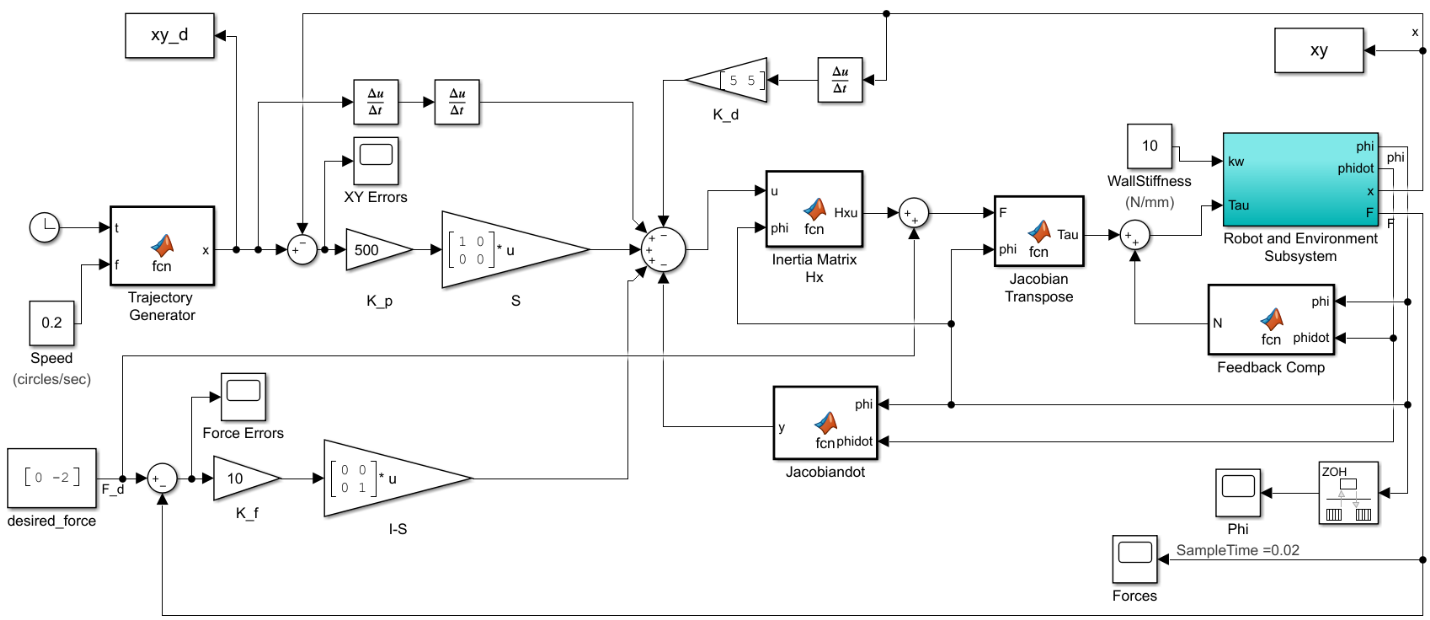
ME EN 6230
Problem Set 10
Ryan Dalby

Note: All simulations were run with a trajectory speed of 0.2 circle/s with a stop time of 10 seconds.

1. Hybrid Position/Force Control

1.1. Hybrid Position/Force Control with P Force Control

1.1.1. Model



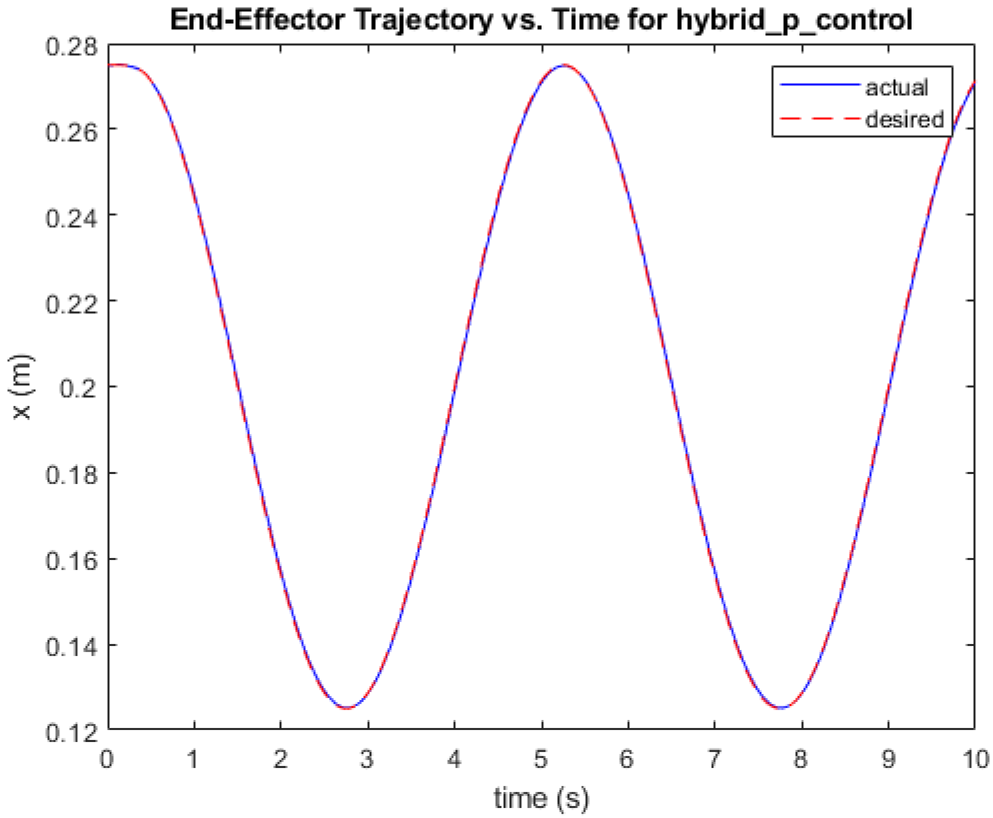
Parameter values: K_p=500, K_f=10, K_{dx}=K_{dy}=5

1.1.2. Code

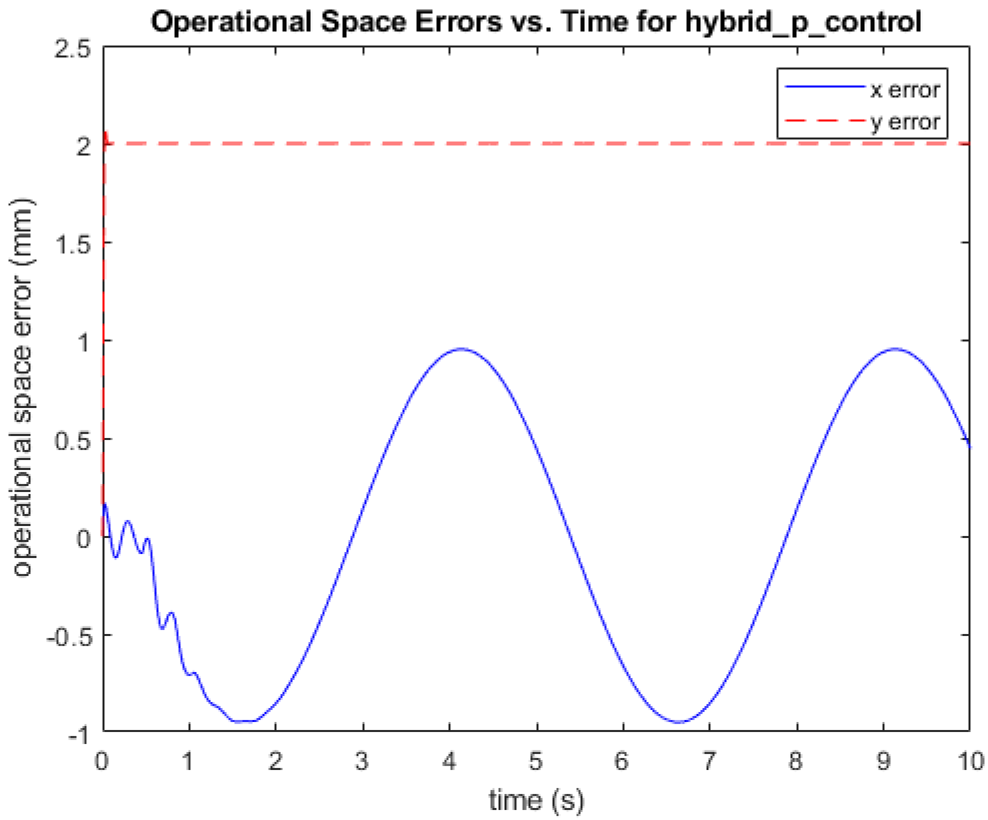
No additional MATLAB function blocks were needed for this controller that were not already described by the template or above.

