

## Contents

---

- [Parameters of Manipulator](#)
- [Implement the inverse dynamic controller for tracking a circle as trajectory in task space](#)
- [Plotting the figures](#)

## Parameters of Manipulator

---

```
I1=10; I2 = 10; m1=5; r1=.5; m2=5; r2=.5; l1=1; l2=1;
g=9.8;
% 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];
G = [m1*g*r1*cos(symx(1))+m2*g*(l1*cos(symx(1))+r2*cos(symx(1)+symx(2)));
      m2*g*r2*cos(symx(1)+symx(2))];

invM = inv(M);
invMC= inv(M)*C;

% the options for ode
% initial condition % feel free to change
x0= [0.5;0.2;0.1;0.1];
tf = 100;
n = 100;
```

## Implement the inverse dynamic controller for tracking a circle as trajectory in task space

---

```
options = odeset('RelTol',1e-4,'AbsTol',[1e-4, 1e-4, 1e-4, 1e-4]);
[T,X] = ode45(@(t,x) odeInverseDyn(t,x,[I1,I2,m1,r1,m2,r2,l1,l2]),[0 tf],x0, options);
```

## Plotting the figures

---

Calculate X and Y from joint angles

```
x(:,1)=l1*cos(X(:,1))+l2*cos(X(:,2)+X(:,1));
y(:,1)=l1*sin(X(:,1))+l2*sin(X(:,2)+X(:,1));
figure()
plot(x(:,1),y(:,1),'r--')
```

```
title('Cartesian Space position of the endeffector')  
xlabel('X')  
ylabel('Y')
```

---

