Robot Control Problem Set 7

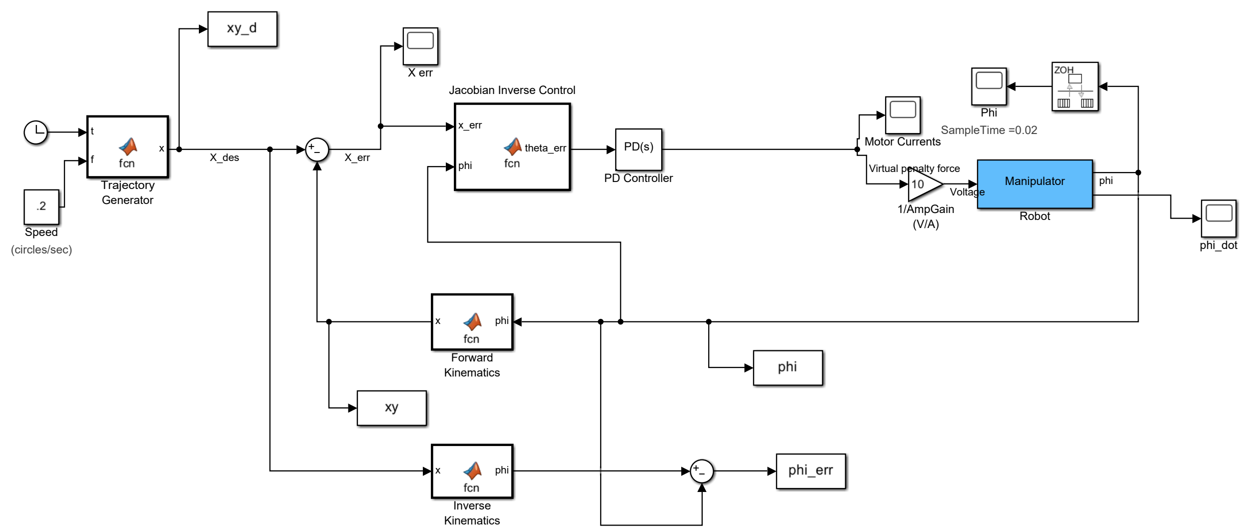
Operational Space Control

Will Graham

1 JACOBIAN INVERSE CONTROL

1. Begin with **Jacobian Inverse Control**, in which case you can use the same the same PD gains that you used when doing decentralized control in joint space (PS#5, problem 1). Simulate at low speed (f=0.2 circles/s) and high speed (f=1 circles/s), and compare the **trajectory** and **joint errors** from with Problem 1.1 of PS#5.