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

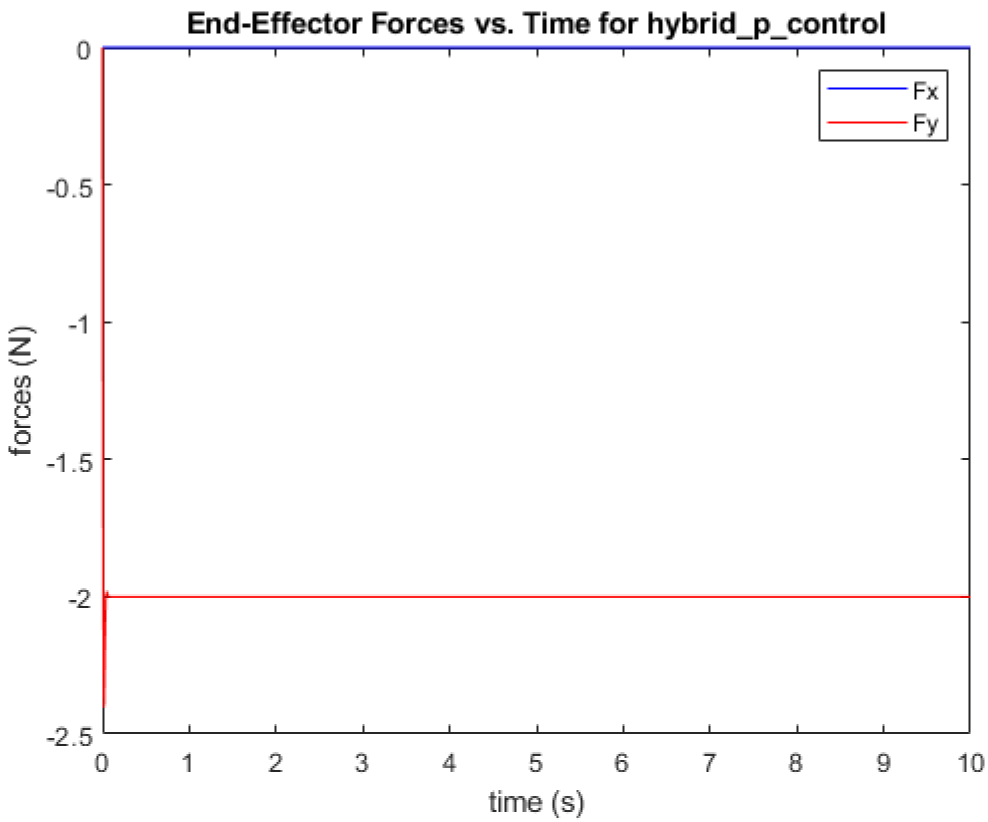
1.1.3.1. End-Effector Trajectory



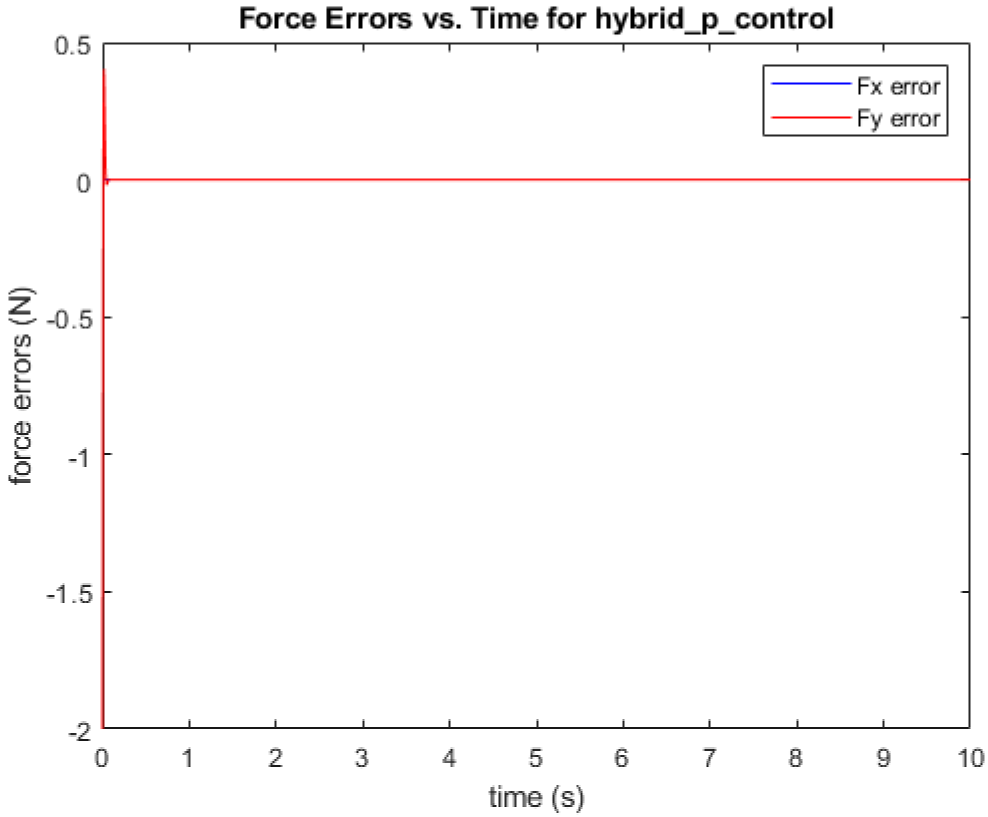
1.1.3.2. Operational Space Errors



1.1.3.3. End-Effector Forces

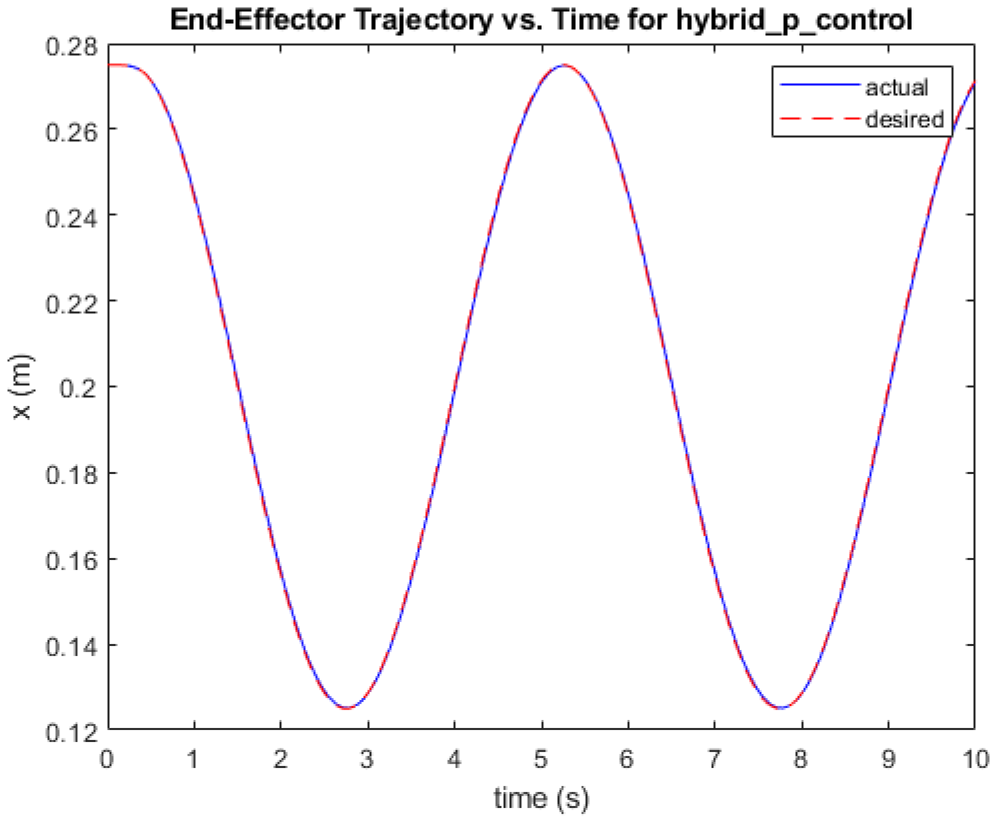


1.1.3.4. Force Errors

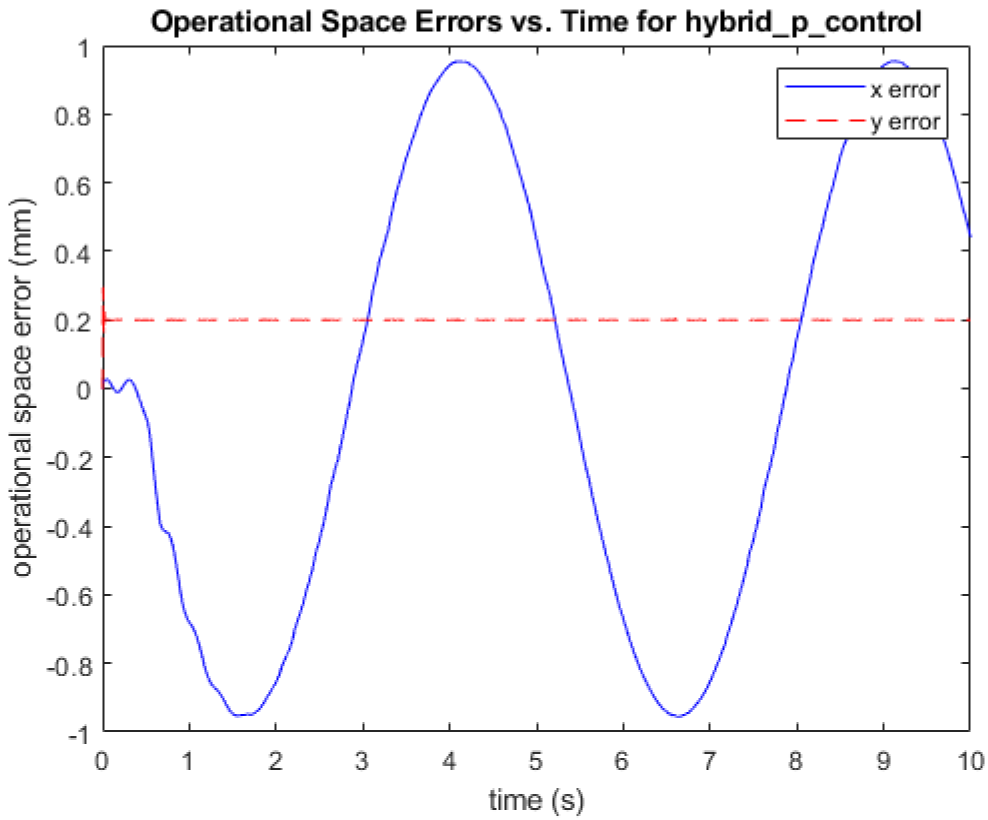


1.1.4. Hard Wall (k_w=10N/mm)

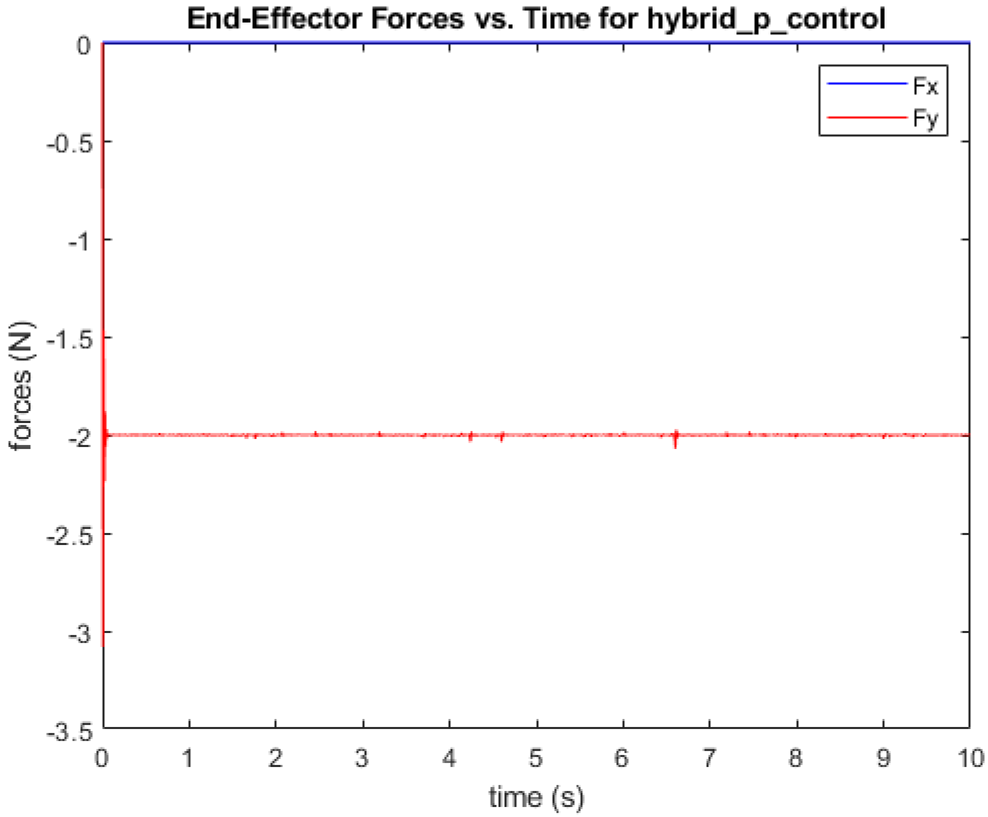
1.1.4.1. End-Effector Trajectory



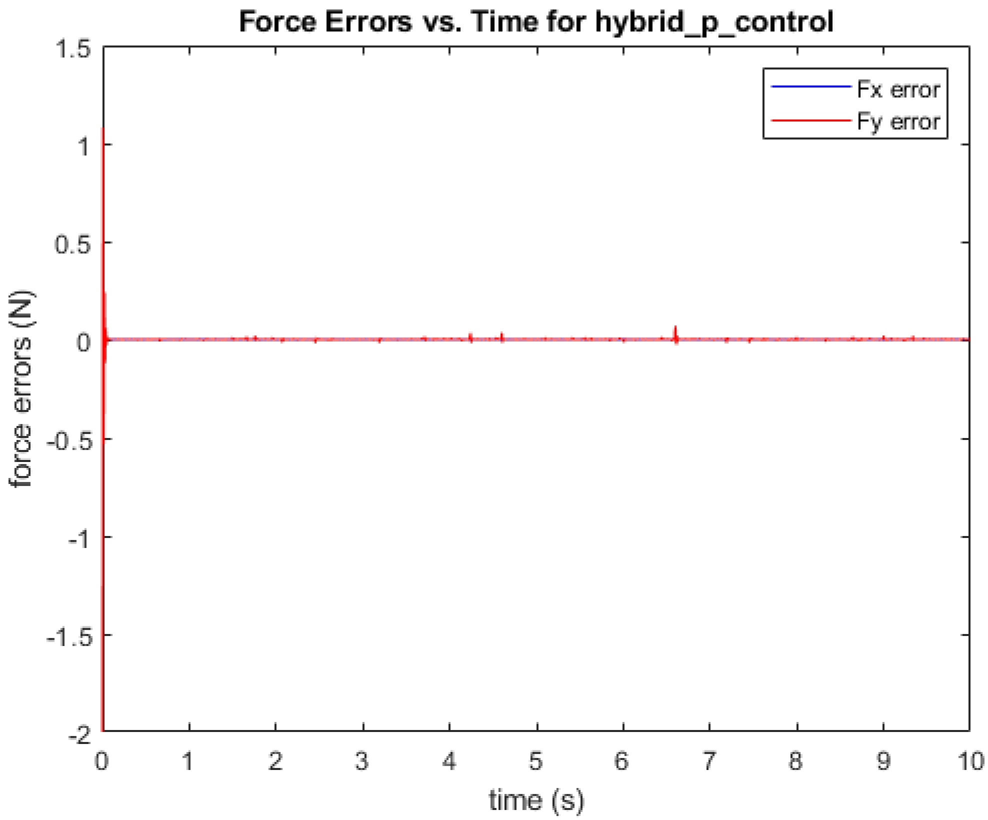
1.1.4.2. Operational Space Errors



1.1.4.3. End-Effector Forces



1.1.4.4. Force Errors

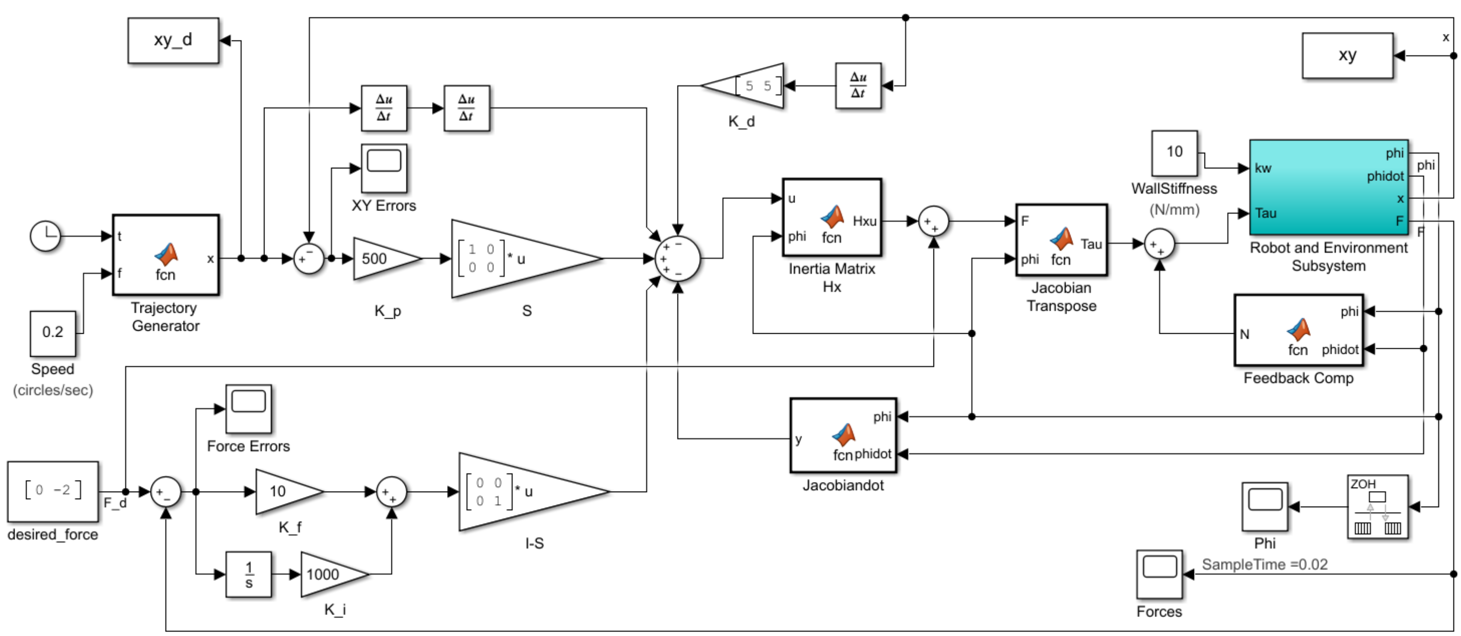


1.1.5. Analysis

