
Table of Contents

.....	1
OpenManipulator Parameters	1
Input: Enter Desired Position	2
Output: Desired Joint Variables	2
Move Arm Function	2
MoveArmU	3
OpenManipulator Joint Space Positions	6
OpenManipulator Stick Model Positions	7
PlanarArmODE Function	10
Inverse Kinematics of the 4DOF Robot Manipulator	13

```
%-----  
%-----  
% COURSE:    RBE502 ROBOT CONTROL  
%  
% ASSIGNMENT:TERM PROJECT  
%  
% AUTHORS:   KYLE CANTRELL (KJCANTRELL@WPI.EDU)  
%            CRAIG MILLER  (CDMILLER@WPI.EDU)  
%            JORDAN NELSON (JVNELSON@WPI.EDU)  
%  
% DATE:      04/27/2019  
%-----  
% TITLE: Open_Manipulator_Dynamic Model  
%-----  
  
clear all  
clc
```

OpenManipulator Parameters

```
l1 = 0.077; %m  
l2 = 0.130; %m  
l3 = 0.124; %m  
l4 = 0.126; %m  
  
lc1 = 0.077/2; %m (estimate)  
lc2 = 0.130/2; %m (estimate)  
lc3 = 0.124/2; %m (estimate)  
lc4 = 0.126/2; %m (estimate)  
  
m1 = 0.14; %kg (estimate)  
m2 = 0.14; %kg (estimate)  
m3 = 0.14; %kg (estimate)  
m4 = 0.14; %kg (estimate)  
  
I1 = 0.12; %kg*m^2 (estimate)
```

```
I2 = 0.12; %kg*m^2 (estimate)
I3 = 0.12; %kg*m^2 (estimate)
I4 = 0.12; %kg*m^2 (estimate)

g = 9.8; %m/s^2
```

Input: Enter Desired Position

```
% Joint angles are in degrees

% i) Take the input from the user for the desired end-effector
position

Xd = 0.119; % input('Enter desired X for end-effector(m)0.119: ');
Yd = 0.0;   % input('Enter desired Y for end-effector(m)0.0: ');
Zd = 0.1206; % ('Enter desired Z for end-effector(m)0.1206: ');
elb_pos = 1; %input('Enter elbow position 1 (0 - elbow down, 1 - elbow
up):');
```

Output: Desired Joint Variables

```
%Desired joint variables corresponding to the end-effector position
[thetali, theta2i, theta3i, theta4i] = invKin([Xd Yd Zd elb_pos], l1,
l2, l3, l4);
```

Move Arm Function

```
%End Effector is middle of gripper arm of OpenManipulator

[T4, X4, end_eff_x4, end_eff_y4, end_eff_z4 torque4] =
moveArmU([0.1411, 0, .31729 1],...
[.119, 0, 0.1206 1]);

% Comparing joint 3 torques
figure('Name','OpenManipulator:MATLAB Joint Torques Vs. Time 23');
plot(T4, torque4(1,1:size(T4,1)), 'g-');
hold on
plot(T4, torque4(2,1:size(T4,1)), 'r-');
plot(T4, torque4(3,1:size(T4,1)), 'b-');
plot(T4, torque4(4,1:size(T4,1)), 'm-');
ylim([-50 50]);
grid on
xlabel('Time in Seconds');
ylabel('Torque (Nm)');
legend('Theta 1', 'Theta 2', 'Theta 3', 'Theta 4','Location','Best');
title('OpenManipulator:MATLAB Joint Torques Vs. Time 23');
hold off

%Comparing end-effector X error
figure('Name','End-Effector X Position Error Control law PID');
```

```

plot(T4,Xd - end_eff_x4, 'r--');
xlabel('Time (seconds)');
ylabel('Position Error (meters)');
legend('PID: End Effector X Position Error');
title('End-Effector X Position Error Control law PID');

% Comparing end-effector Y error
figure('Name','OpenManipulator:End-Effector Y Position Error Control
Law PID');
plot(T4,Yd - end_eff_y4, 'r--');
xlabel('Time (seconds)');
ylabel('Position Error (meters)');
legend('U: End Effector Y Position Error');
title('OpenManipulator:End-Effector Y Position Error Control Law
PID');

% Comparing end-effector Z error
figure('Name','OpenManipulator:End-Effector Z Position Error Control
Law PID');
plot(T4, Zd - end_eff_z4, 'r--');
xlabel('Time (seconds)');
ylabel('Position error (meters)');
legend('PID: End effector Z Position Error');
title('OpenManipulator:End-Effector Z Position Error Control Law
PID');

```

MoveArmU

```

function [T, X, end_eff_x, end_eff_y, end_eff_z torque] =
    moveArmU(initial_config,...
        final_config)

l1 = 0.077; %m
l2 = 0.130; %m
l3 = 0.124; %m
l4 = 0.126; %m

lc1 = 0.077/2; %m
lc2 = 0.130/2; %m
lc3 = 0.124/2; %m
lc4 = 0.126/2; %m

m1 = 0.14; %kg
m2 = 0.14; %kg
m3 = 0.14; %kg
m4 = 0.14; %kg

I1 = 0.12; %kg*m^2
I2 = 0.12; %kg*m^2
I3 = 0.12; %kg*m^2
I4 = 0.12; %kg*m^2

% Gains