1.1.1 Simulink Model



1.1.2 Jacobian Inverse Control Function

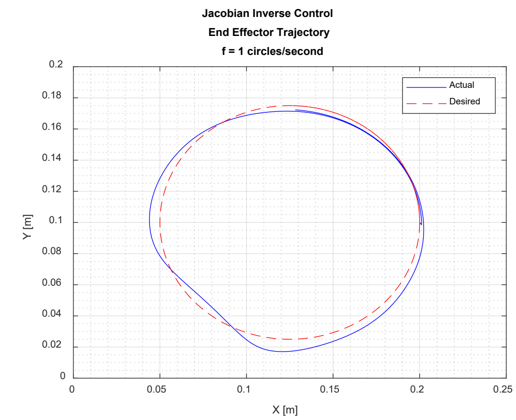
Jacobian Inverse Control

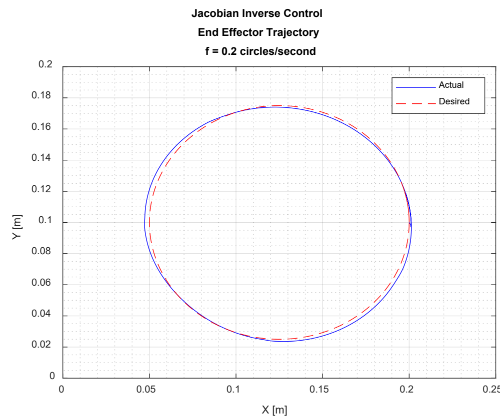
J_inv_ctrl ▸ Jacobian Inverse Control

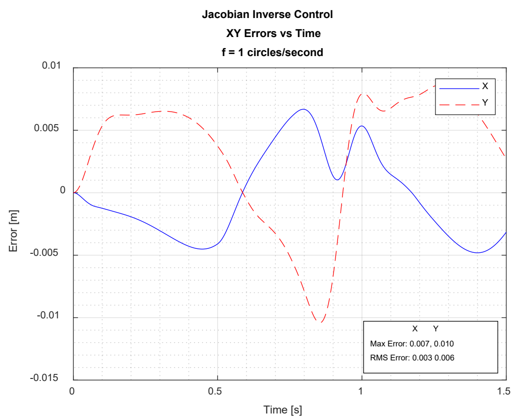
```
function theta_err = fcn(x_err,phi)
% This block supports an embeddable subset of the MATLAB language.
% See the help menu for details.

% 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));
Jt = [1 0;-1 1]; % transmission Jacobian for belt drive

theta = Jt*phi;

J =[- (3*sin(theta(1) + theta(2)))/20 - (3*sin(theta(1)))/20, -(3*sin(theta(1) + theta(2)))/20;
(3*cos(theta(1) + theta(2)))/20 + (3*cos(theta(1)))/20,  (3*cos(theta(1) + theta(2)))/20];


theta_err = inv(J)*x_err
```

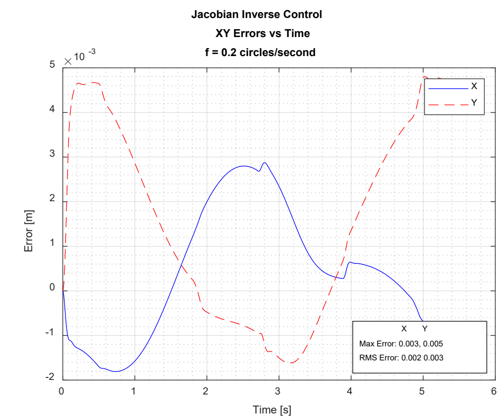
Note: The symbolic calculation of the Jacobian is found within the attached PS7 code. This was used in future Jacobian calculations.

