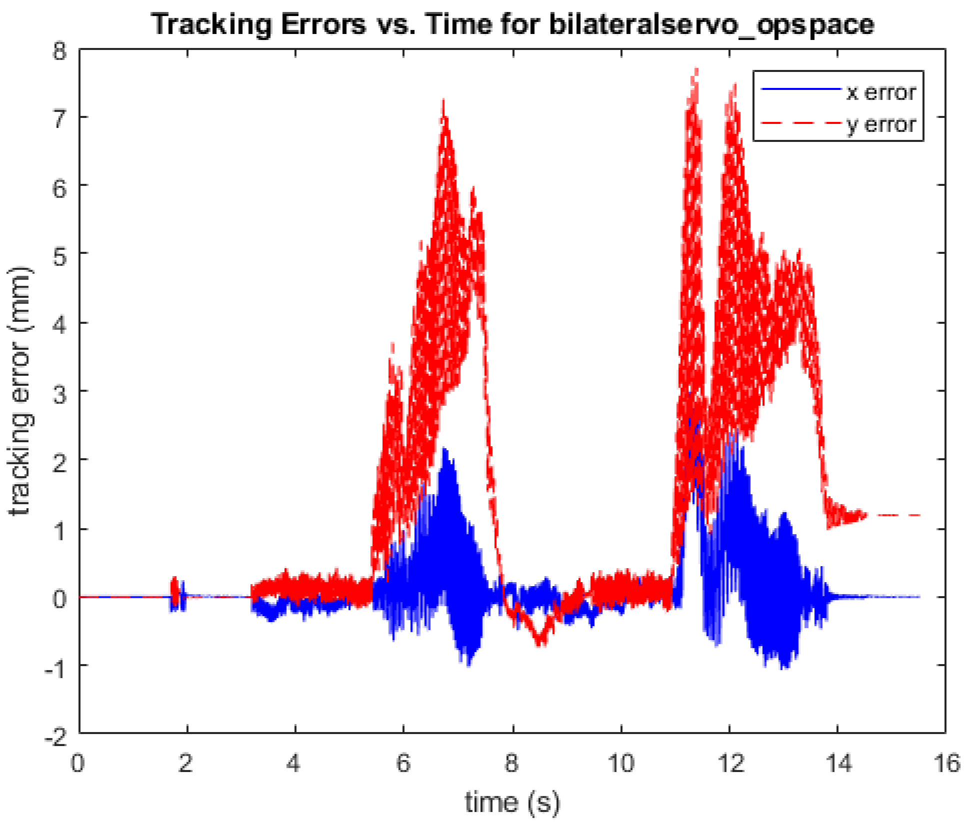


2.3.2. Force Feedback

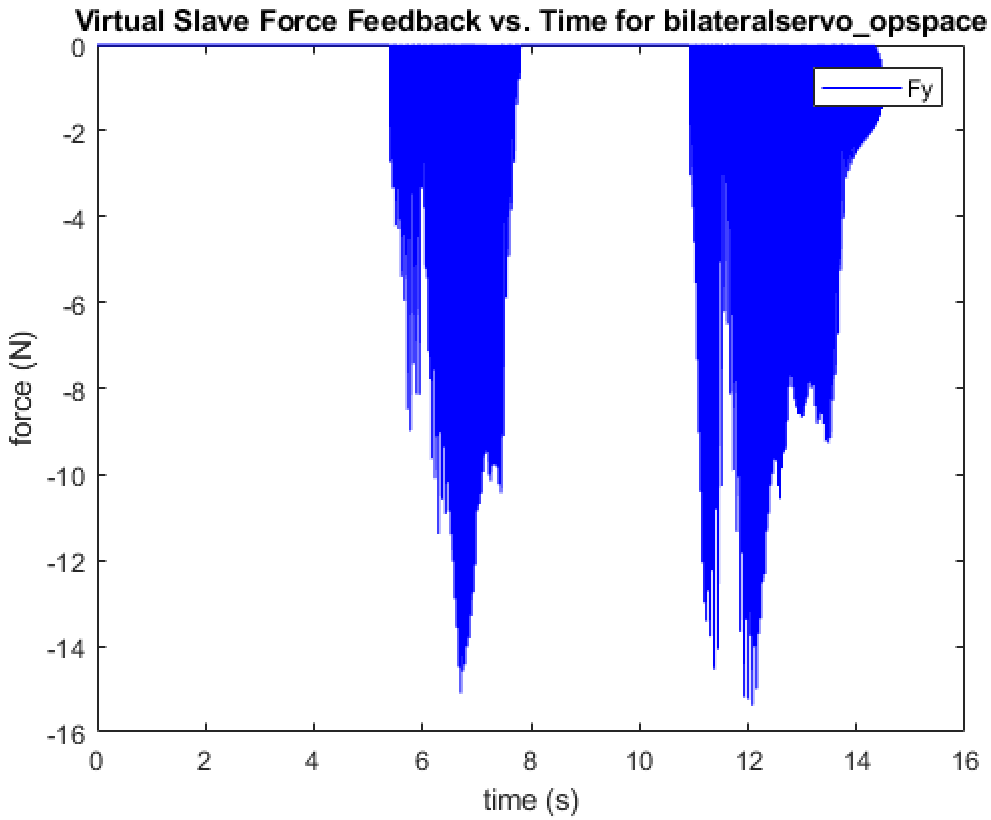


2.4. Hard Wall (k_w = 10 N/mm)

2.4.1. Position Tracking Error



2.4.2. Force Feedback

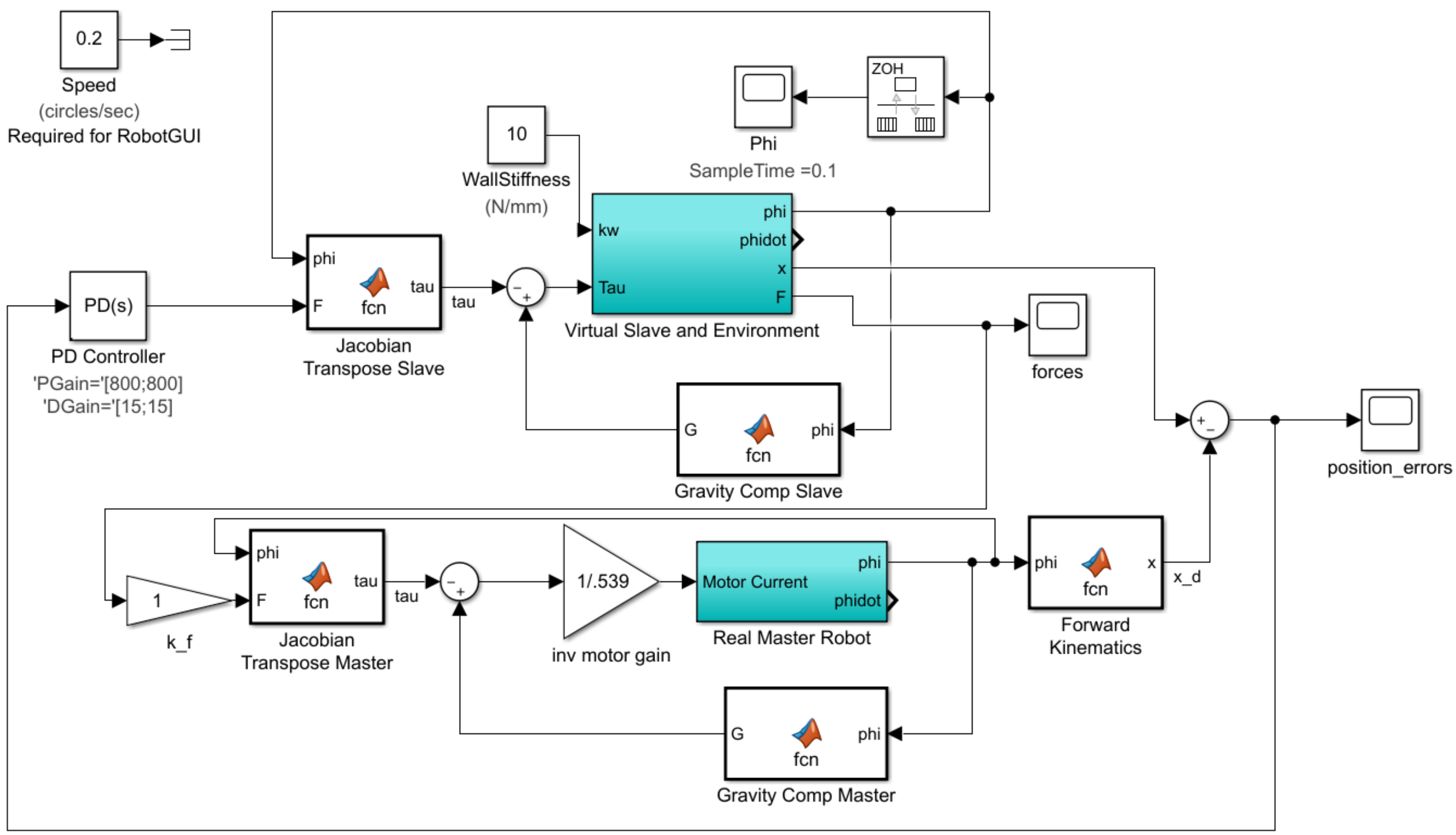


2.5. Analysis

The bilateral servo controller in operational space manages to achieve stable control (unless near a singularity) and also has good tracking performance for both soft and hard virtual walls. The transparency of this controller was the exact same as the bilateral servo controller in joint space, lacking the true feel of a sudden contact with a wall. The one caveat to this controller was that if I extended the master robot arm out near its singularity the slave robot would often toggle to be the opposite elbow position. If the slave robot was a physical robot this could result in unstable control since the robot can unexpectedly contact surfaces and induce large disturbances into the system.