With P force control alone the hybrid position/force controller does a good job at tracking the x-trajectory and maintaining the desired 2N force. The controller does have some minor force error fluctuations when on the hard surface. Overall, since there is no disturbance the P force control can sufficiently track force and position.

1.2. Hybrid Position/Force Control with PI Force Control

1.2.1. Model



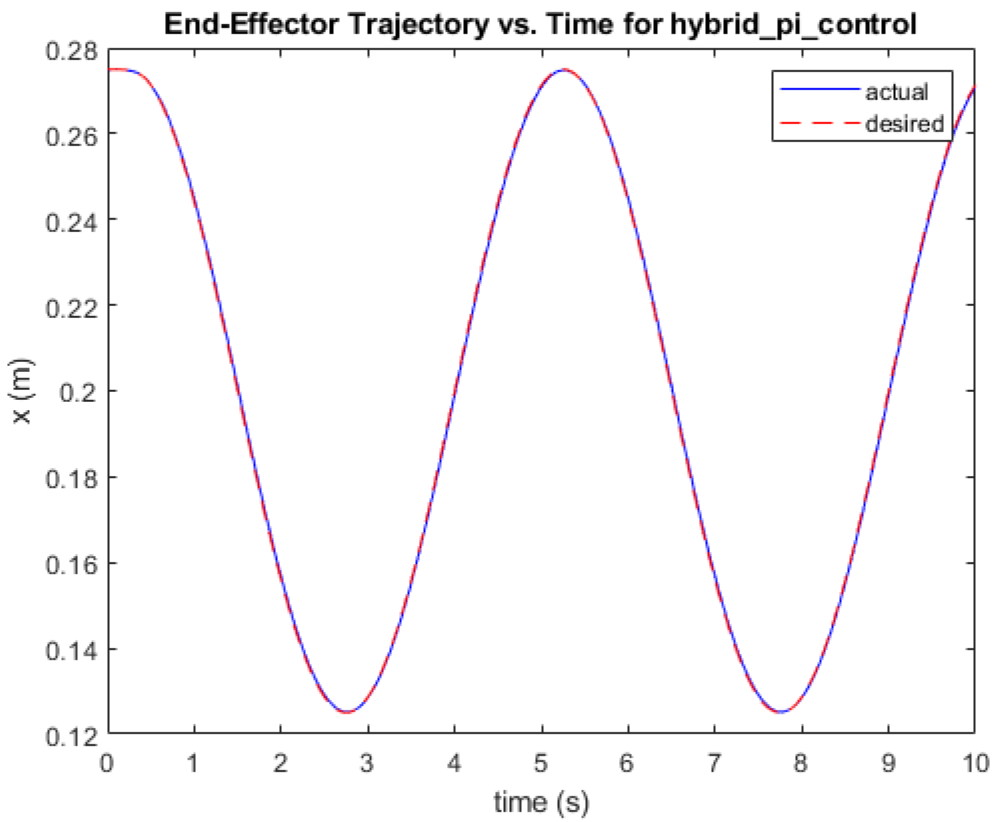
Parameter values: K_p=500, K_i=1000, K_f=10, K_{dx}=K_{dy}=5

1.2.2. Code

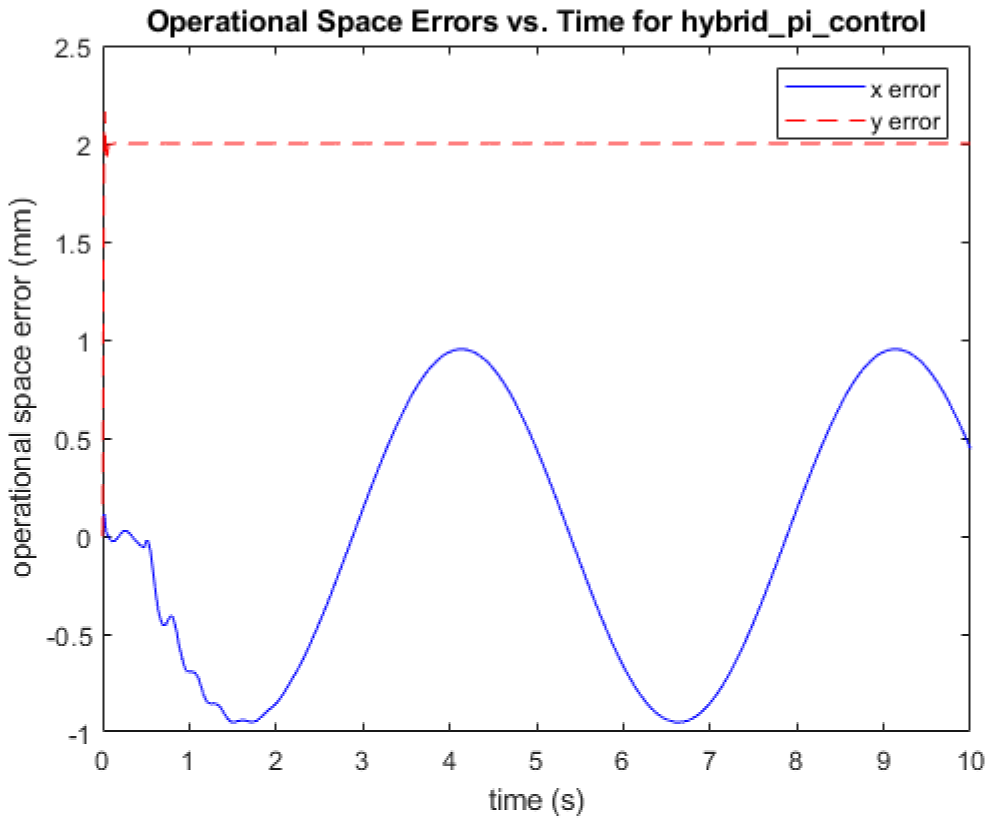
No additional MATLAB function blocks were needed for this controller that were not already described by the template or above.