```

```

Kp = [100 0 0 0; 0 100 0 0; 0 0 100 0; 0 0 0 100];
Kv = [100 0 0 0; 0 70 0 0; 0 0 70 0; 0 0 0 70];
Ki = [100 0 0 0; 0 100 0 0; 0 0 100 0; 0 0 0 100];

g = 9.8; %m/s^2

[theta1i, theta2i, theta3i, theta4i] = invKin1(initial_config, l1, l2,
    l3, l4);

[theta1f, theta2f, theta3f, theta4f] = invKin(final_config, l1, l2,
    l3, l4);

x0 = [theta1i, theta2i, theta3i, theta4i, 0, 0, 0, 0]; % Initial
    Condition
xd = [theta1f, theta2f, theta3f, theta4f, 0, 0, 0, 0]; % Desired
    Condition

tf=5.953;

% ode15 solves the differential equation and returns X with respect to
    T.

global torque
global t_old; % Time old
t_old = 0;
global dt; % Time dt for integration
dt = 0;
global ie; % Integrated error
ie = [0;0;0;0];

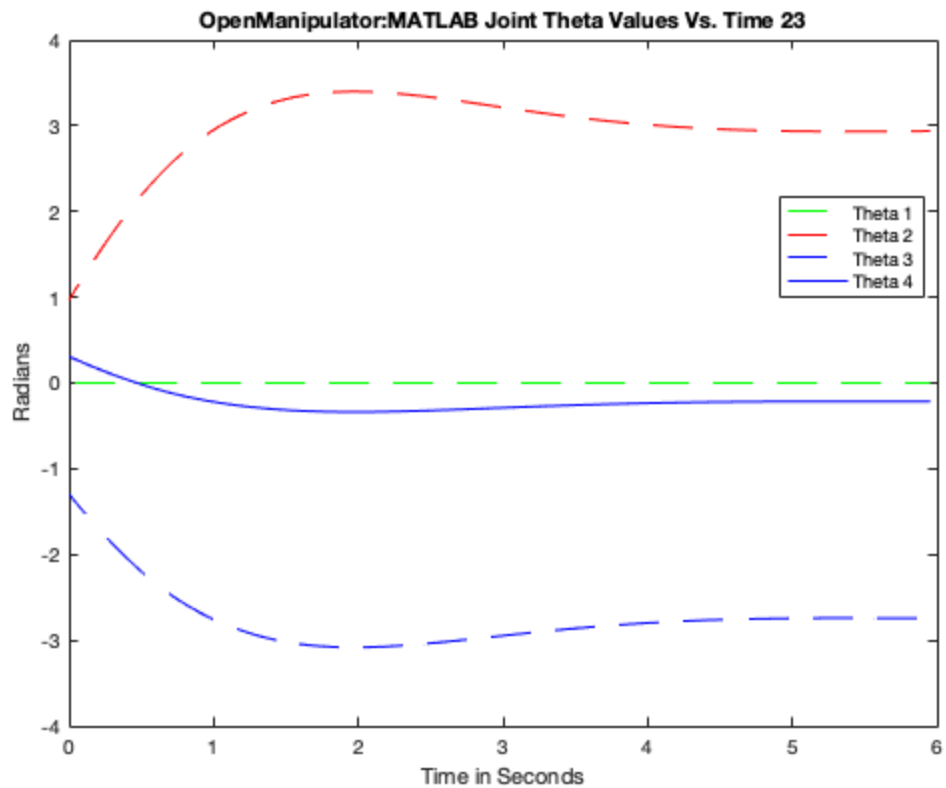
torque = [];
%opts = odeset('RelTol',1e-3);
[T,X] = ode15s(@(t,x)planarArmODE(t,x),[0 tf],x0);

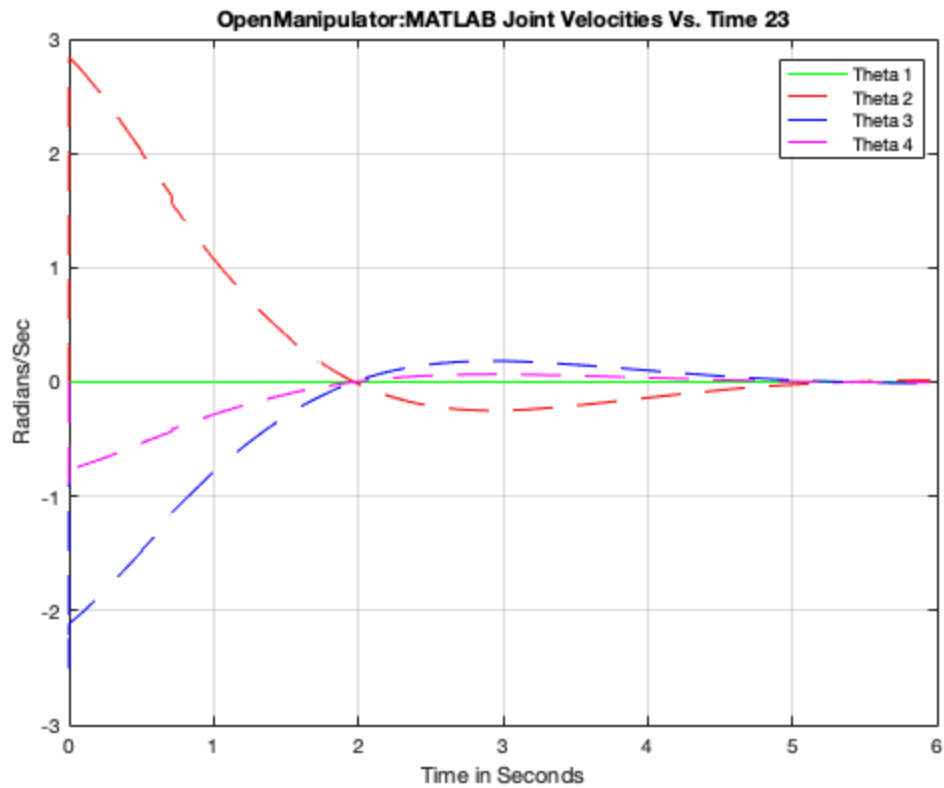
% Joint Angles vs Time
figure('Name','OpenManipulator:MATLAB Joint Theta Values Vs. Time
    23');
plot(T, X(:,1),'g--');
hold on
plot(T, X(:,2),'r--');
plot(T, X(:,3),'b--');
plot(T, X(:,4),'b-');
xlabel('Time in Seconds');
ylabel('Radians');
legend('Theta 1', 'Theta 2', 'Theta 3', 'Theta 4','Location','Best');
title('OpenManipulator:MATLAB Joint Theta Values Vs. Time 23');

% Joint Velocities vs Time
figure('Name','OpenManipulator:MATLAB Joint Velocities Vs. Time 23');
plot(T, X(:,5),'g-');