3. Slave Servo with Direct Force Feedback to Master

3.1. Model

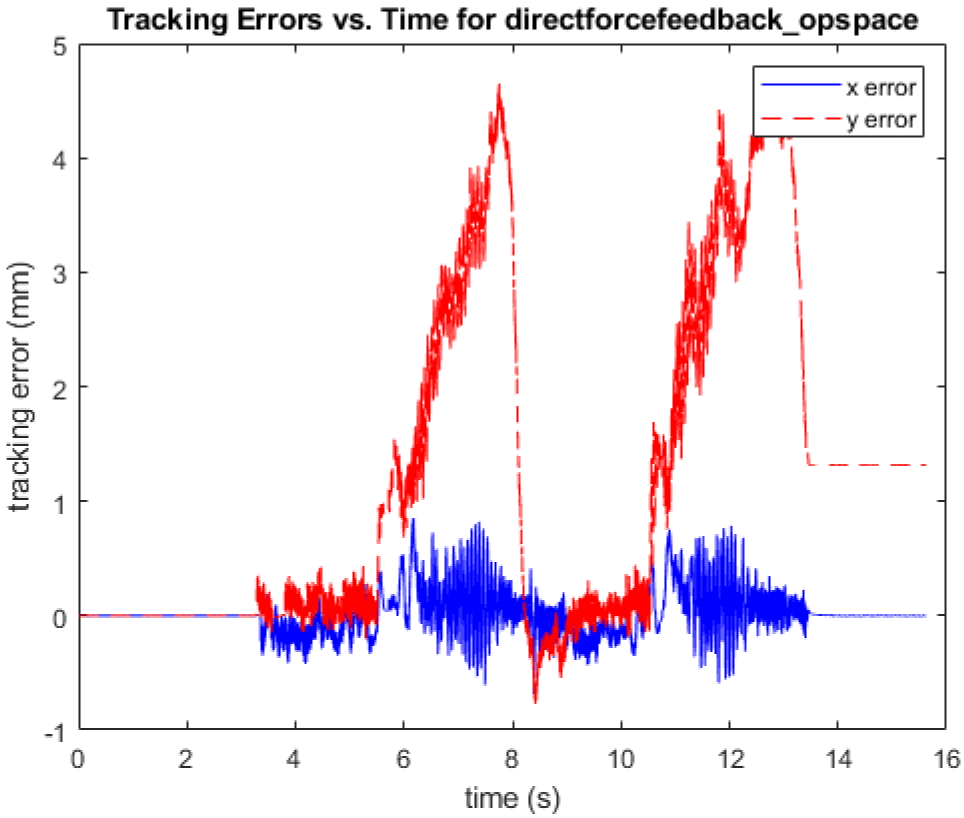


3.2. Code

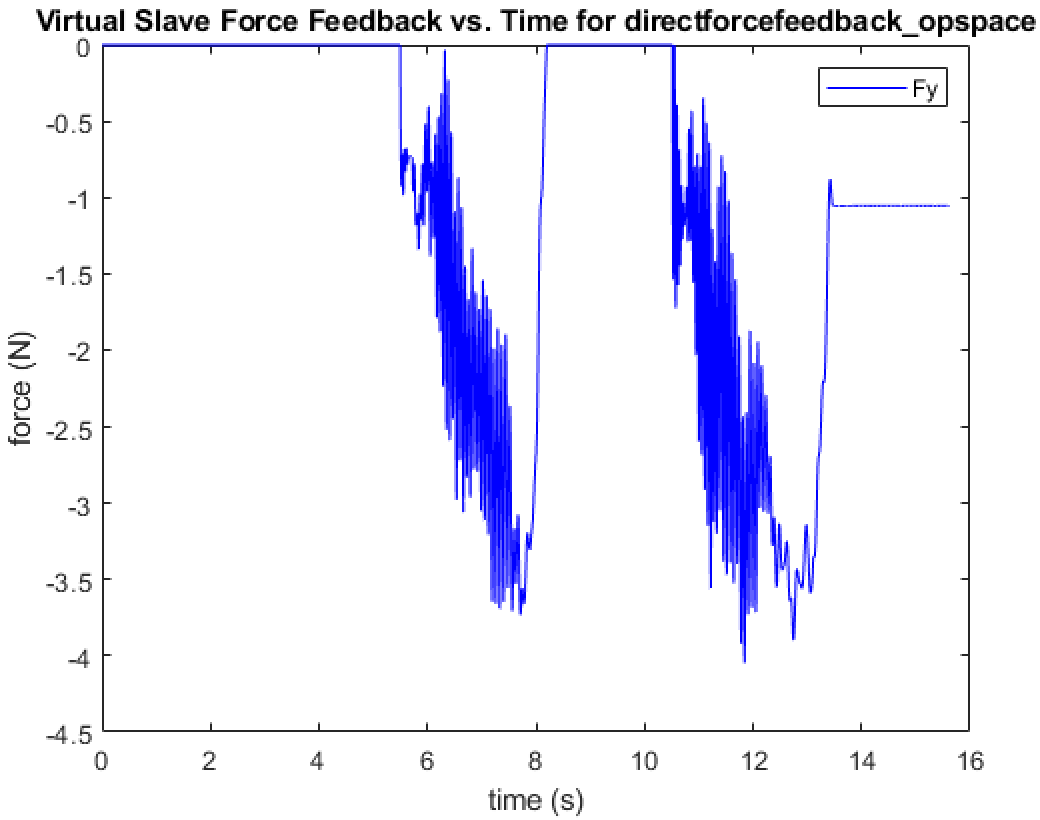
No additional MATLAB function blocks were needed for this controller that were not already described by the template or above.