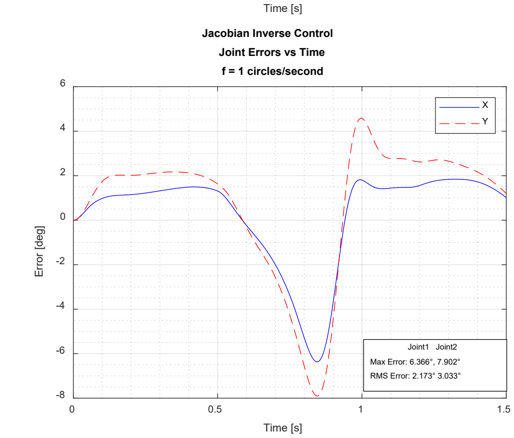
1.1.3 Jacobian Inverse Control Results

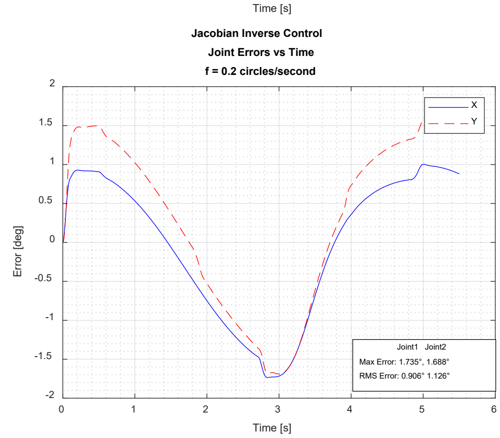




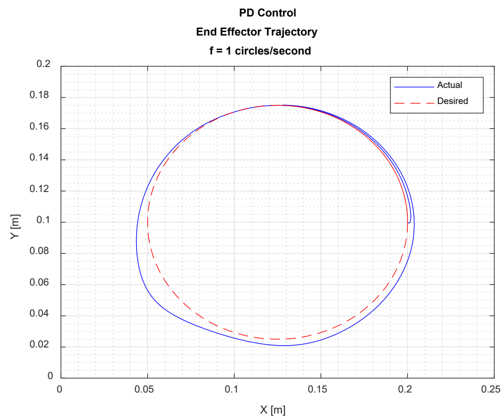


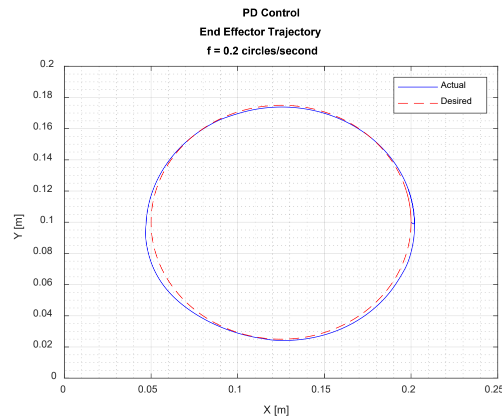


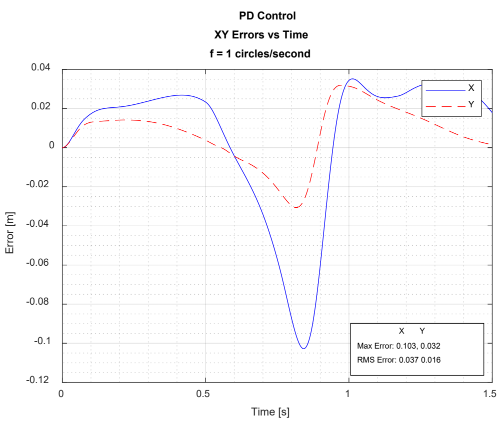


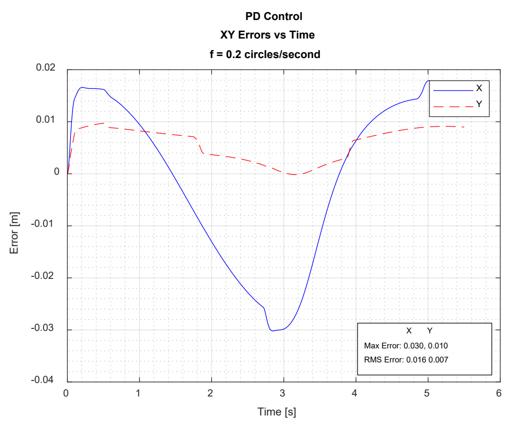


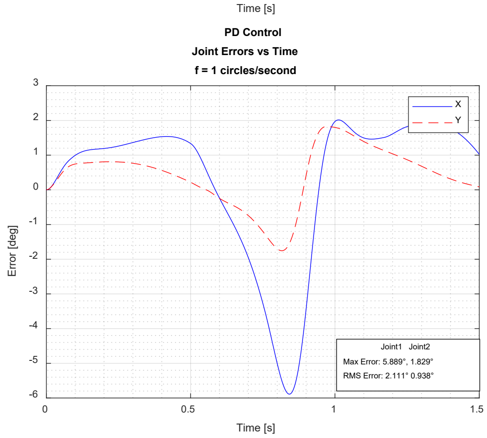
1.1.4 PD Control

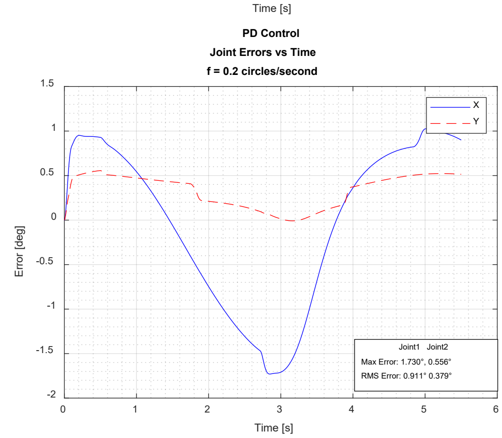












1.1.5 Comparison Table for f = 1 circles/second

	JIC Control	PD Control	Difference
X Max, Y Max [m]	0.007, 0.010	0.103, 0.032	-0.096,-0.022
X RMS, Y RMS [m]	0.003, 0.006	0.037, 0.016	-0.034,-0.01
Joint 1 Max, Joint 2 Max [deg]	6.366, 7.902	5.889, 1.829	0.477,6.073
Joint 1 RMS, Joint 2 RMS [deg]	2.173, 3.033	2.111, 0.938	0.062,2.095

1.1.6 Comparison Table for f = 0.2 circles/second

	JIC Control	**PD Control**	**Difference**
X Max, Y Max [m]	0.003, 0.005	0.030, 0.010	-0.027,-0.005
X RMS, Y RMS [m]	0.002, 0.003	0.016, 0.007	-0.014,-0.004