```

```
hold on
grid on
plot(T, X(:,6),'r--');
plot(T, X(:,7),'b--');
plot(T, X(:,8),'m--');
xlabel('Time in Seconds');
ylabel('Radians/Sec');
legend('Theta 1', 'Theta 2', 'Theta 3','Theta 4','Location','Best');
title('OpenManipulator:MATLAB Joint Theta Values Vs. Time 23');
```





OpenManipulator Joint Space Positions

```
%X(:,1) Theta 1
%X(:,2) Theta 2
%X(:,3) Theta 3
%X(:,4) Theta 4

%Base Position
base_om_x = 0;
base_om_y = 0;
base_om_z = 0;

%Joint 1 Position
joint_1_x = 0;
joint_1_y = 0;
joint_1_z = 11;

%Joint 2 Position
joint_2_x = 12.*cos(X(:,2)).*cos(X(:,1));
joint_2_y = 12.*cos(X(:,2)).*sin(X(:,1));
joint_2_z = 11 + 12.*sin(X(:,2));

%Joint 3 Position
joint_3_x = (12.*cos(X(:,2))+13.*cos(X(:,2)+X(:,3))).*cos(X(:,1));
joint_3_y = (12.*cos(X(:,2))+13.*cos(X(:,2)+X(:,3))).*sin(X(:,1));
```

```

joint_3_z = l1 + l2.*sin(X(:,2))+l3.*sin(X(:,2)+X(:,3));

%End Effector Position
end_eff_x =
    (l2.*cos(X(:,2))+l3.*cos(X(:,2)+X(:,3))+l4.*cos(X(:,2)+X(:,3)+X(:,4))).*cos(X(:,1))
end_eff_y =
    (l2.*cos(X(:,2))+l3.*cos(X(:,2)+X(:,3))+l4.*cos(X(:,2)+X(:,3)+X(:,4))).*sin(X(:,1))
end_eff_z = l1 +
    l2.*sin(X(:,2))+l3.*sin(X(:,2)+X(:,3))+l4.*sin(X(:,2)+X(:,3)+X(:,4));