3.3. Soft Wall (k_w = 1 N/mm)

3.3.1. Position Tracking Error

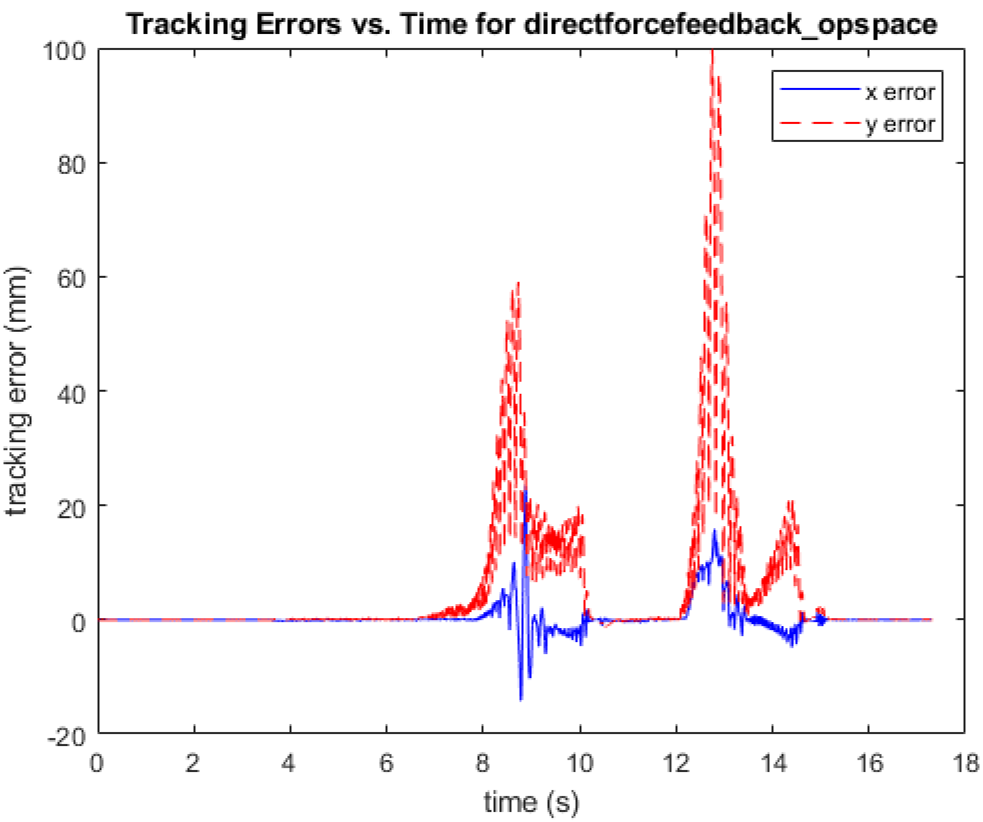


3.3.2. Force Feedback

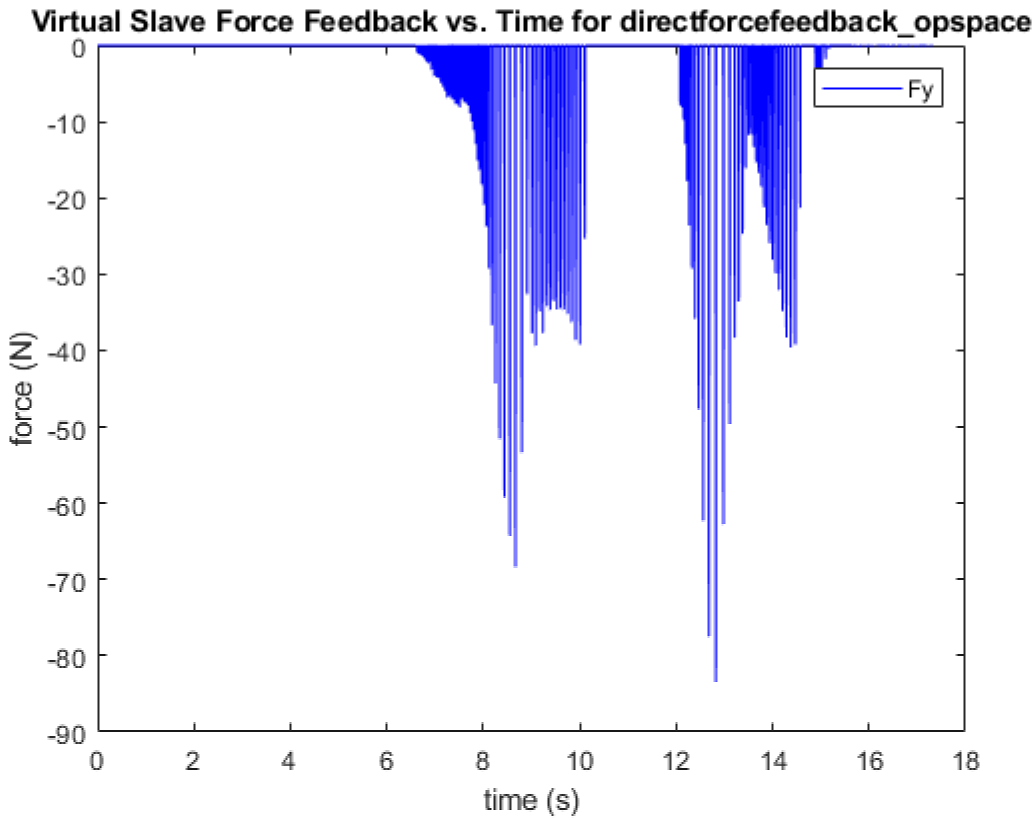


3.4. Hard Wall (k_w = 10 N/mm)

3.4.1. Position Tracking Error



3.4.2. Force Feedback

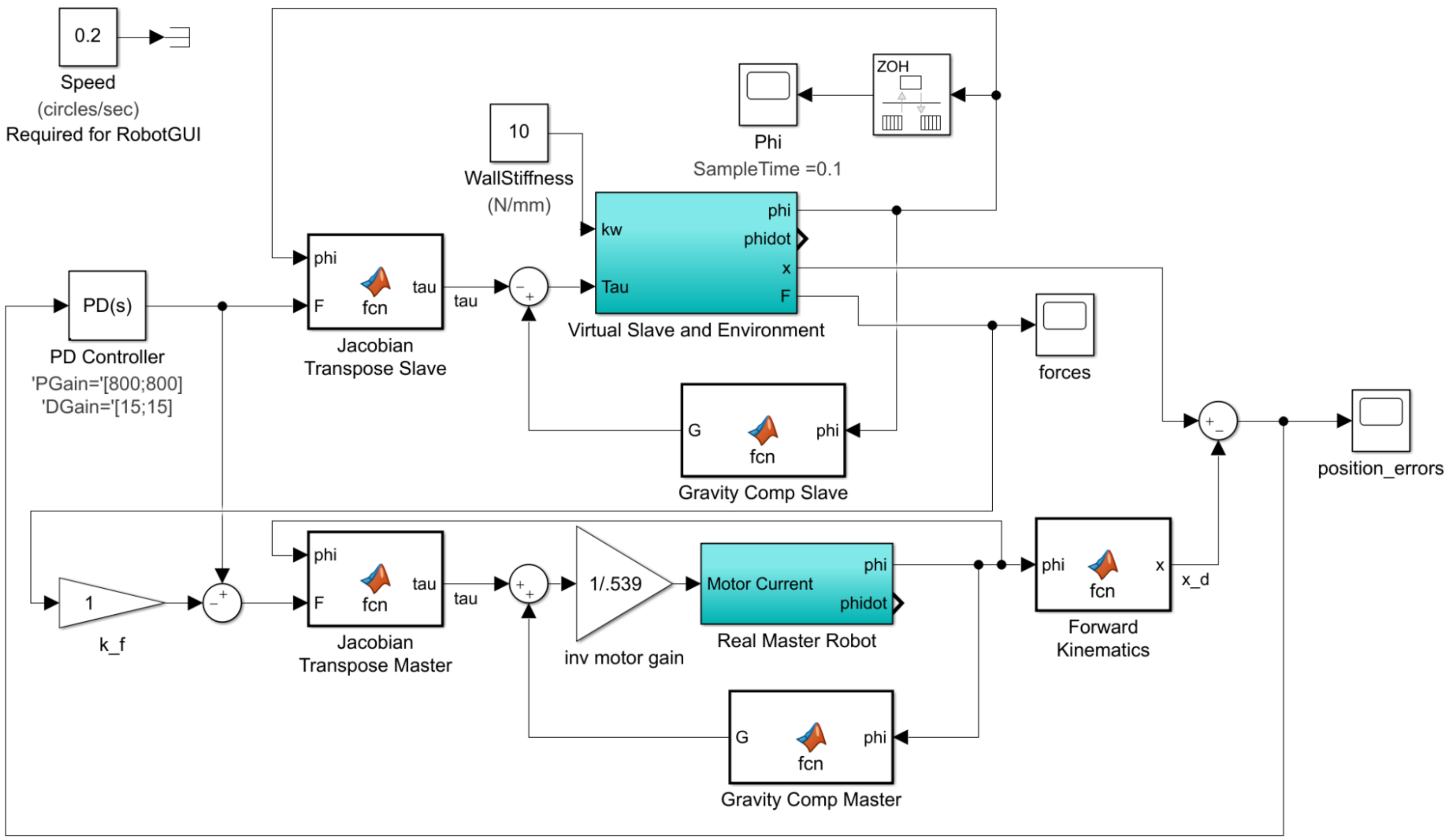


3.5. Analysis

The slave servo controller with direct feedback to the master robot resulted in a controller that was often unstable. For a soft virtual wall the controller was generally stable as long as the master robot was not pulled on too hard once in contact with the virtual wall. For hard virtual walls the controller was unstable as soon as it came into contact with the wall. In terms of transparency I was able to perceive the “hard”/”sudden” stop better than the bilateral servo controllers. The main issue with the transparency was that the perception of the virtual slave was lost once the controller went unstable.