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

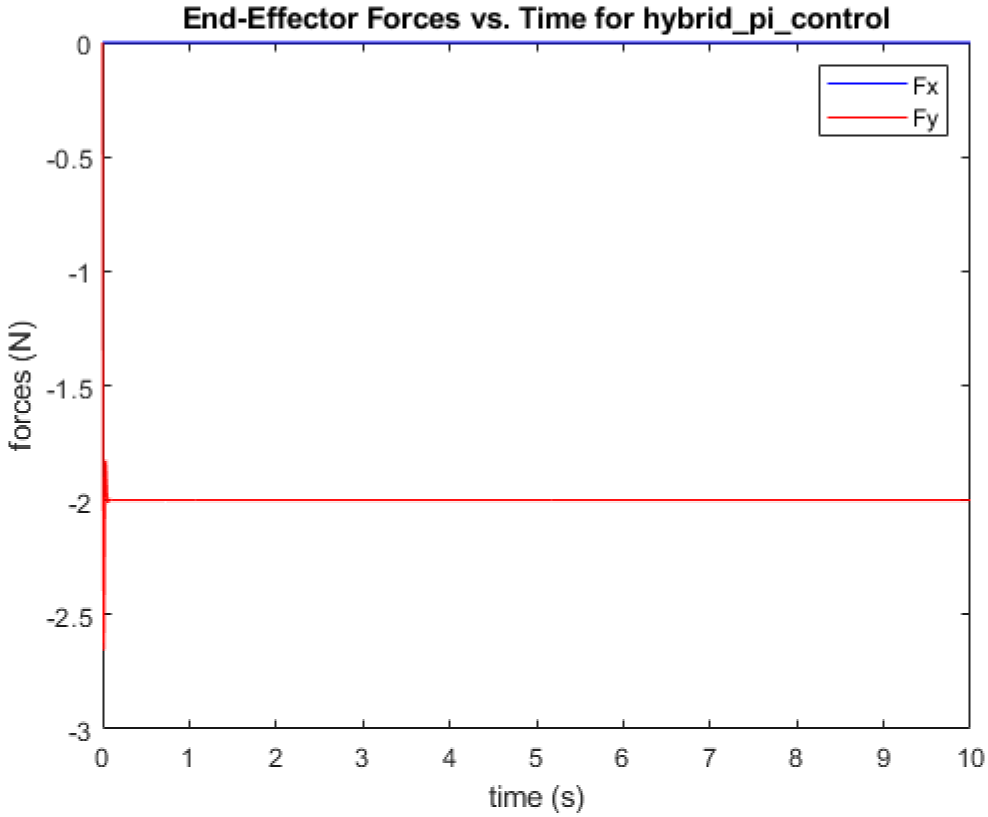
1.2.3.1. End-Effector Trajectory



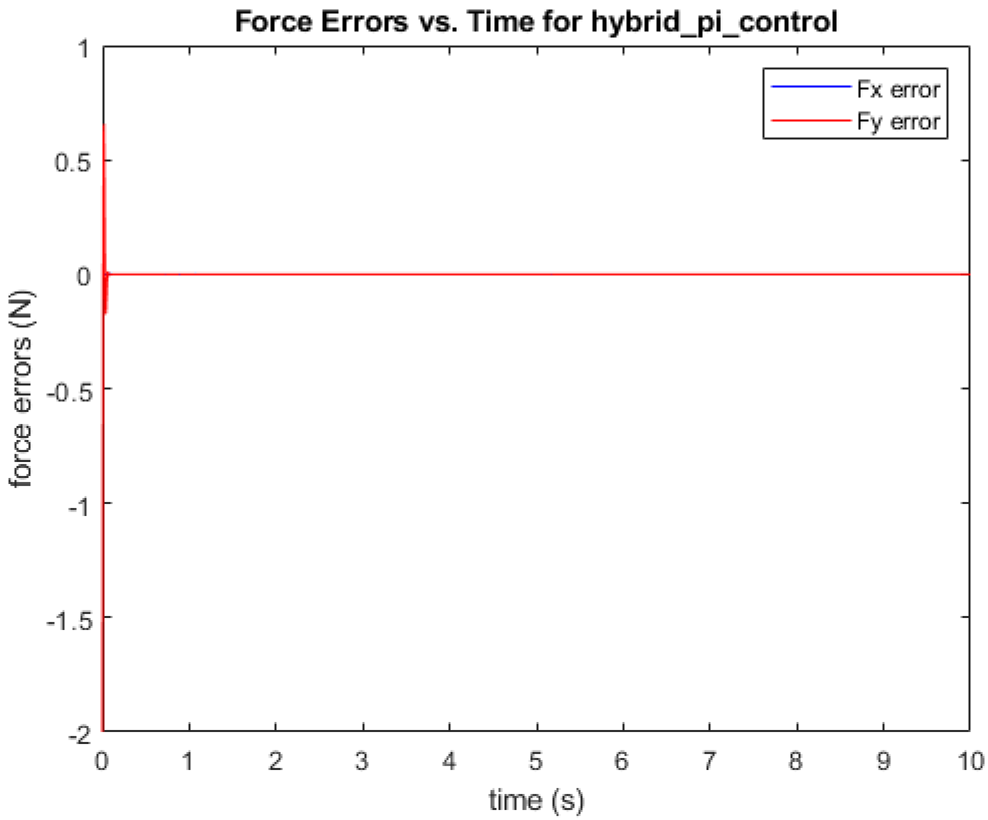
1.2.3.2. Operational Space Errors



1.2.3.3. End-Effector Forces

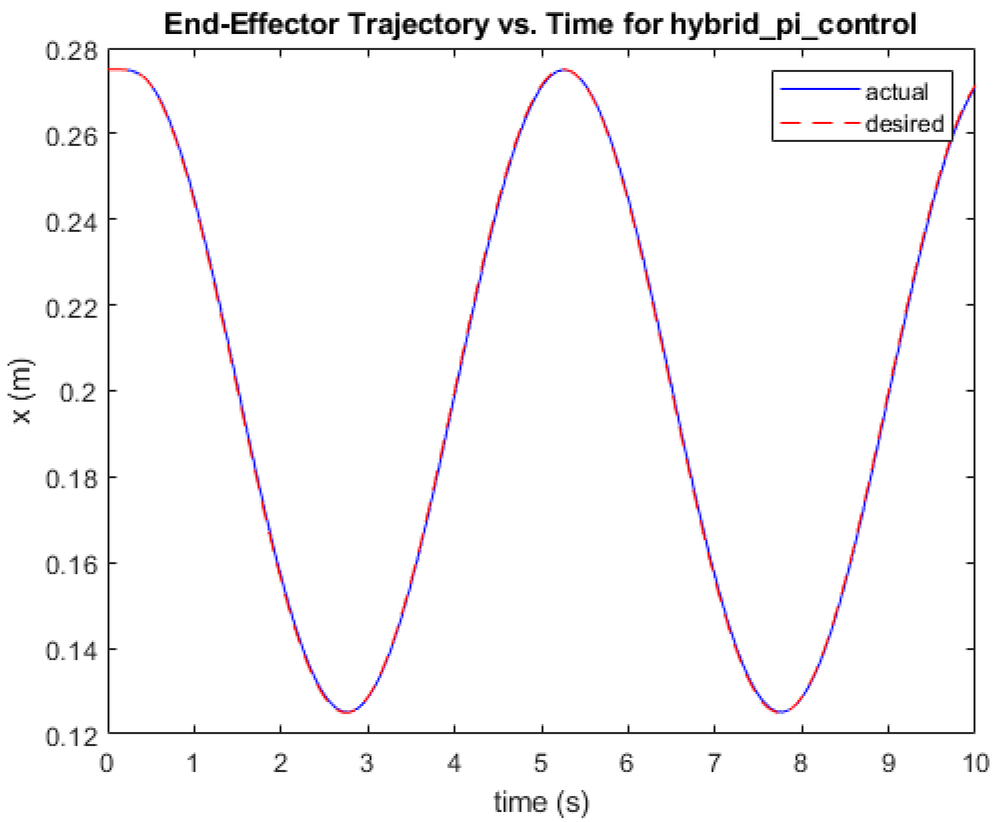


1.2.3.4. Force Errors

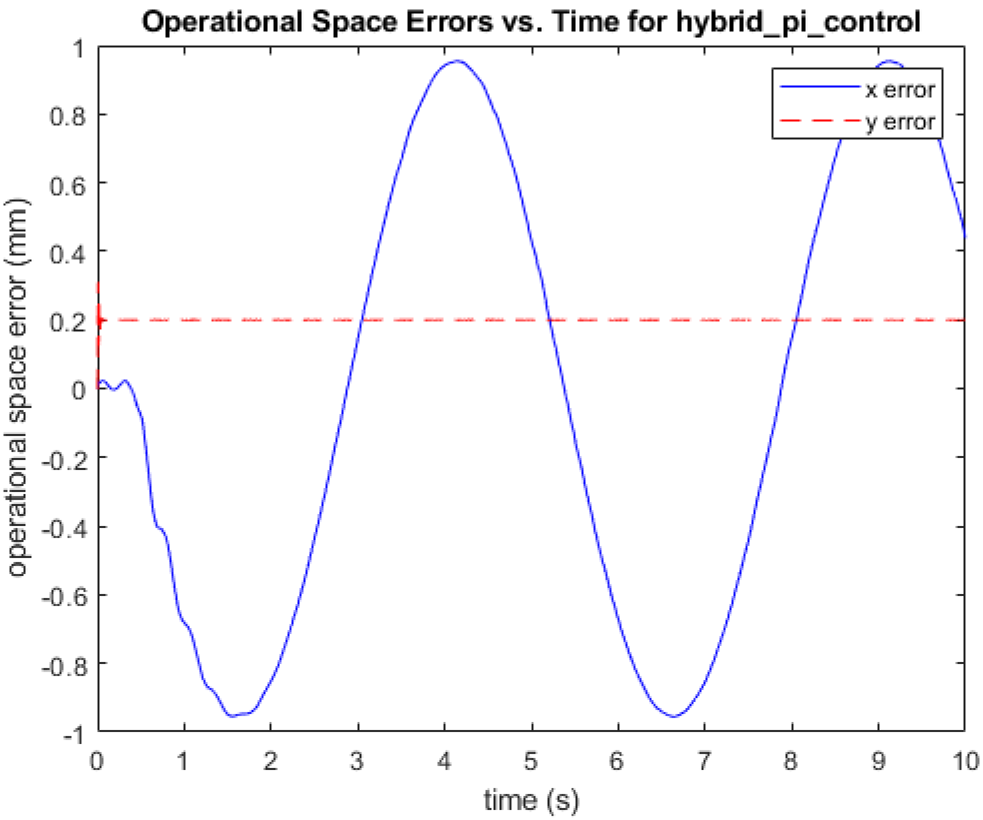


1.2.4. Hard Wall (k_w=10N/mm)

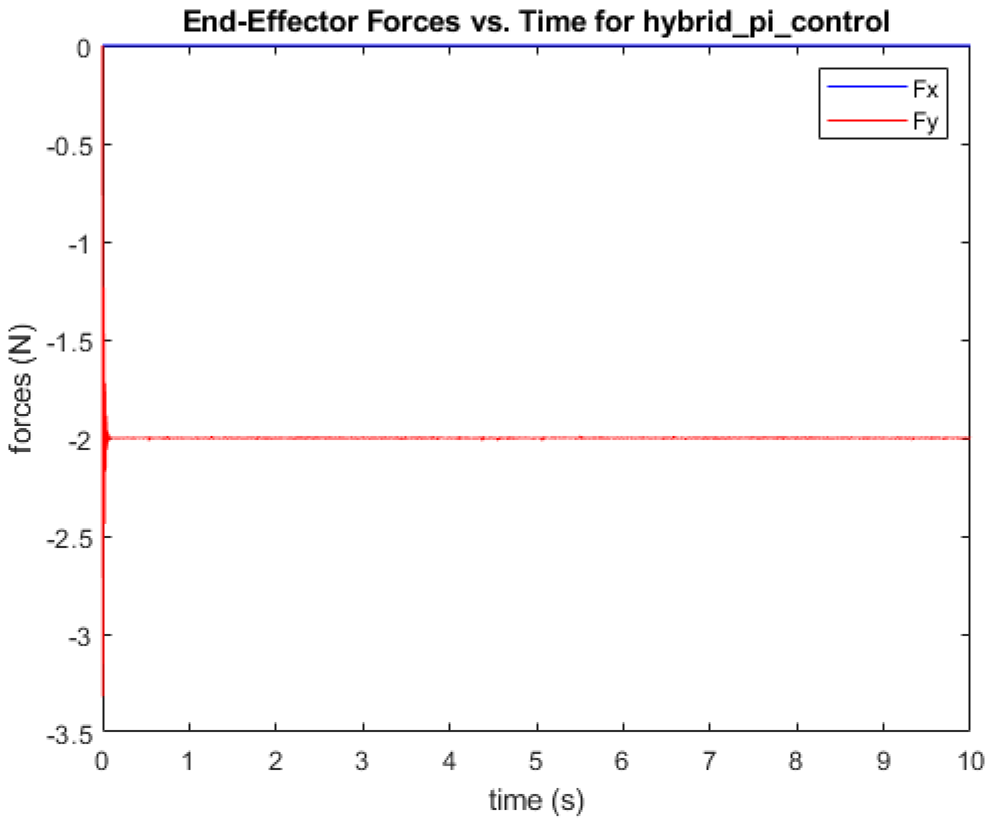
1.2.4.1. End-Effector Trajectory



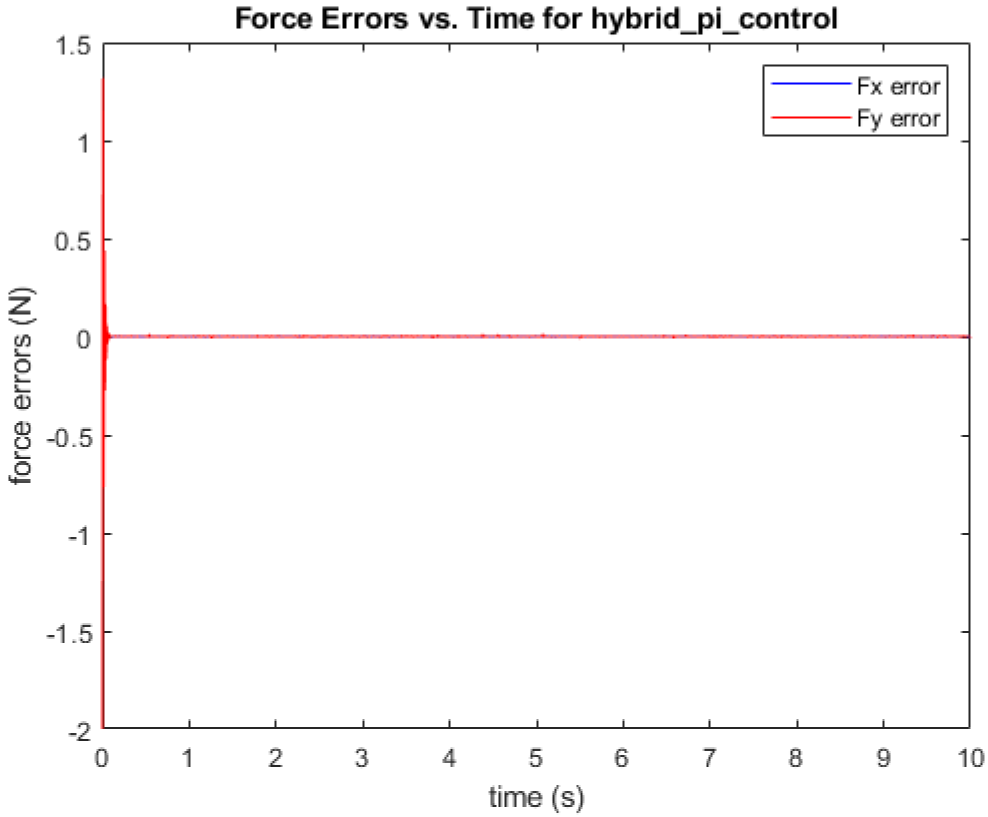
1.2.4.2. Operational Space Errors



1.2.4.3. End-Effector Forces



1.2.4.4. Force Errors



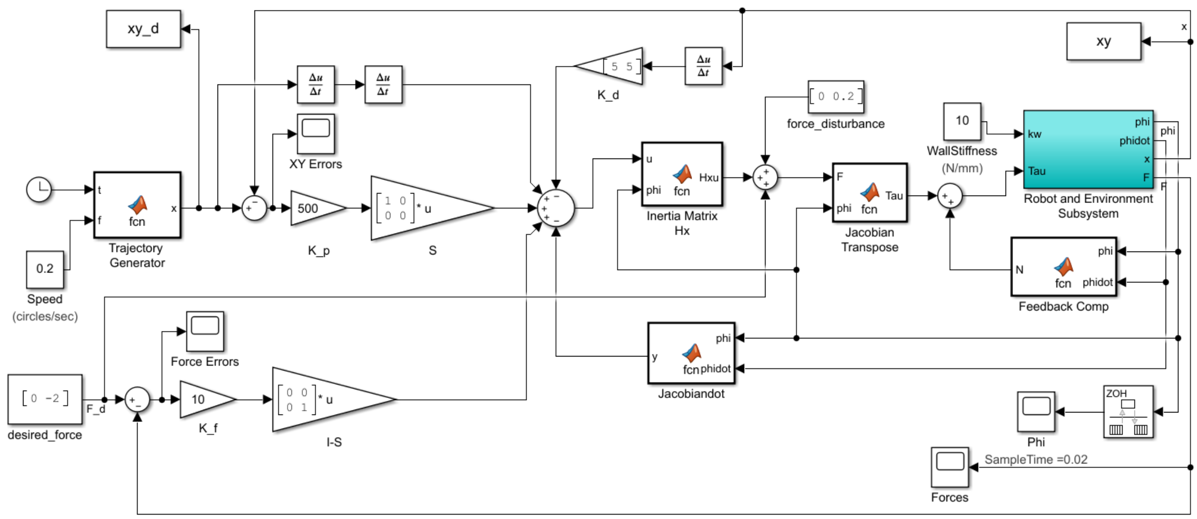
1.2.5. Analysis

With P and I force control the hybrid position/force controller does an even better job at tracking the x-trajectory and maintaining