```

OpenManipulator Stick Model Positions

```

% Initial Position
line1_x = [0 joint_1_x(1,1) joint_2_x(1,1) joint_3_x(1,1)
    end_eff_x(1,1)];
line1_y = [0 joint_1_y(1,1) joint_2_y(1,1) joint_3_y(1,1)
    end_eff_y(1,1)];
line1_z = [0 joint_1_z(1,1) joint_2_z(1,1) joint_3_z(1,1)
    end_eff_z(1,1)];

%Final Position
line3_x = [0 joint_1_x(end,1) joint_2_x(end,1) joint_3_x(end,1)
    end_eff_x(end,1)];
line3_y = [0 joint_1_y(end,1) joint_2_y(end,1) joint_3_y(end,1)
    end_eff_y(end,1)];
line3_z = [0 joint_1_z(end,1) joint_2_z(end,1) joint_3_z(end,1)
    end_eff_z(end,1)];

%End-Effector Position vs Time
figure('Name', 'OpenManipulator:MATLAB End-effector Position vs Time
    23')
plot(T, (end_eff_x-1.5), 'r--', T, (end_eff_y+3.75), T,
    end_eff_z, 'g--');
grid on;
xlabel('Time (s)');
ylabel('Position (m)');
legend('End Effector X Position', 'End Effector Y Position', 'End
    Effector Z Position')
title('OpenManipulator:MATLAB End-Effector Position vs Time 23');

% End-effector position Error vs Time
figure('Name', 'PID: End-effector Position Error vs Time')