4. Bilateral Servo with Direct Force Feedback to Master

4.1. Model

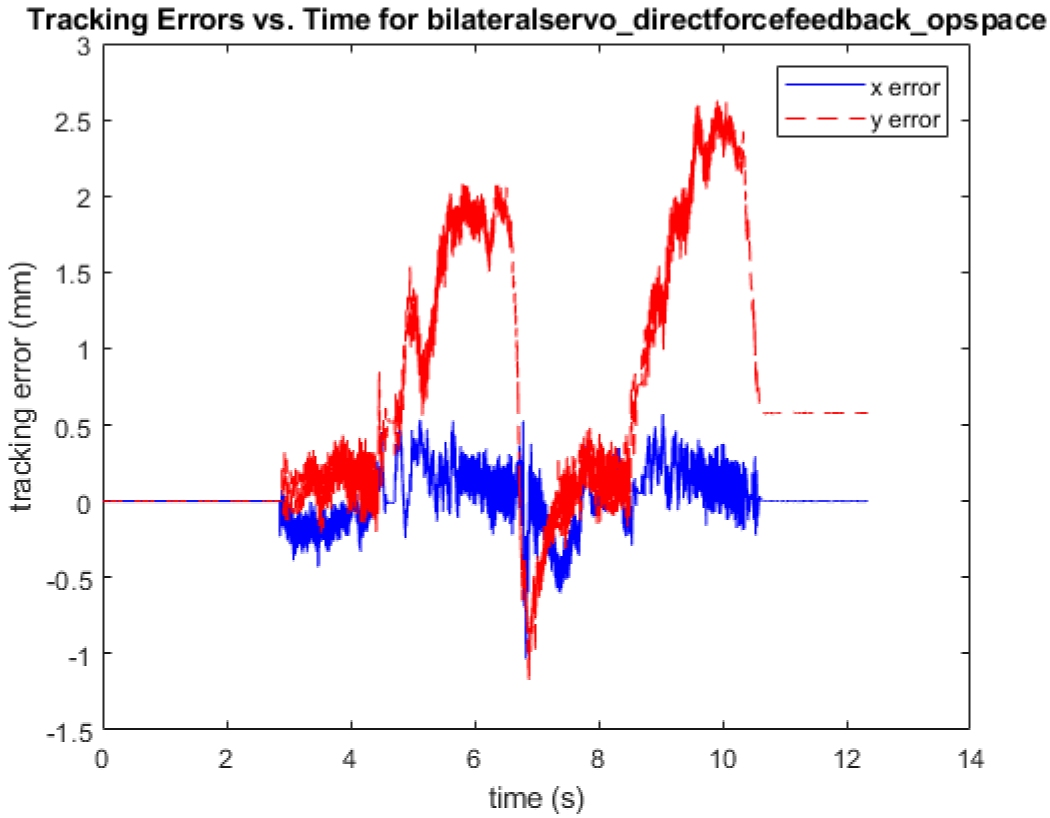


4.2. Code

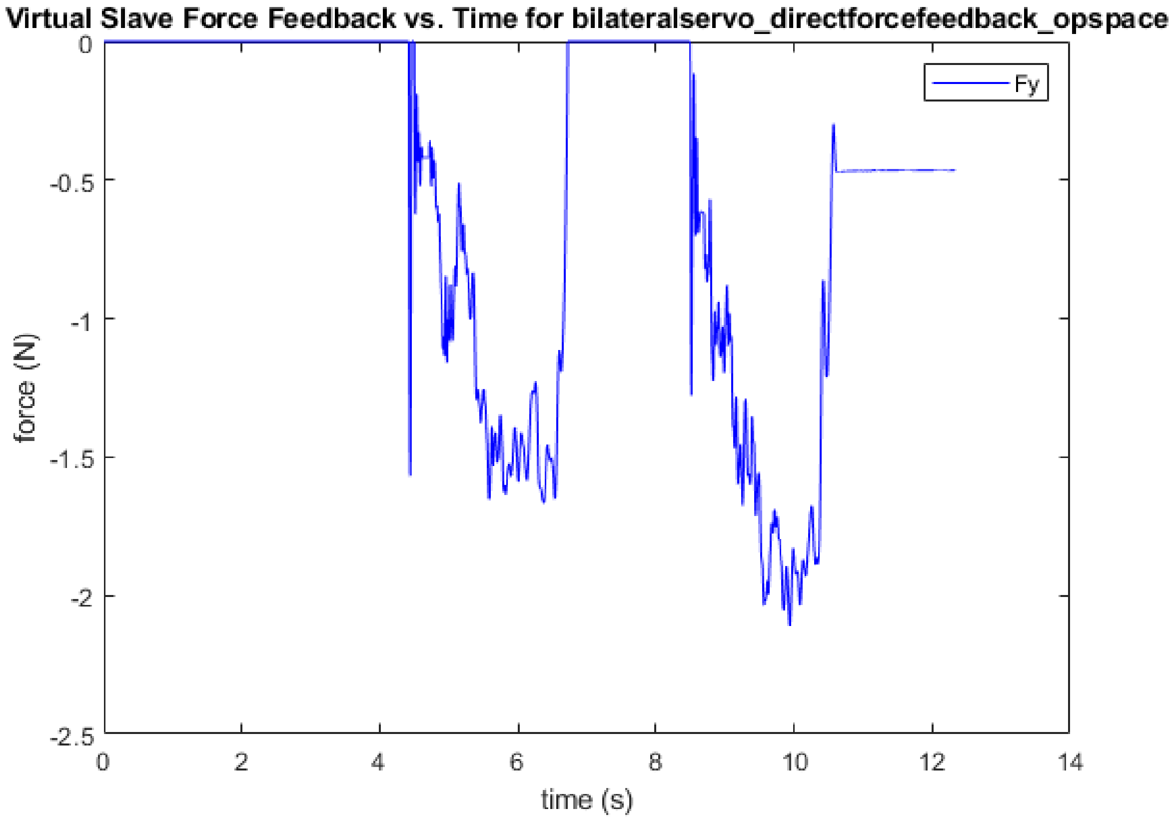
No additional MATLAB function blocks were needed for this controller that were not already described by the template or above.