the desired 2N force than P force control alone. The controller manages to eliminate the minor force error fluctuations when on the hard surface which is the main improvement over the P force controller. Overall, with no disturbances the tracking of both force and position is good.

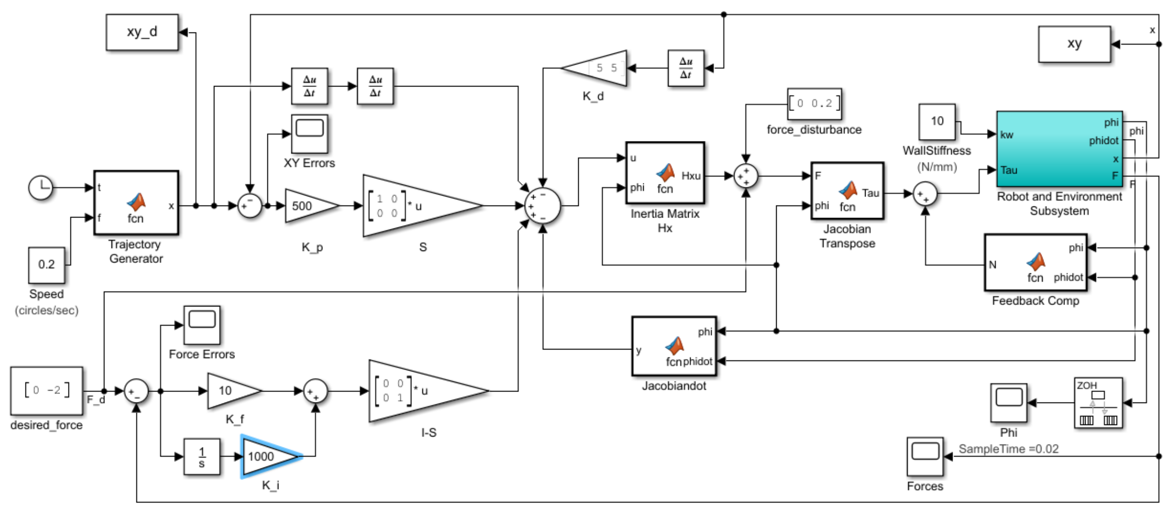
1.3. Hybrid Position/Force Control with Disturbance Comparison

1.3.1. Models

P Control with Disturbance:



PI Control with Disturbance:



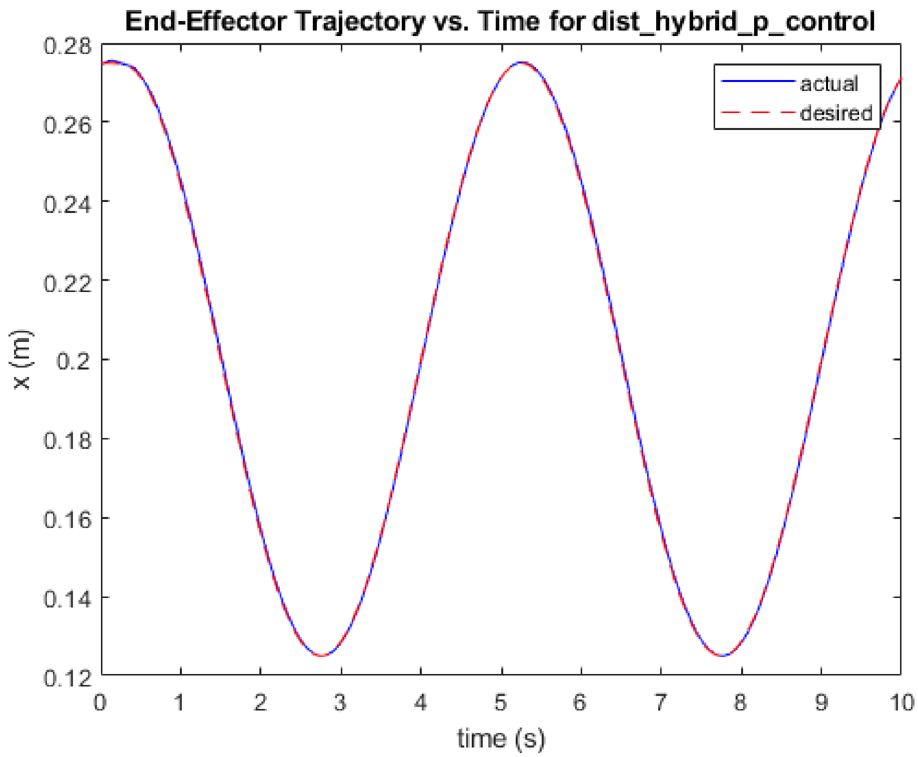
Parameter values: Same as corresponding P and PI control above (1.1 and 1.2 respectively)

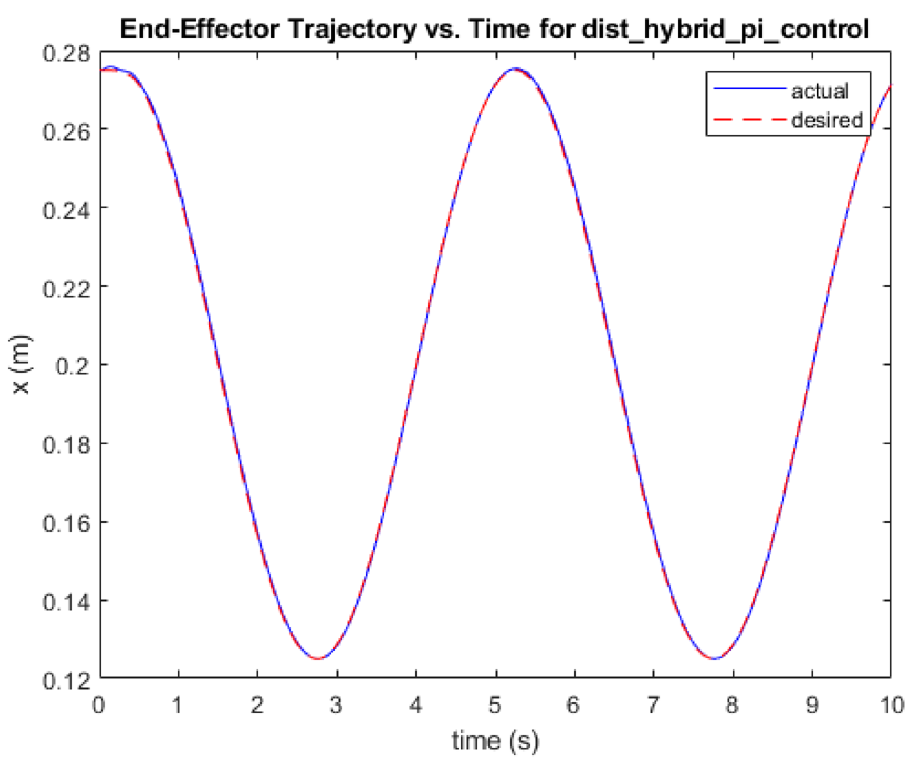
1.3.2. Code

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

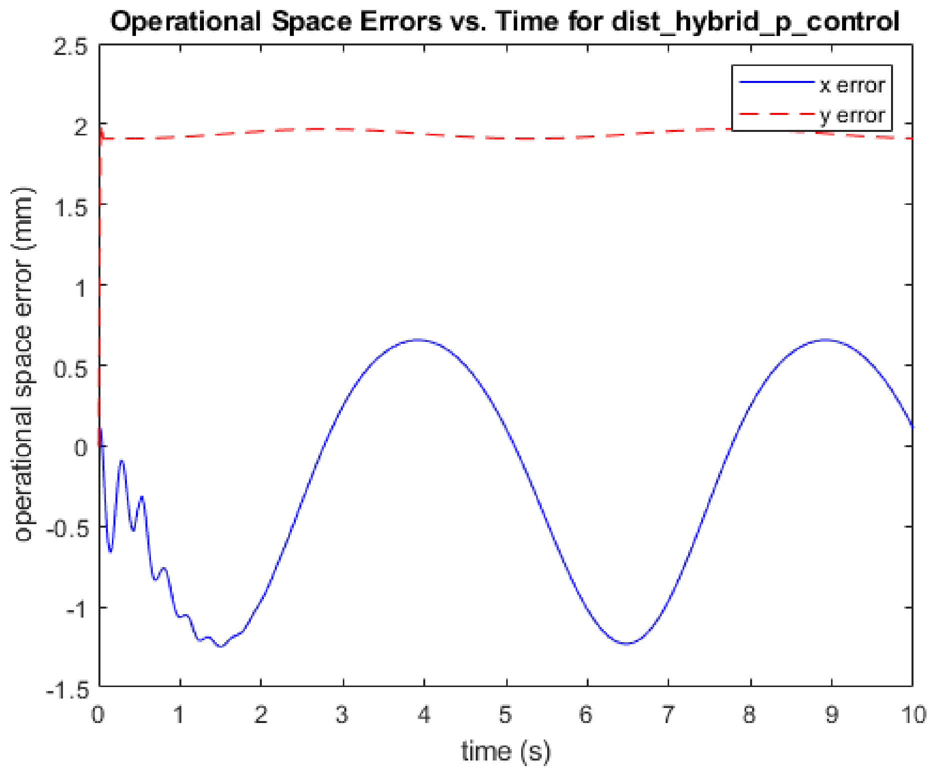
1.3.3. Soft Wall (k_w=1N/mm)