plot(T, final_config(1,1) - end_eff_x, 'r--', T, final_config(1,2) -
    end_eff_y, T, final_config(1,3) - end_eff_z, 'g--');
xlabel('Time (seconds)');
ylabel('Position Error (meters)');
legend('End Effector X Position Error', 'End Effector Y Position
    Error', 'End Effector Z Position')
title('End-Effector Position Error vs Time');

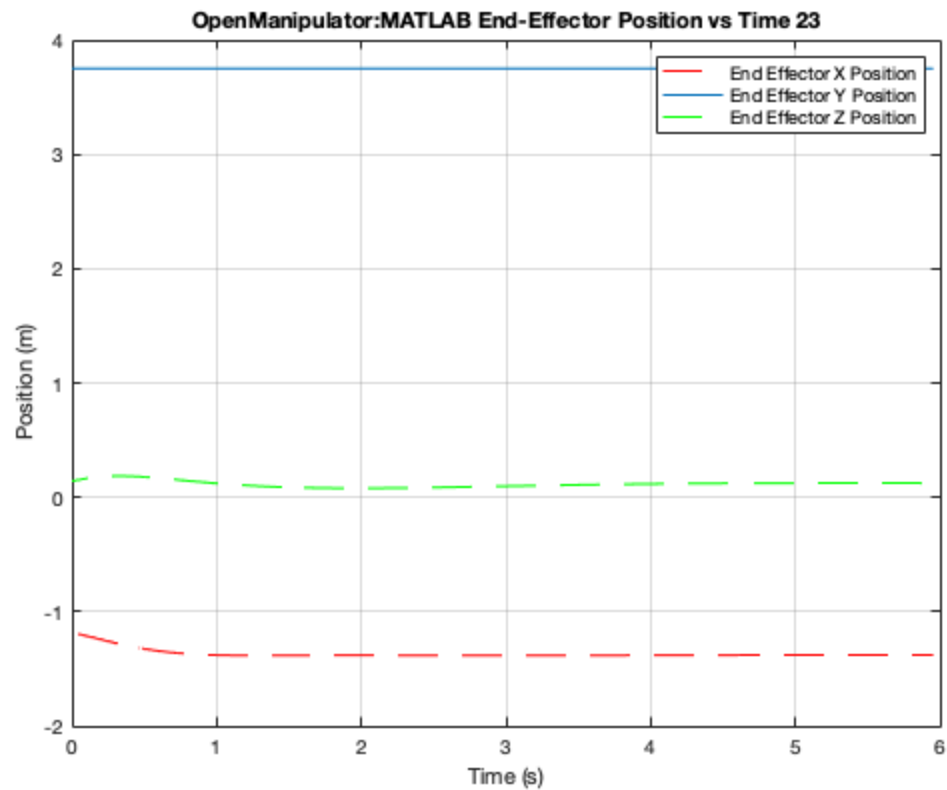
%Stick model Trajectories
figure('Name', 'OpenManipulator:Stick Model Trajectories 23')

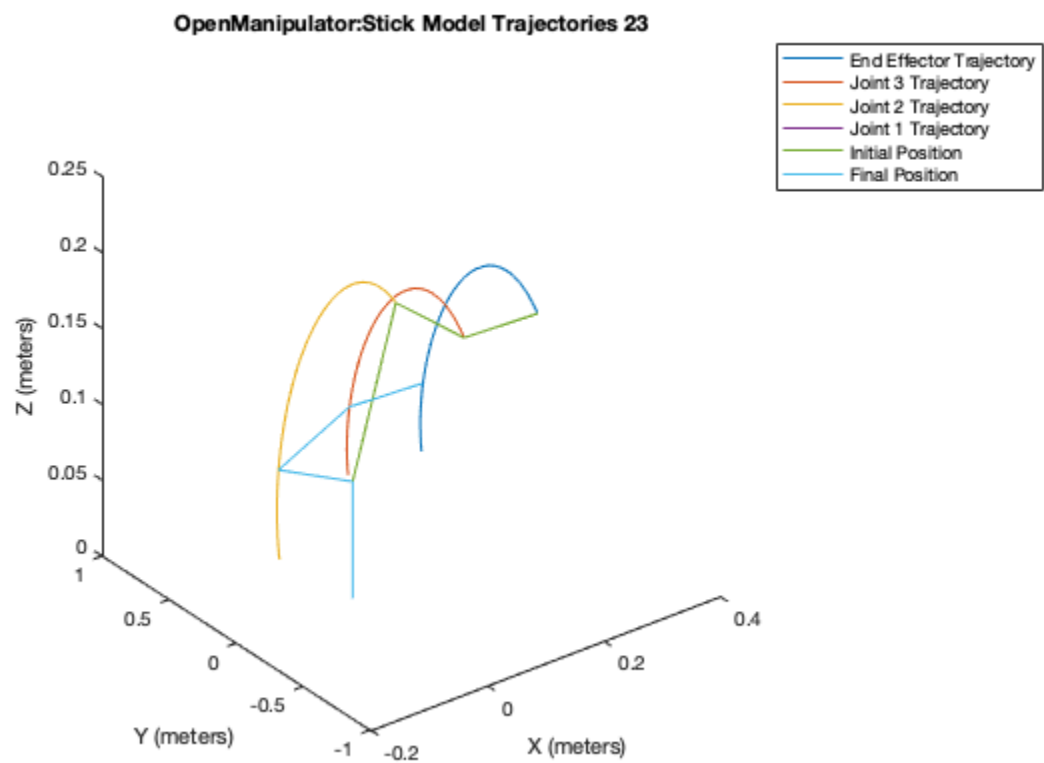
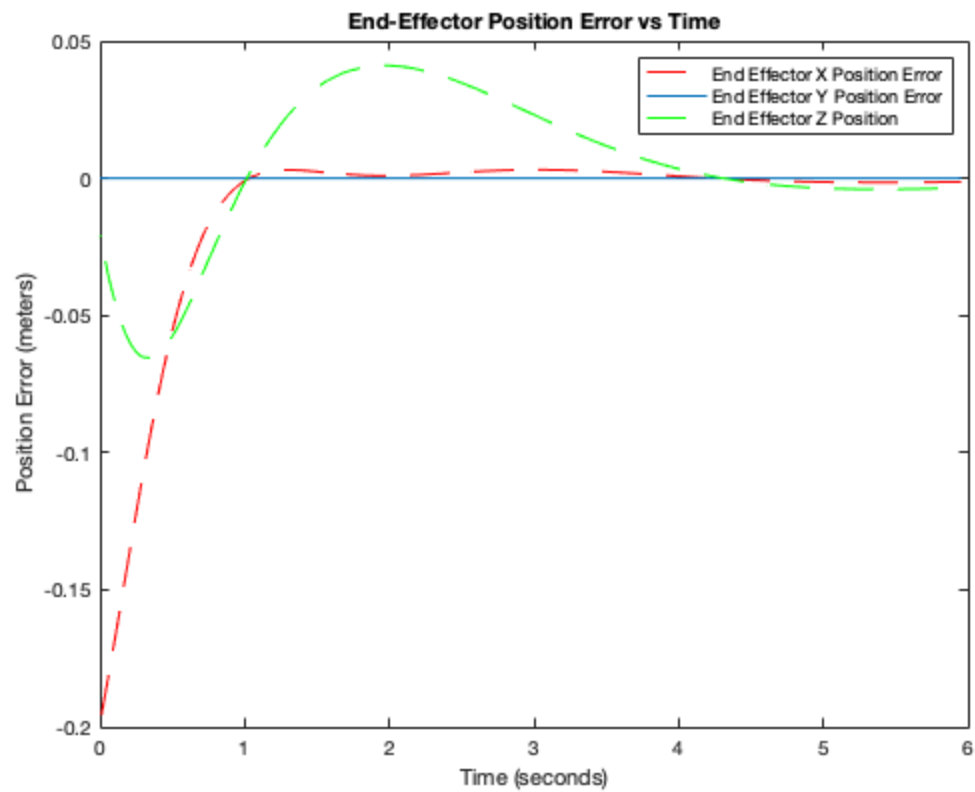
```

```

plot3(end_eff_x, end_eff_y, end_eff_z, joint_3_x, joint_3_y,
      joint_3_z, joint_2_x, joint_2_y, joint_2_z, joint_1_x, joint_1_y,
      joint_1_z, line1_x, line1_y, line1_z, line3_x, line3_y, line3_z);
xlabel('X (meters)');
ylabel('Y (meters)');
zlabel('Z (meters)');
legend('End Effector Trajectory','Joint 3 Trajectory','Joint 2
      Trajectory','Joint 1 Trajectory', 'Initial Position', 'Final
      Position');
title('OpenManipulator:Stick Model Trajectories 23');