4.3. Soft Wall (k_w = 1 N/mm)

4.3.1. Position Tracking Error

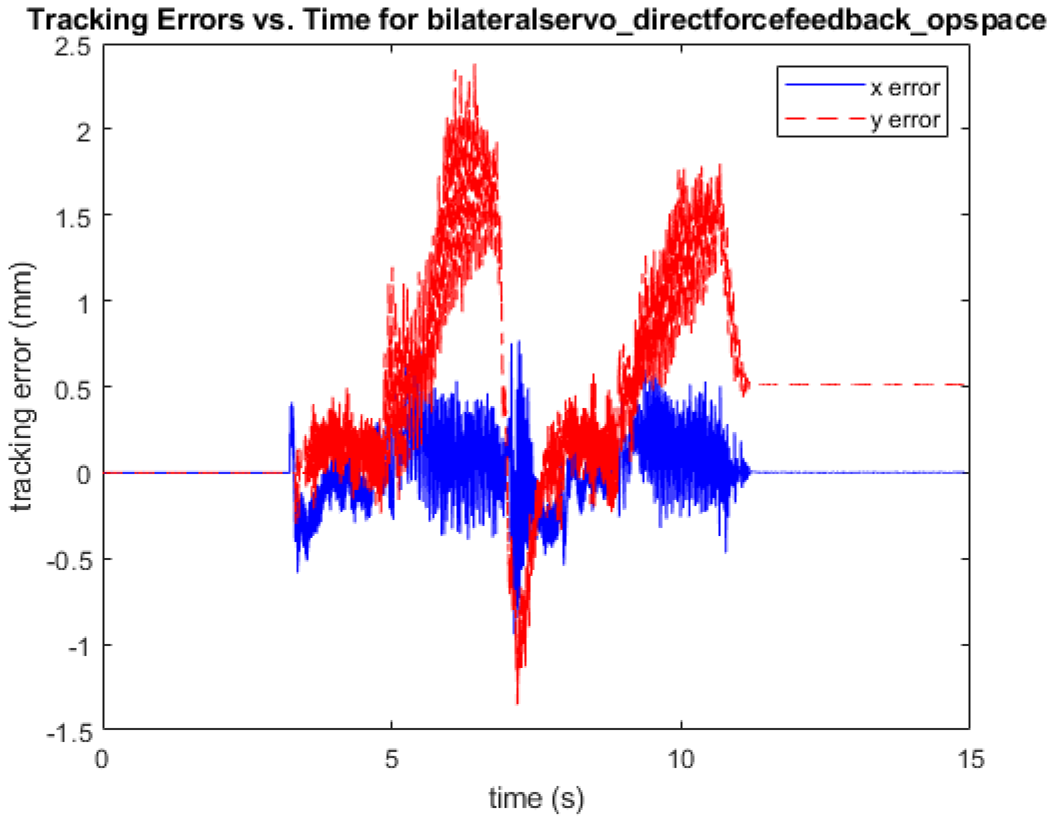


4.3.2. Force Feedback

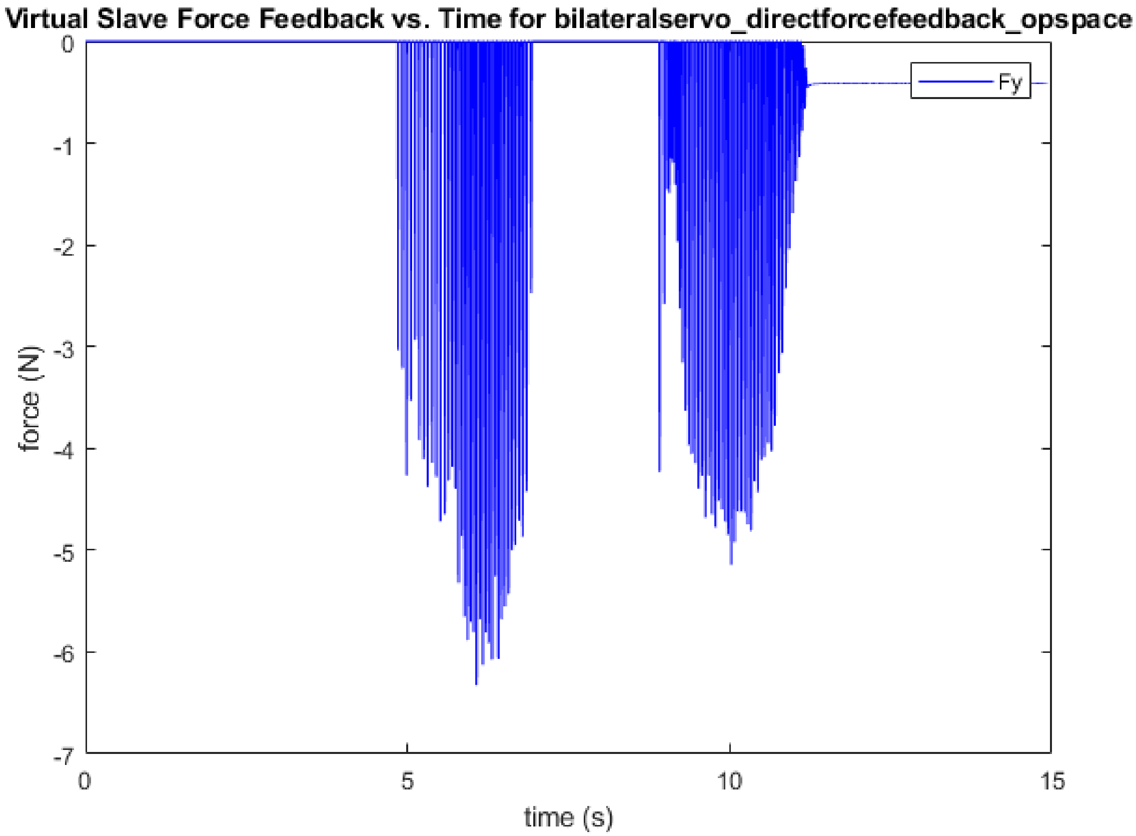


4.4. Hard Wall (k_w = 10 N/mm)

4.4.1. Position Tracking Error



4.4.2. Force Feedback



4.5. Analysis

The bilateral servo controller combined with force feedback was overall stable (but dependent on the gain of the force feedback) for both soft and hard virtual walls. With a force feedback gain of 1 the controller was as stable as the bilateral servo controller but had more of a transparent and realistic feeling contact with the virtual wall. The controller did seem to combine the stability of the bilateral servo and transparency of the direct force feedback. Another advantage of this controller was the ability to tune how much stability and conversely how much transparency the controller had. This could be done by changing the gain of the force feedback (k_f). I experimented with a high k_f value of 10 and the controller was closer to direct force feedback while with a k_f value of 0.3 the controller was close to the bilateral servo. This tunability allows for finding the optimal tradeoff between stability and transparency.

5. Appendix- Plotting Code (Used to make simulation plots)

```
%% ME EN 6230 Lab 6 Ryan Dalby
% close all;
set(groot, 'DefaultTextInterpreter', 'none') % Prevents underscore from
becoming subscript

% Note that for this lab the Forces block was renamed to forces and the
% output was changed to a structure with time

% Extract necessary data, will error if the data does not exist
time = forces.time; % s
model_title = extractBefore(forces.blockName, "/");
tracking_errors = (position_errors.signals.values)*1000; % mm
force_feedback = forces.signals.values; % N

% Plot Tracking Errors
figure;
plot(time, tracking_errors(:,1), 'b-');
hold on;
plot(time, tracking_errors(:,2), 'r--');
title(strcat("Tracking Errors vs. Time for ", model_title));
xlabel("time (s)");
ylabel("tracking error (mm)");
legend("x error", "y error");

% Plot Virtual Slave Force Feedback
figure;
plot(time, force_feedback(:,2), 'b-');
title(strcat("Virtual Slave Force Feedback vs. Time for ", model_title));
xlabel("time (s)");
ylabel("force (N)");
legend("Fy");
```