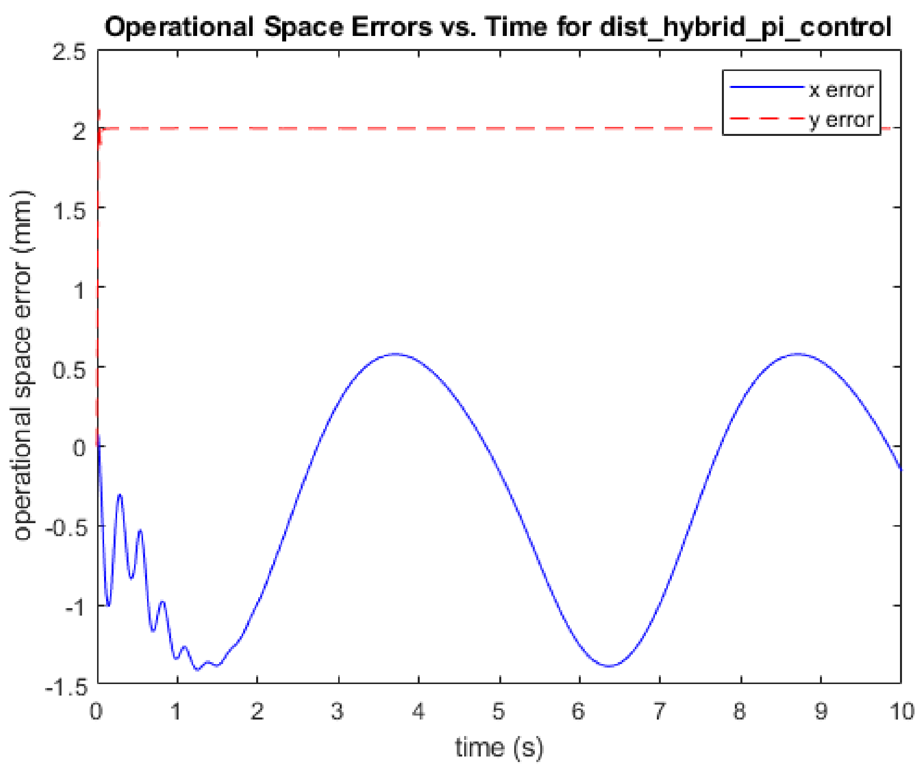
1.3.3.1. End-Effector Trajectories



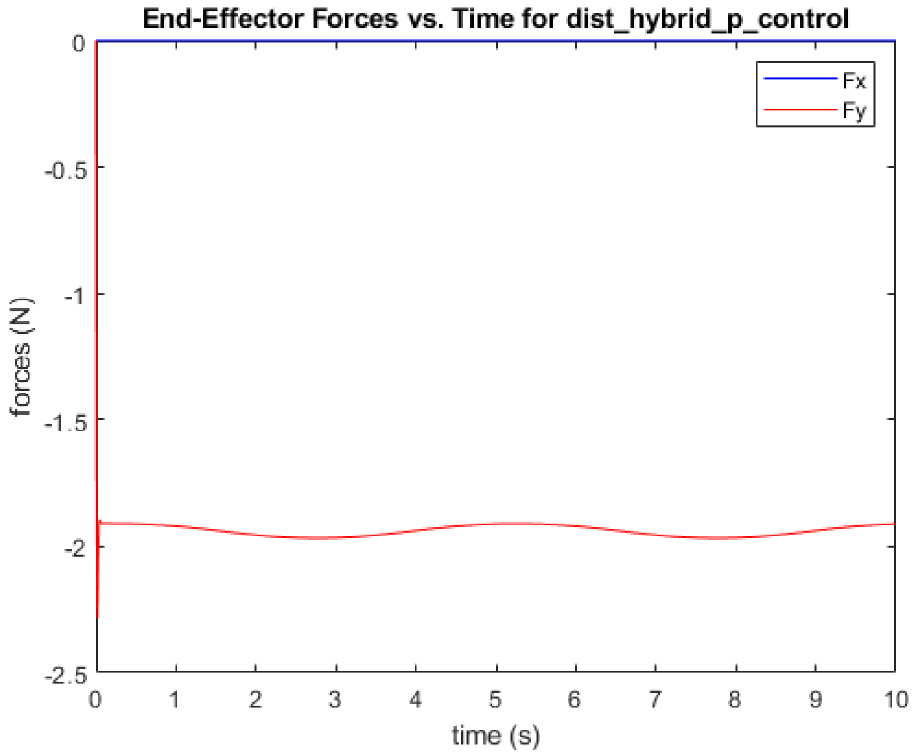


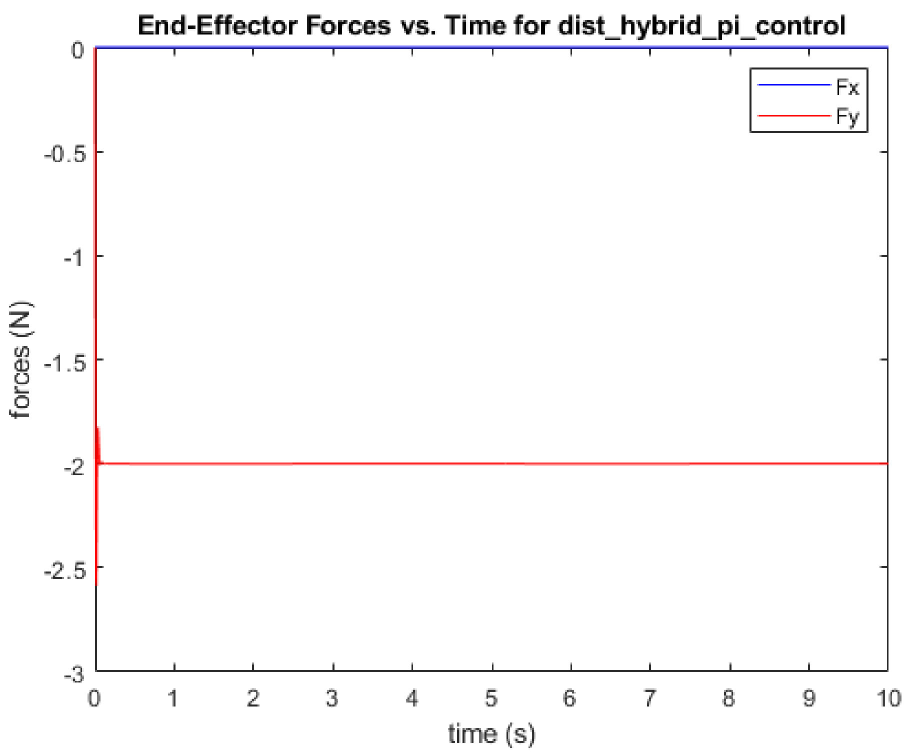
1.3.3.2. Operational Space Errors



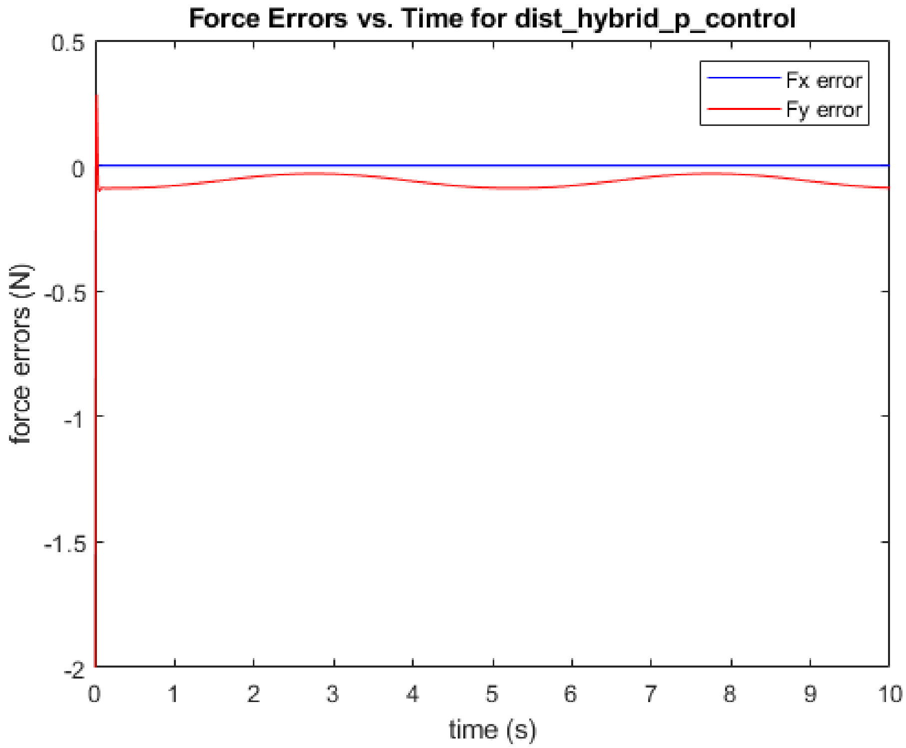


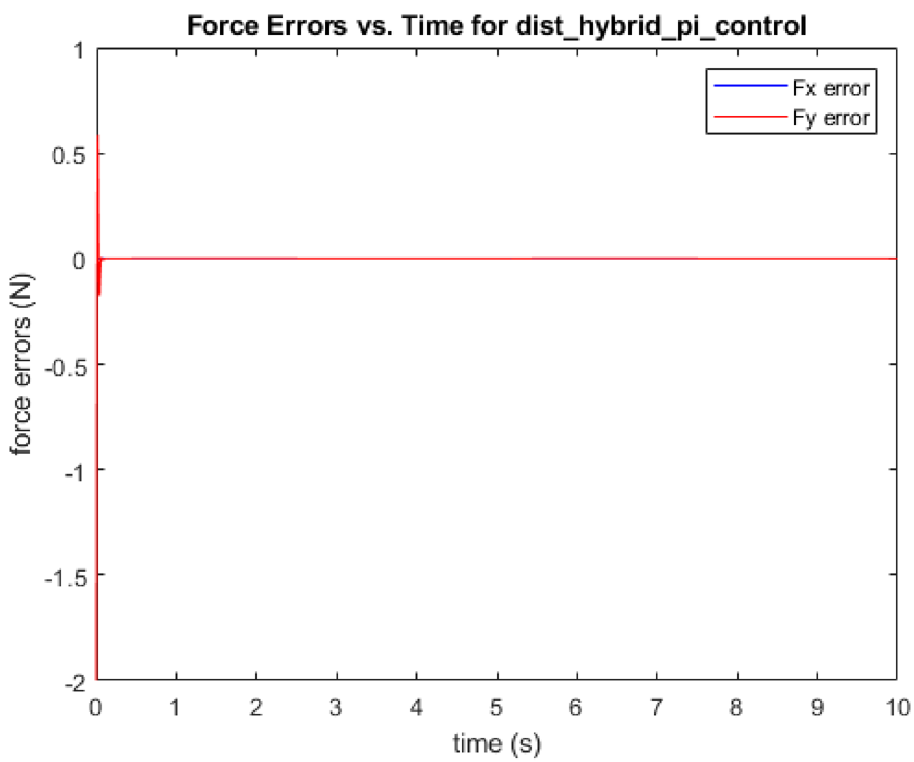
1.3.3.3. End-Effector Forces





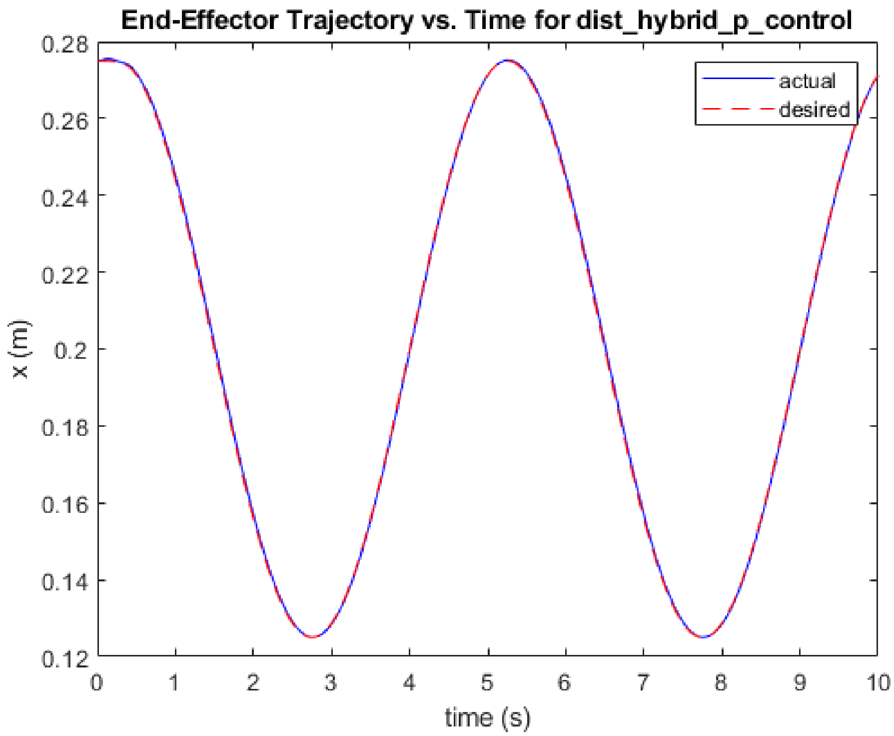
1.3.3.4. Force Errors

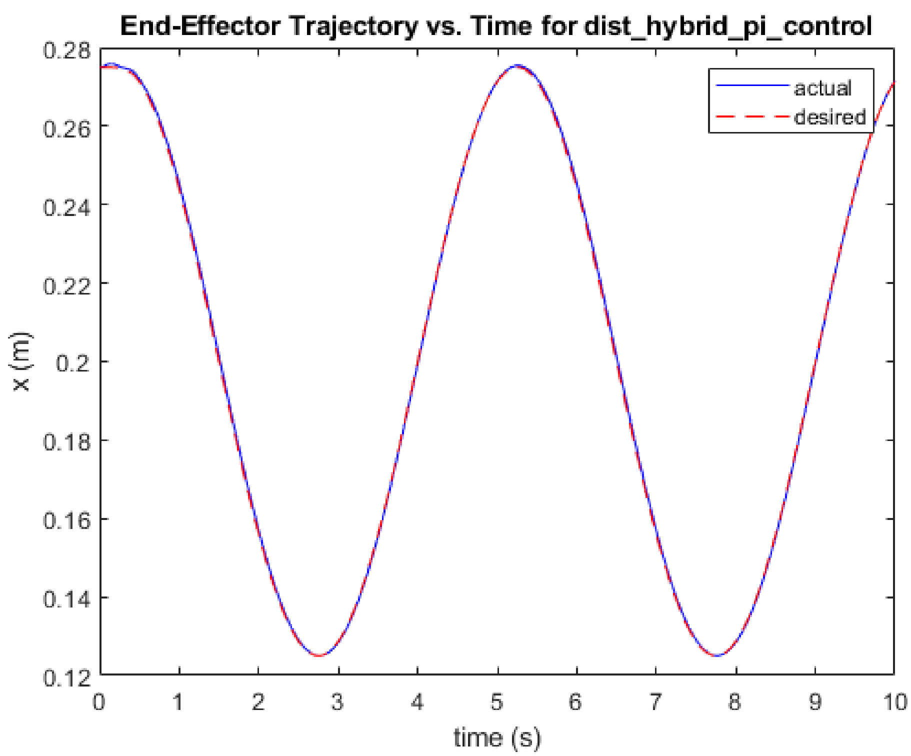




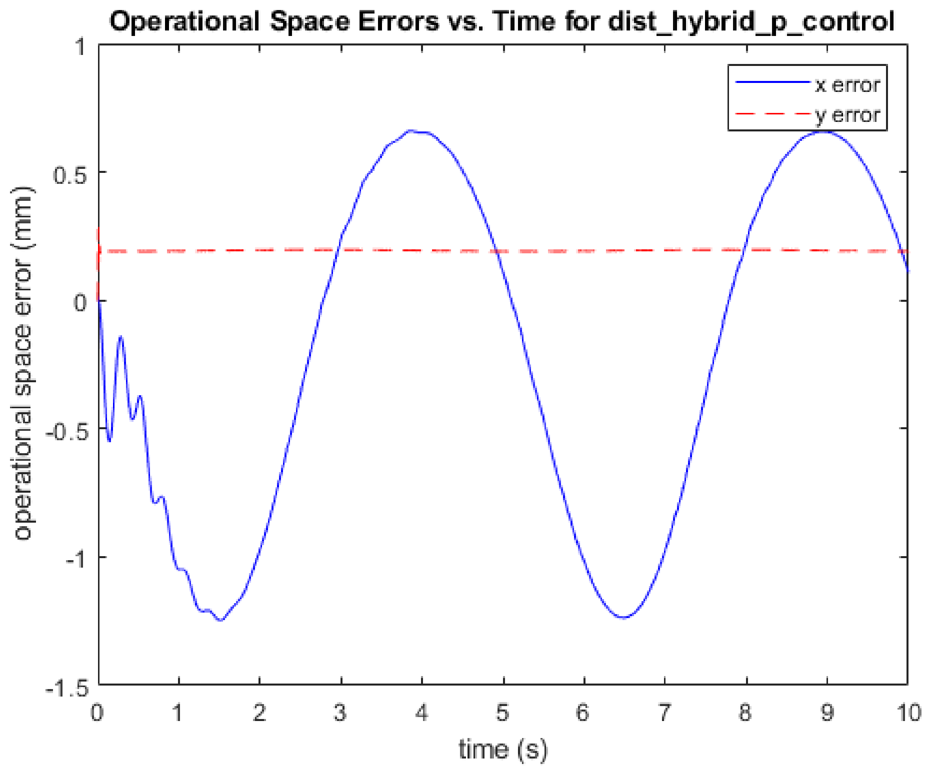
1.3.4. Hard Wall (k_w=10N/mm)