```





PlanarArmODE Function

```
function dx = planarArmODE(t,x)

theta_d = xd(1, 1:4)'; % Desired Set-Point Position
dtheta_d = xd(1, 5:8)'; % Desired velocity (Derivative of theta_d)

ddtheta_d = [0;0;0;0];

theta = x(1:4,1);
dtheta = x(5:8,1);
global Mmat Cmat gmat

%Calculated Inverse Kinematics with Separate M File and subbed in
(TBD)
%See MCG Calculation File for Proposed Values:

M10 = (l2^2*m3)/2 + (l2^2*m4)/2 + (l3^2*m4)/2 + (lc2^2*m2)/2 +
(lc3^2*m3)/2 + (lc4^2*m4)/2 + (lc4^2*m4*cos(2*x(2) + 2*x(3) +
2*x(4)))/2 + (l2^2*m3*cos(2*x(2)))/2 + (l2^2*m4*cos(2*x(2)))/2
- (lc2^2*m2*cos(2*x(1)))/2 + (lc2^2*m2*cos(2*x(2)))/2 +
(l3^2*m4*cos(2*x(2) + 2*x(3)))/2 - (lc2^2*m2*cos(2*x(1) - 2*x(2)))/4
- (lc2^2*m2*cos(2*x(1) + 2*x(2)))/4 + (lc3^2*m3*cos(2*x(2) +
2*x(3)))/2 + 12*lc4*m4*cos(x(3) + x(4)) + 12*lc4*m4*cos(2*x(2) +
x(3) + x(4)) + 12*13*m4*cos(2*x(2) + x(3)) + 12*lc3*m3*cos(2*x(2) +
x(3)) + 13*lc4*m4*cos(2*x(2) + 2*x(3) + x(4)) + 12*13*m4*cos(x(3)) +
12*lc3*m3*cos(x(3)) + 13*lc4*m4*cos(x(4));

M11 = (lc2^2*m2*sin(2*x(1))*sin(2*x(2)))/2;

M12 = 0;

M13 = 0;

M20 = (lc2^2*m2*sin(2*x(1))*sin(2*x(2)))/2;

M21 = l2^2*m3 + l2^2*m4 + l3^2*m4 + lc2^2*m2 + lc3^2*m3 + lc4^2*m4
+ (lc2^2*m2*cos(2*x(1)))/2 - (lc2^2*m2*cos(2*x(1))*cos(2*x(2)))/2 +
2*12*13*m4*cos(x(3)) + 2*12*lc3*m3*cos(x(3)) + 2*13*lc4*m4*cos(x(4))
+ 2*12*lc4*m4*cos(x(3))*cos(x(4)) - 2*12*lc4*m4*sin(x(3))*sin(x(4));

M22 = m4*13^2 + 2*m4*cos(x(4))*13*lc4 + 12*m4*cos(x(3))*13 + m3*lc3^2
+ 12*m3*cos(x(3))*lc3 + m4*lc4^2 + 12*m4*cos(x(3) + x(4))*lc4;

M23 = lc4^2*m4 + 12*lc4*m4*cos(x(3) + x(4)) + 13*lc4*m4*cos(x(4));

M30 = 0;

M31 = m4*13^2 + 2*m4*cos(x(4))*13*lc4 + 12*m4*cos(x(3))*13 + m3*lc3^2
+ 12*m3*cos(x(3))*lc3 + m4*lc4^2 + 12*m4*cos(x(3) + x(4))*lc4;

M32 = m4*13^2 + 2*m4*cos(x(4))*13*lc4 + m3*lc3^2 + m4*lc4^2;
```

```

M33 = m4*lc4^2 + l3*m4*cos(x(4))*lc4;

M40 = 0;

M41 = lc4^2*m4 + l2*lc4*m4*cos(x(3) + x(4)) + l3*lc4*m4*cos(x(4));

M42 = m4*lc4^2 + l3*m4*cos(x(4))*lc4;

M43 = lc4^2*m4;

C1 = (lc2^2*m2*sin(2*x(1) - 2*x(2))*x(6)^2)/4 + (lc2^2*m2*sin(2*x(1)
+ 2*x(2))*x(6)^2)/4 + (lc2^2*m2*sin(2*x(1))*x(6)^2)/2 -
l3^2*m4*sin(2*x(2) + 2*x(3))*x(5)*x(6) - l3^2*m4*sin(2*x(2) +
2*x(3))*x(5)*x(7) - lc2^2*m2*sin(2*x(1) - 2*x(2))*x(5)*x(6) +
lc2^2*m2*sin(2*x(1) + 2*x(2))*x(5)*x(6) - lc3^2*m3*sin(2*x(2)
+ 2*x(3))*x(5)*x(6) - lc3^2*m3*sin(2*x(2) + 2*x(3))*x(5)*x(7)
- lc4^2*m4*sin(2*x(2) + 2*x(3) + 2*x(4))*x(5)*x(6)
- lc4^2*m4*sin(2*x(2) + 2*x(3) + 2*x(4))*x(5)*x(7) -
l2^2*m3*sin(2*x(2))*x(5)*x(6) - l2^2*m4*sin(2*x(2))*x(5)*x(6) -
lc2^2*m2*sin(2*x(2))*x(5)*x(6) - 2*l2*lc4*m4*sin(2*x(2) + x(3) +
x(4))*x(5)*x(6) - 2*l2*lc4*m4*sin(2*x(2) + x(3) + x(4))*x(5)*x(7)
- 2*l2*l3*m4*sin(2*x(2) + x(3))*x(5)*x(6) - 2*l3*m4*sin(2*x(2)
+ x(3))*x(5)*x(7) - 2*l2*lc3*m3*sin(2*x(2) + x(3))*x(5)*x(6) -
2*l2*lc3*m3*sin(2*x(2) + x(3))*x(5)*x(7) - 2*l3*lc4*m4*sin(2*x(2)
+ 2*x(3) + x(4))*x(5)*x(6) - 2*l3*lc4*m4*sin(2*x(2) +
2*x(3) + x(4))*x(5)*x(7) - 2*l3*m4*sin(x(3))*x(5)*x(7) -
2*l2*lc3*m3*sin(x(3))*x(5)*x(7) - 2*lc4*m4*sin(x(3) + x(4))*x(5)*x(7);

C2 = (lc2^2*m2*sin(2*x(1) + 2*x(2))*x(6)^2)/2 - (lc2^2*m2*sin(2*x(1)
- 2*x(2))*x(6)^2)/2 - l2*l3*m4*sin(x(3))*x(7)^2 -
l2*lc3*m3*sin(x(3))*x(7)^2 - l2*lc4*m4*sin(x(3) + x(4))*x(7)^2
+ lc2^2*m2*sin(2*x(1) - 2*x(2))*x(5)*x(6) + lc2^2*m2*sin(2*x(1)
+ 2*x(2))*x(5)*x(6) - lc2^2*m2*sin(2*x(1))*x(5)*x(6) -
2*l2*l3*m4*sin(x(3))*x(6)*x(7) - 2*l2*lc3*m3*sin(x(3))*x(6)*x(7) -
2*l3*lc4*m4*sin(x(4))*x(6)*x(8) - 2*l3*lc4*m4*sin(x(4))*x(7)*x(8)
- 2*l2*lc4*m4*sin(x(3) + x(4))*x(6)*x(7) - 2*l2*lc4*m4*sin(x(3) +
x(4))*x(6)*x(8) - 2*l2*lc4*m4*sin(x(3) + x(4))*x(7)*x(8);

C3 = l2*l3*m4*sin(x(3))*x(6)^2 + l2*lc3*m3*sin(x(3))*x(6)^2 +
l2*lc4*m4*sin(x(3) + x(4))*x(6)^2 - l2*l3*m4*sin(x(3))*x(6)*x(7)
- l2*lc3*m3*sin(x(3))*x(6)*x(7) - 2*l3*lc4*m4*sin(x(4))*x(6)*x(8)
- 2*l3*lc4*m4*sin(x(4))*x(7)*x(8) - l2*lc4*m4*sin(x(3) +
x(4))*x(6)*x(7);

C4 = l3*lc4*m4*sin(x(4))*x(6)^2 + l3*lc4*m4*sin(x(4))*x(7)^2 +
l2*lc4*m4*sin(x(3) + x(4))*x(6)^2 + 2*l3*lc4*m4*sin(x(4))*x(6)*x(7)
- l3*lc4*m4*sin(x(4))*x(6)*x(8) - l3*lc4*m4*sin(x(4))*x(7)*x(8) -
l2*lc4*m4*sin(x(3) + x(4))*x(6)*x(8);

G1 = (lc2^2*m2*sin(2*x(1) - 2*x(2))*x(5)^2)/2 + (lc2^2*m2*sin(2*x(1) +
2*x(2))*x(5)^2)/2 + lc2^2*m2*sin(2*x(1))*x(5)^2 - lc4^2*m4*sin(2*x(2)
+ 2*x(3) + 2*x(4))*x(5)*x(8) - l2*lc4*m4*sin(2*x(2) + x(3) +
x(4))*x(5)*x(8) - l3*lc4*m4*sin(2*x(2) + 2*x(3) + x(4))*x(5)*x(8) -
l3*lc4*m4*sin(x(4))*x(5)*x(8) - l2*lc4*m4*sin(x(3) + x(4))*x(5)*x(8);

