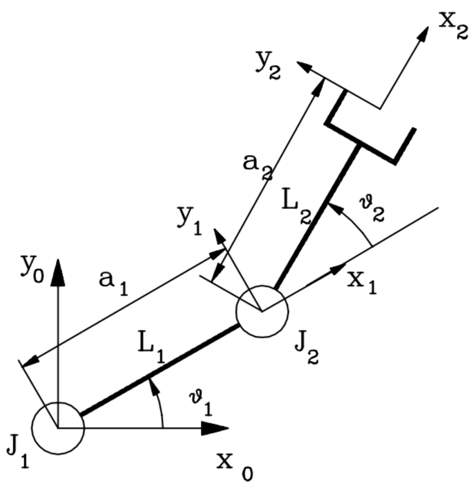
The take-home part is 20% of the total exam grade.
Please do your assignment independently.

Problem 1

(20pt) Consider the 2D planner robotic arm:



Using the provided matlab code for a 2d plannar arm manipulator.
Let $\mathbf{q} = [q_1, q_2]^\intercal$ where $q_1 = \theta_1$ and $q_2 = \theta_2$. The dynamic model of the system is given by

$$M(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \tau$$

Using the provided matlab code, which include the symbolic expressions for matrices $M(\cdot)$ and $C(\cdot)$.
The initial state is given

$$\begin{bmatrix} q_1(0) & q_2(0) & \dot{q}_1(0) & \dot{q}_2(0) \end{bmatrix} = \begin{bmatrix} -0.5 & 0.2 & 0.1 & 0.1 \end{bmatrix}$$

Design the following two controllers:

- Stabilize the system to the origin $\mathbf{q} = [0, 0]^\intercal$ and $\dot{\mathbf{q}} = [0, 0]^\intercal$ using PD control.
- Track a desired trajectory $q_1(t) = 0.2$ for all $t \geq 0$ and $q_2(t) = \sin 2t$ for all $t \geq 0$ using inverse dynamic control.

You are expected to implement: two separate ode function for simulating the system dynamics under the closed loop control and demonstrate the performance of controllers with and without initial errors.

```
% Notations: For a given variable, x, dx is its time derivative, ddx is
% 2nd-order derivative.
clc
clear all;
close all;
% the following parameters for the arm
I1=10;  I2 = 10; m1=5; r1=.5; m2=5; r2=.5; l1=1; l2=1;

% we compute the parameters in the dynamic model
```

```
a = I1+I2+m1*r1^2+ m2*(l1^2+ r2^2);
b = m2*l1*r2;
d = I2+ m2*r2^2;

%% create symbolic variable for x.
% x1 - theta1
% x2 - theta2

symx= sym('symx',[4,1]);

M = [a+2*b*cos(symx(2)), d+b*cos(symx(2));
    d+b*cos(symx(2)), d];
C = [-b*sin(symx(2))*symx(4), -b*sin(symx(2))*(symx(3)+symx(4)); b*sin(symx(2))*symx(3),0];
invM = inv(M);
invMC= inv(M)*C;

% the options for ode
% initial condition
x0= [-0.5,0.2,0.1,0.1];
w=0.2;



%% Implement the PD control for set point tracking.
xf = [0, 0, 0, 0];
options = odeset('RelTol',1e-4,'AbsTol',[1e-4, 1e-4, 1e-4, 1e-4]);
[T,X] = ode45(@(t,x) PDControl(t,x),[0 tf],x0, options);

%
% figure('Name','Theta_1 under PD SetPoint Control');
% plot(T, X(:,1),'r-');
% hold on
%
% figure('Name','Theta_2 under PD SetPoint Control');
% plot(T, X(:,2),'r--');
% hold on

%% Implement the inverse dynamic control.
options = odeset('RelTol',1e-4,'AbsTol',[1e-4, 1e-4, 1e-4, 1e-4]);
[T,X] = ode45(@(t,x) inverseDynamicControl(t,x),[0 tf],x0, options);

% figure('Name','Theta_1 under Computed Torque Control');
% plot(T, X(:,1),'r-');
% hold on
% plot(T, w*ones(size(T,1),1),'b-');
% figure('Name','Theta_2 under Computed Torque Control');
% plot(T, X(:,2),'r--');
% hold on
% plot(T, sin(2*T),'b-');
```