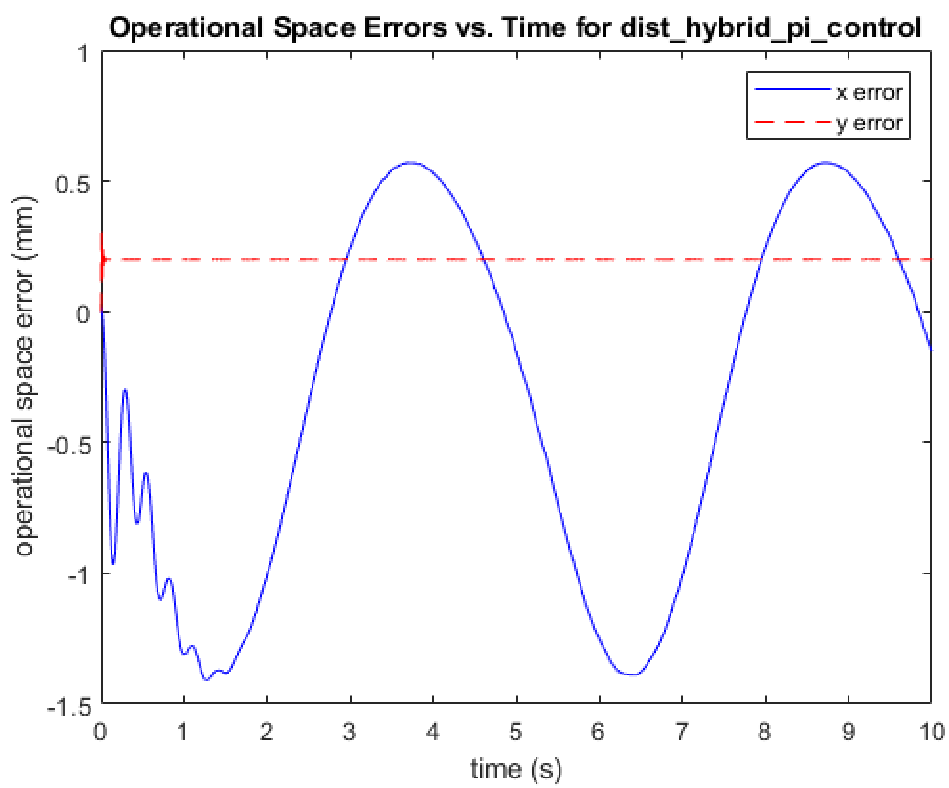
1.3.4.1. End-Effector Trajectories



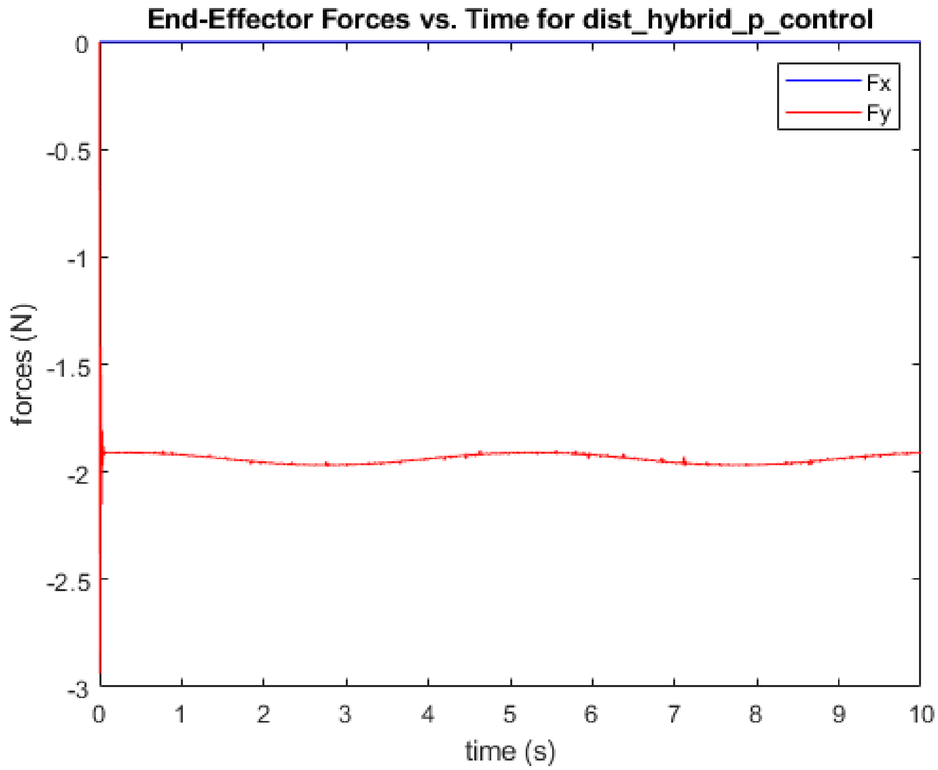


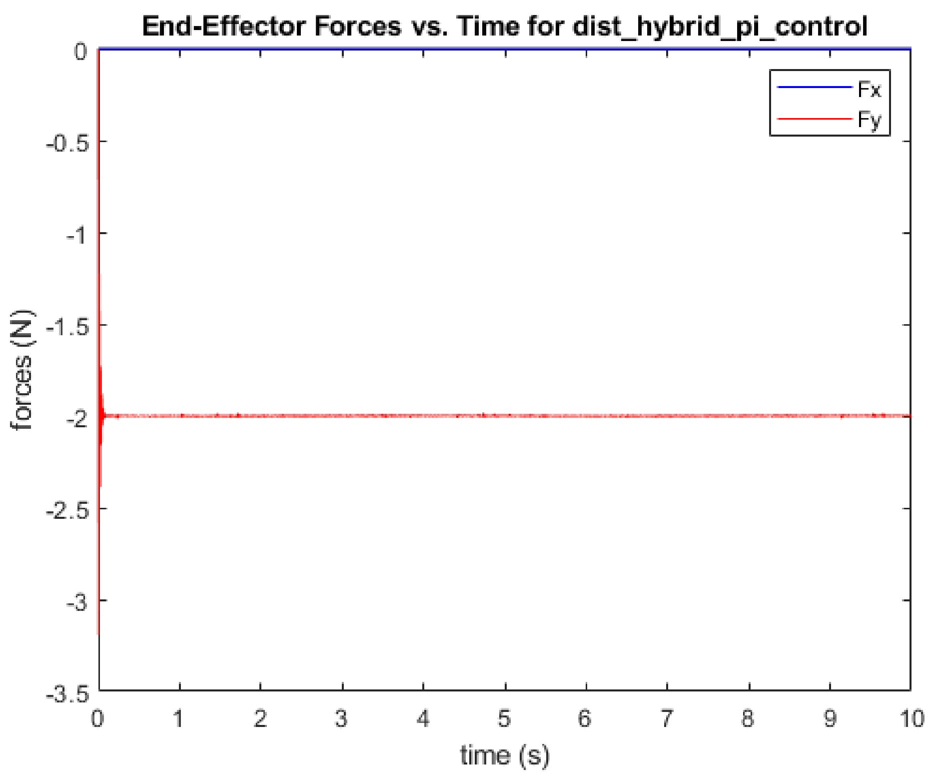
1.3.4.2. Operational Space Errors



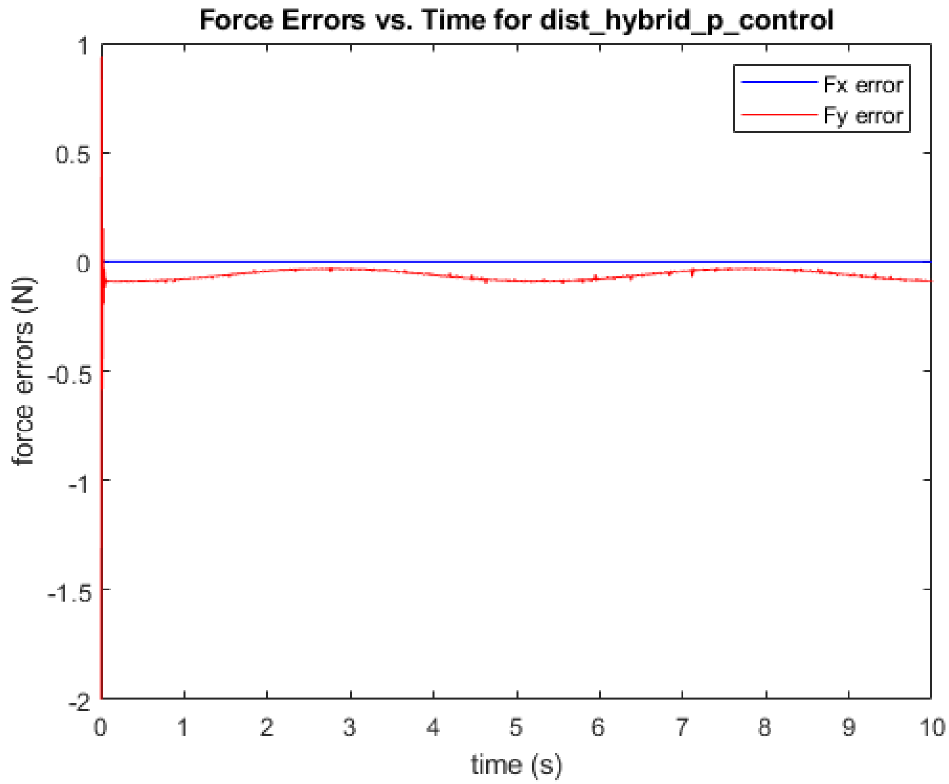


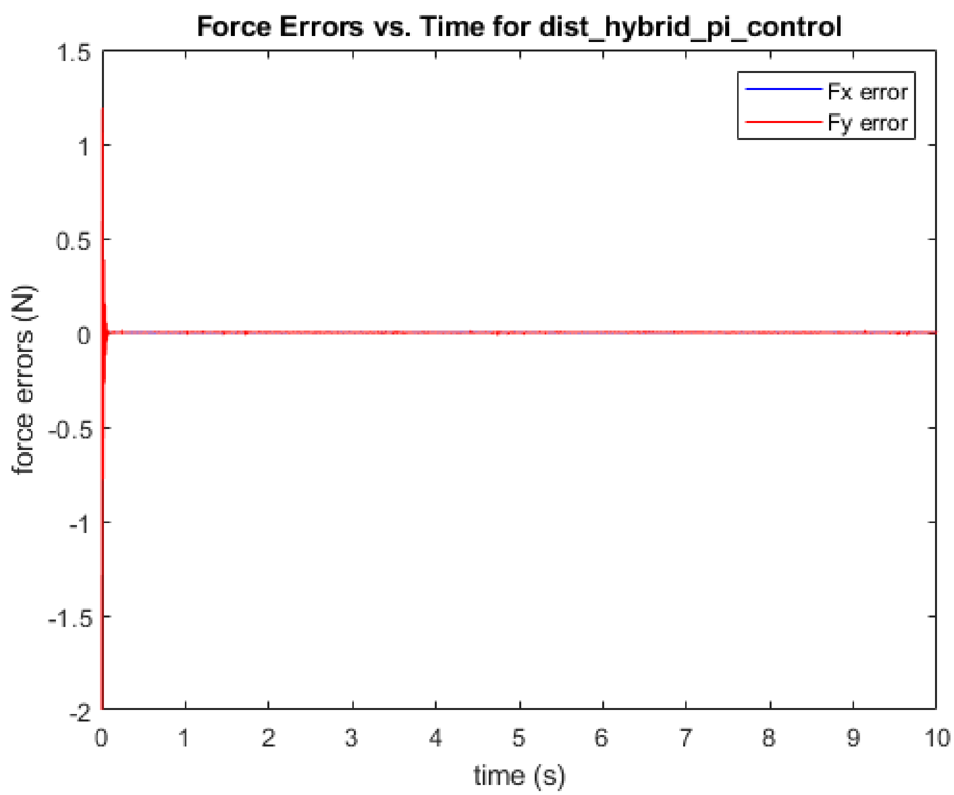
1.3.4.3. End-Effector Forces





1.3.4.4. Force Errors





1.3.5. Comparison

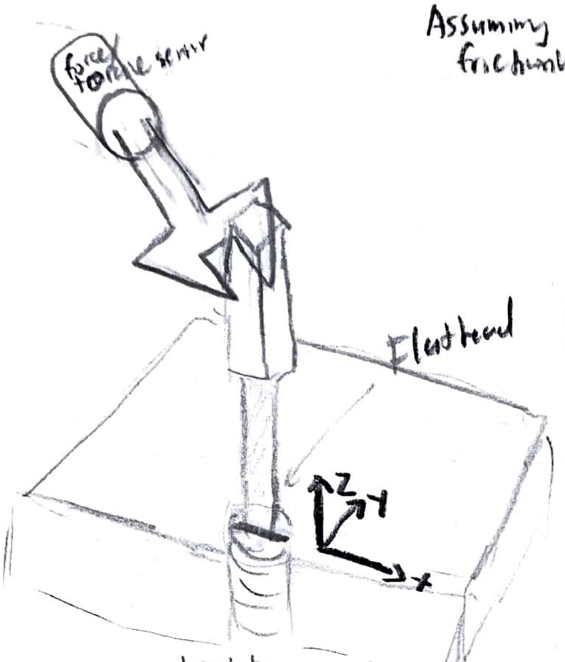
When comparing the hybrid position/force controllers with a disturbance it becomes clear how much better the controller with PI force control is compared to the controller with just P force control. This is evidenced by looking at the force errors for both the soft and hard surfaces. With just proportional (P) control the controller has a lot of steady-state force error along with some perturbations in the force error. Adding integral (I) control virtually eliminates both the steady-state force error and the perturbations in the force error. This is because the integral control allows the controller to compensate for the built up error that comes from the disturbance for which proportional control cannot deal with. In terms of tracking position, the two controllers are virtually the same (other than some influence from the change in position that is a result of the forces at the beginning) since the controllers are the same for position tracking in the x-direction. Overall, the hybrid PI controller is more robust to force disturbances than the hybrid P controller.

Appendix- Plotting Code (Used to make simulation plots)

```
%% ME EN 6230 Problem Set 10 Ryan Dalby
% close all;
set(groot, 'DefaultTextInterpreter', 'none') % Prevents underscore from
becoming subscript

% Extract necessary data, will error if the data does not exist
time = xy_errors.time; % s
model_title = extractBefore(xy_errors.blockName, "/");
actual_trajectory = xy; % m
desired_trajectory = xy_d; % m
opspace_errors = xy_errors.signals.values*1000; % mm
forces = F.signals.values; % N
force_errors = F_errors.signals.values; % N

% Plot End-Effector Trajectory
figure;
plot(time, xy(:,1), 'b-');
hold on;
plot(time, xy_d(:,1), 'r--');
title(strcat("End-Effector Trajectory vs. Time for ", model_title));
xlabel("time (s)");
ylabel("x (m)");
legend("actual", "desired");

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


% Plot End-Effector Forces
figure;
plot(time, forces(:,1), 'b-');
hold on;
plot(time, forces(:,2), 'r-');
title(strcat("End-Effector Forces vs. Time for ", model_title));
xlabel("time (s)");
```

```
ylabel("forces (N)");
legend("Fx", "Fy");

% Plot Force Errors
figure;
plot(time, force_errors(:,1), 'b-');
hold on;
plot(time, force_errors(:,2), 'r-');
title(strcat("Force Errors vs. Time for ", model_title));
xlabel("time (s)");