```

```

G2 = (l3^2*m4*sin(2*x(2) + 2*x(3))*x(5)^2)/2 - (lc2^2*m2*sin(2*x(1)
- 2*x(2))*x(5)^2)/4 + (lc2^2*m2*sin(2*x(1) + 2*x(2))*x(5)^2)/4
+ (lc3^2*m3*sin(2*x(2) + 2*x(3))*x(5)^2)/2 + g*lc4*m4*cos(x(2)
+ x(3) + x(4)) + g*l3*m4*cos(x(2) + x(3)) + g*lc3*m3*cos(x(2)
+ x(3)) + (lc4^2*m4*sin(2*x(2) + 2*x(3) + 2*x(4))*x(5)^2)/2 +
(l2^2*m3*sin(2*x(2))*x(5)^2)/2 + (l2^2*m4*sin(2*x(2))*x(5)^2)/2
+ (lc2^2*m2*sin(2*x(2))*x(5)^2)/2 + g*l2*m3*cos(x(2)) +
g*l2*m4*cos(x(2)) + g*lc2*m2*cos(x(2)) + l3*lc4*m4*sin(2*x(2)
+ 2*x(3) + x(4))*x(5)^2 - l3*lc4*m4*sin(x(4))*x(8)^2 -
l2*lc4*m4*sin(x(3) + x(4))*x(8)^2 + l2*lc4*m4*sin(2*x(2) +
x(3) + x(4))*x(5)^2 + l2*l3*m4*sin(2*x(2) + x(3))*x(5)^2 +
l2*lc3*m3*sin(2*x(2) + x(3))*x(5)^2;

G3 = (l3^2*m4*sin(2*x(2) + 2*x(3))*x(5)^2)/2 + (lc3^2*m3*sin(2*x(2)
+ 2*x(3))*x(5)^2)/2 + g*lc4*m4*cos(x(2) + x(3) + x(4))
+ g*l3*m4*cos(x(2) + x(3)) + g*lc3*m3*cos(x(2) + x(3))
+ (lc4^2*m4*sin(2*x(2) + 2*x(3) + 2*x(4))*x(5)^2)/2
+ l3*lc4*m4*sin(2*x(2) + 2*x(3) + x(4))*x(5)^2 +
(l2*l3*m4*sin(x(3))*x(5)^2)/2 + (l2*lc3*m3*sin(x(3))*x(5)^2)/2 -
l3*lc4*m4*sin(x(4))*x(8)^2 + (l2*lc4*m4*sin(x(3) + x(4))*x(5)^2)/2 +
(l2*lc4*m4*sin(2*x(2) + x(3) + x(4))*x(5)^2)/2 + (l2*l3*m4*sin(2*x(2)
+ x(3))*x(5)^2)/2 + (l2*lc3*m3*sin(2*x(2) + x(3))*x(5)^2)/2;

G4 = g*lc4*m4*cos(x(2) + x(3) + x(4)) + (lc4^2*m4*sin(2*x(2)
+ 2*x(3) + 2*x(4))*x(5)^2)/2 + (l3*lc4*m4*sin(2*x(2) +
2*x(3) + x(4))*x(5)^2)/2 + (l3*lc4*m4*sin(x(4))*x(5)^2)/2 +
(l2*lc4*m4*sin(x(3) + x(4))*x(5)^2)/2 + (l2*lc4*m4*sin(2*x(2) + x(3)
+ x(4))*x(5)^2)/2;

Mmat = [M10 M11 M12 M13; M20 M21 M22 M23; M30 M31 M32 M33; M40 M41 M42
M43];

Cmat = [C1; C2; C3; C4];

gmat = [G1; G2; G3; G4];

invM = inv(Mmat);

invMC = invM*Cmat;

invMg = invM*gmat;

tau = PIDControl(theta_d,dtheta_d,ddtheta_d,theta,dtheta,t);

torque =[torque, tau];

dx=zeros(8,1);

dx(1) = x(5); %dtheta1
dx(2) = x(6); %dtheta2
dx(3) = x(7); %dtheta3
dx(4) = x(8); %dtheta4