Joint 1 Max, Joint 2 Max [deg]	1.735, 1.688	1.730, 0.556	0.005,1.132
Joint 1 RMS, Joint 2 RMS [deg]	0.906, 1.126	0.911, 0.379	-0.005,0.747

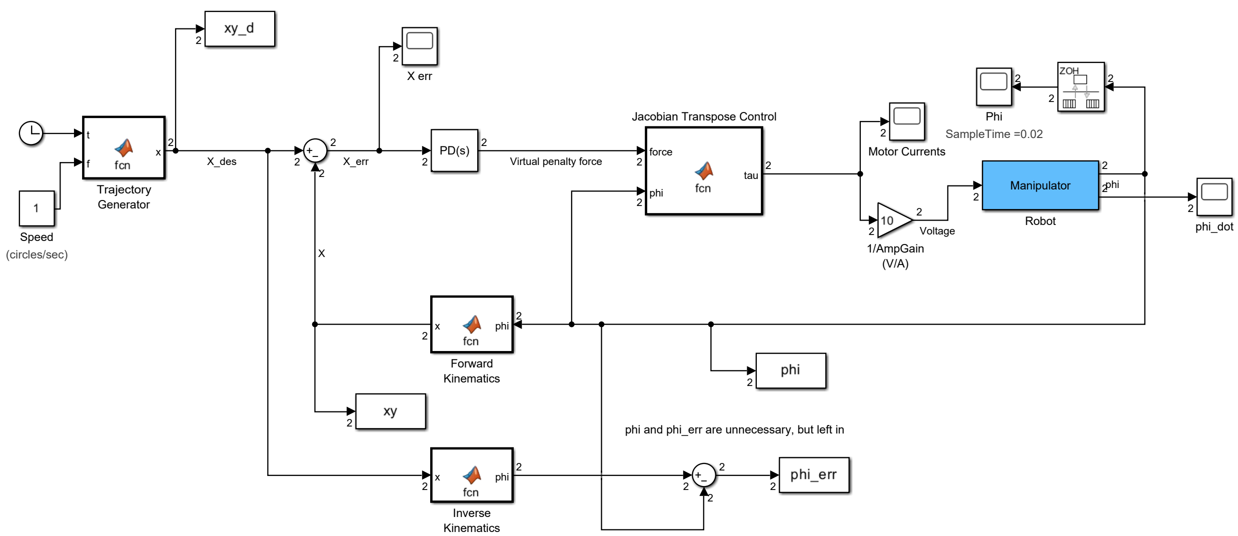
2 JACOBIAN TRANSPOSE CONTROL

2. Next, you will implement a series of operational space controllers based on the **Jacobian Transpose**. For each of the following control schemes, simulate at low speed (f=0.2 circles/s) and high speed (f=1 circles/s) and compare the tracking performance with the other control schemes. For these controllers, you will not be computing the joint errors at all, so just compare the errors in operational space. Since the PD gains for these controllers will now be penalizing errors in operational space, you will have to find a different set of PD gains than you used in problem 1. For problem 2.1, find a set of PD gains that results in comparable RMS errors to problem 1. Use the same PD gains in problems 2.1-2.3 for a fair comparison.