ylabel("force errors (N)");
legend("Fx error", "Fy error");
```

2. 2.1)



Assuming frictionless:

6DoF:
V_x, V_y, V_z
w_x, w_y, w_z

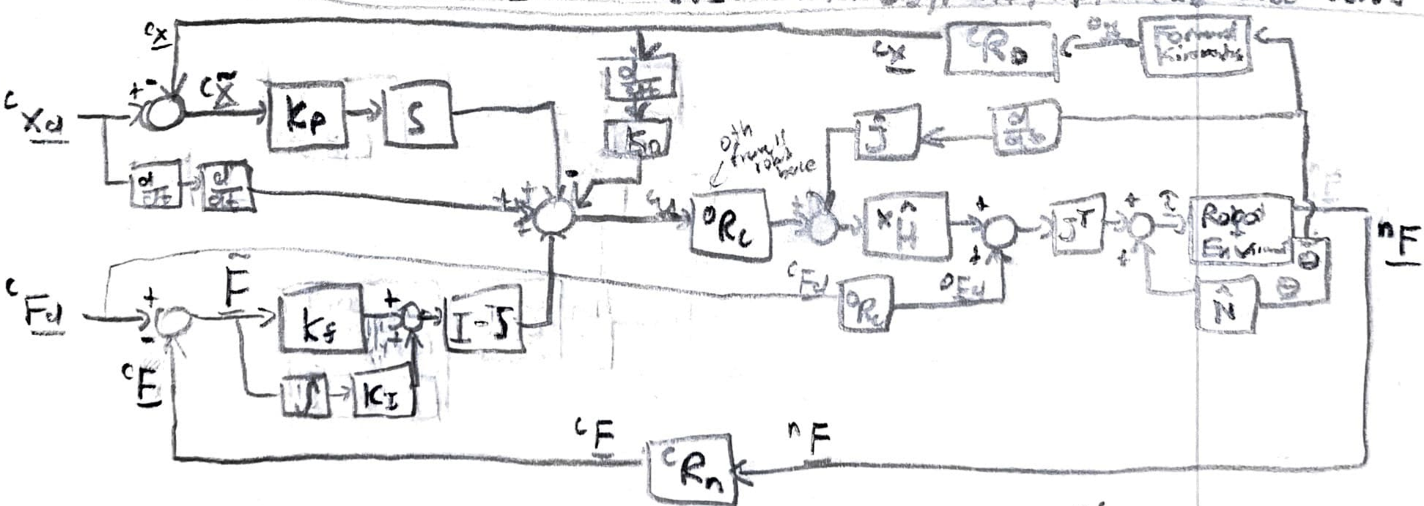
	(Velocity) Kinematic	(Force) Static
Natural Constraints (Geometry)	$V_y=0$ $V_z=0$ $w_x=0$ $w_y=0$	$f_x=0$ $\tau_z=0$
Artificial Constraints (control)	$V_x=V_{xd}=0$ $w_z=w_{zd}$	$f_y=f_{yd}=0$ $f_z=f_{zd}$ $\tau_x=\tau_{xd}=0$ $\tau_y=\tau_{yd}=0$
	Position Control	Force Control

$$\sigma = \begin{array}{c} \overbrace{\begin{array}{ccc} x & y & z \end{array}}^{\text{translation}} \; \overbrace{\begin{array}{ccc} x & y & z \end{array}}^{\text{rotation}} \\ [\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 1 \end{array}] \end{array}$$

$$\operatorname{diag}(\sigma) = \boxed{S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}$$

2.2)

Note: ${}^c\underline{X_d} = [{}^cx_d \; {}^cy_d \; {}^cz_d \; {}^c\theta_d \; {}^c\phi_d \; {}^c\psi_d]$ (Euler angles); ${}^c\underline{F_d} = [{}^cF_{xd} \; {}^cF_{yd} \; {}^cF_{zd} \; {}^c\tau_{xd} \; {}^c\tau_{yd} \; {}^c\tau_{zd}]$;
$I - S = \operatorname{diag}([0\;1\;1\;1\;1\;0])$; K_f, K_p, and K_D can be vectors



for τ's of F would have to use ${}^c\tau = {}^cR_n {}^n\tau + {}^c d_{\text{cross}} {}^cR_n {}^nF$

2. 2.2 continued

Control law: $$\underline{\tau} = \hat{N} + J^T\left({}^0\underline{F_d} + {}^x\hat{H}\left({}^0\underline{u} - \dot{J}\dot{\theta}\right)\right)$$

where:

$$ {}^0\underline{u} = {}^0R_c\left[{}^c\ddot{\underline{x}}_d + K_p S\, {}^c\underline{e} + (I-S)\left(K_s\underline{E} + K_I\int\underline{E}\right) - K_D\, {}^c\dot{\underline{x}}\right]$$

2.3

↳ The constraint frame should move with the tool assuming the tool moves with the screw, this would keep the natural and thus artificial constraints the same even as motion occurs.

↳ Assuming the wrist is the n^{th} frame of this robot the measured force and torque will be in the n^{th} frame not the constraint frame. Thus to be compared to the force in the constraint frame it will be necessary to perform a coordinate transformation.

(Rigid manipulator, so could place final frame (n) on wrist out of convenience)

As shown in 2.2 coordinate transforms would be used to go from the n^{th} frame to the constraint frame, mathematically:

$$\boxed{{}^c\underline{F} = {}^cR_n\, {}^n\underline{F}}$$

Do note the torques would use ${}^c\underline{\tau} = {}^cR_n\, {}^n\underline{\tau} + {}^c\underline{d}_{cn} \times {}^cR_n\, {}^n\underline{f}$ when ${}^c d_{cn}$ is the distance between the constraint and n^{th} frame.

$${}^n\underline{F} = \left[{}^nF_x \; {}^nF_y \; {}^nF_z \; {}^n\tau_x \; {}^n\tau_y \; {}^n\tau_z\right]$$

$${}^c\underline{F} = \left[{}^cF_x \; {}^cF_y \; {}^cF_z \; {}^c\tau_x \; {}^c\tau_y \; {}^c\tau_z\right]$$