```

```

dx(5:8) = -invMC.* x(5:8) +invM*tau - invMg;

end

function tau =
PIDControl(theta_d,dtheta_d,ddtheta_d,theta,dtheta,t_now)
dt = t_now - t_old; % Find the time dt for integration
t_old = t_now;
e = theta_d - theta; % position error
de = dtheta_d - dtheta; % velocity error
ie = ie + (e .* dt); % integrated error
tau = Kp*e + Kv*de + Ki*ie;
end

disp('Finish.');
```

Finish.

```

end
```

Inverse Kinematics of the 4DOF Robot Manipulator

```

function [theta1, theta2, theta3, theta4] =
invKin(arm_pose,l1,l2,l3,l4)
x = arm_pose(1,1); % Get X coordinate of end-effector
y = arm_pose(1,2); % Get Y coordinate of end-effector
z = arm_pose(1,3); % Get Z coordinate of end-effector

elbow_up = arm_pose(1,4); % Get elbow configuration

theta1 = real(atan(y./x));

x_3 = x - l4*cos(theta1);
y_3 = y - l4*sin(theta1);
z_3 = z;

x_1 = 0;
y_1 = 0;
z_1 = l1;

D = ((x_3 - x_1)^2 + (z_3 - z_1)^2 - l2^2 - l3^2)/(2*l2*l3);

theta3 = real(atan(sqrt(1-D^2)/D))+2.4187+0.6926;

if (elbow_up == 1)

theta3 = -theta3;

end

theta2 = atan((z_3-z_1)/(x_3 - x_1)) - atan((l3*sin(theta3))/(l2+l3*cos(theta3)))+3.1174;
```

```

    theta4 = -(theta3+theta2)-.02299;

end

function [theta1, theta2, theta3, theta4] =
    invKin1(arm_pose,l1,l2,l3,l4)
    x = arm_pose(1,1); % Get X coordinate of end-effector
    y = arm_pose(1,2); % Get Y coordinate of end-effector
    z = arm_pose(1,3); % Get Z coordinate of end-effector

    elbow_up = arm_pose(1,4); % Get elbow configuration

    theta1 = real(atan(y./x));

    x_3 = x - l4*cos(theta1);
    y_3 = y - l4*sin(theta1);
    z_3 = z;

    x_1 = 0;
    y_1 = 0;
    z_1 = l1;

    D = ((x_3 - x_1)^2 + (z_3 - z_1)^2 - l2^2 - l3^2)/(2*l2*l3);

    %theta3 = real(atan(sqrt(1-D^2)/D));
    theta3 = atan(sqrt(1-D^2)/D)+0.6463;

    if (elbow_up == 1)

        theta3 = -theta3;

    end

    theta2 = atan((z_3-z_1)/(x_3 - x_1)) - atan((l3*sin(theta3))/(
    l2+l3*cos(theta3)))-1.1726;

    theta4 = -(theta3+theta2)-.0199;

end

```

Published with MATLAB® R2018b