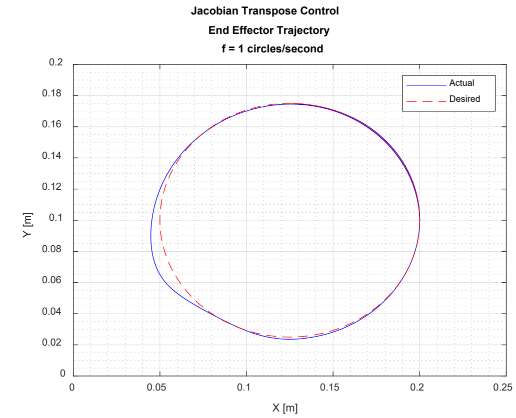
- **2.1** Jacobian Transpose Control
- **2.2** Inverse Dynamics Control in Operational Space
- **2.3** Robust Control in Operational Space

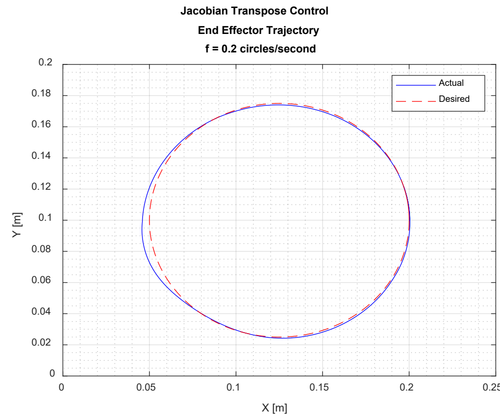
2.1 JACOBIAN TRANSPOSE CONTROL

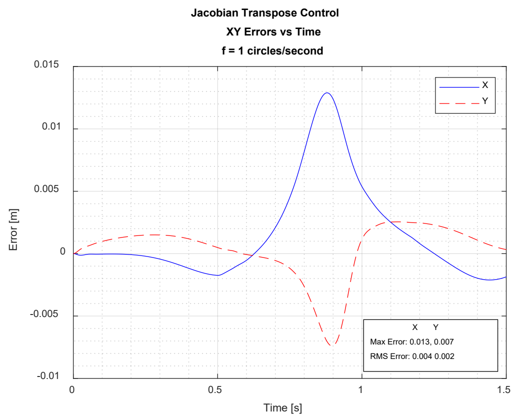
2.1.1 Simulink Model

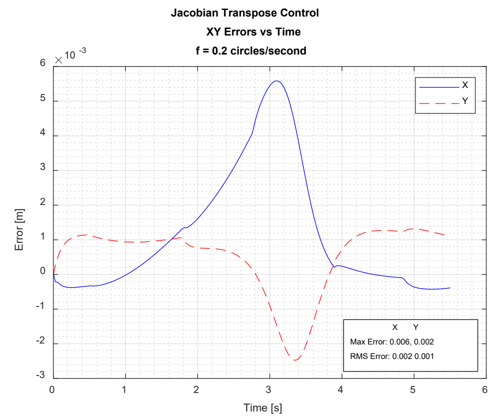


2.1.2 Plots



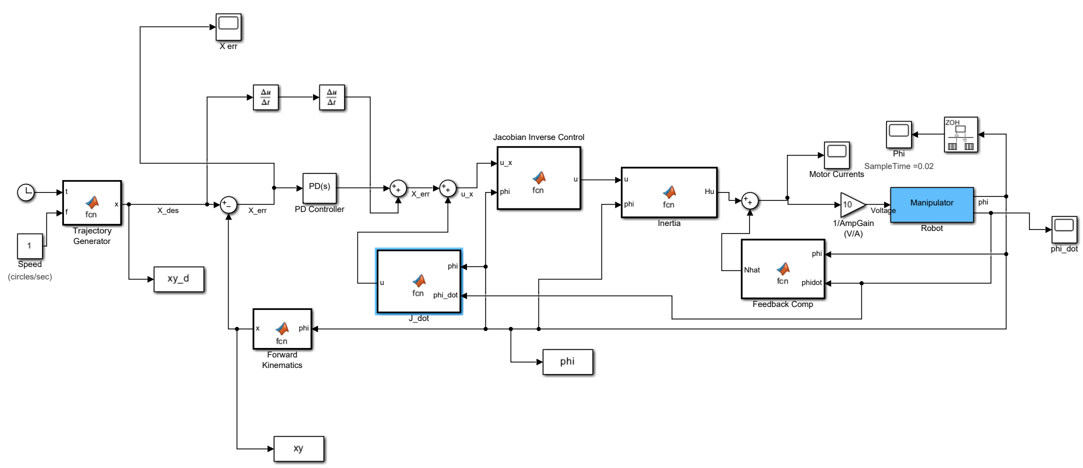




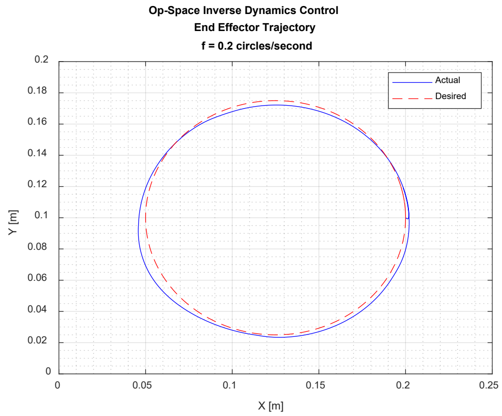


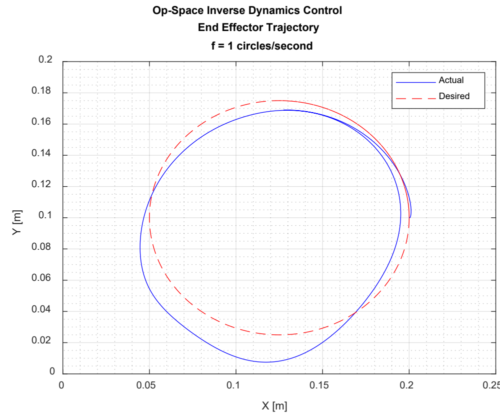
2.2 INVERSE DYNAMICS CONTROL IN OPERATIONAL SPACE

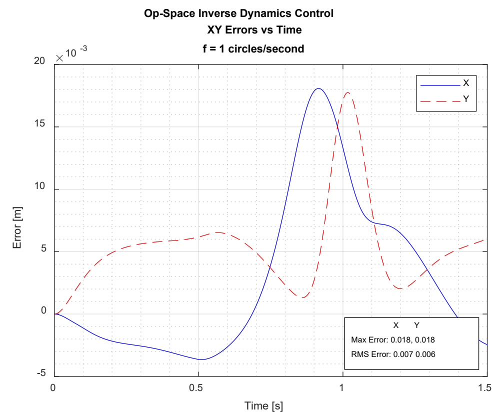
2.2.1 Simulink Model

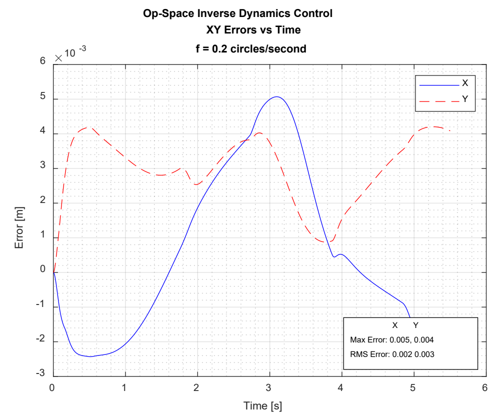


2.2.2 Plots









2.3 Robust Control in Operational Space

Couldn't get around to this. Had a concussion, and didn't have enough time to learn and implement this.