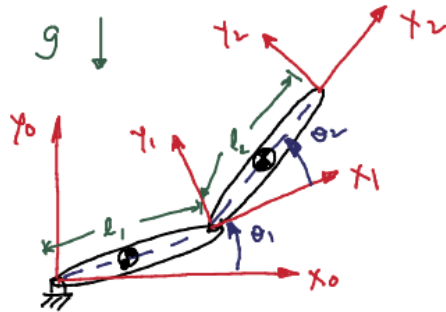


```
clc
close all
clear all
```

Simulation of a two-link manipulator



Find the equation of motion of the two link manipulator using Euler-Lagrange Method.

{Center of mass is mid-way of each link}

SOLUTION

Link i	a_i	α_i	d_i	θ_i
1	l_1	0	0	θ_1
2	l_2	0	0	θ_2

```
%%%%%%%%% INITIALIZE PARAMETERS %%%%%%%%%
```

```
%DH for link 1 except theta1
```

```
parms.a1 = 1;
parms.alpha1 = 0;
parms.d1=0;
```

```
%DH for link 2 except theta2
```

```
parms.a2 = 1;
parms.alpha2 = 0;
parms.d2=0;
```

```
%mass, inertia and gravity.
```

```
parms.m1 = 1;
parms.I1 = 0.5;
parms.m2 = 1;
parms.I2 = 0.5;
parms.g = 10;
```

```
%stuff for animation
```

```
parms.time_delay = 0.1; %delay between frames, will need some fine tuning for diff
parms.framespersec = 30;
```

```
%step size for integration. Accuracy increases as h decreases
```

```
h = 0.001;
```

```
%set the time
```

```
t0 = 0;
```

```
tN = 5;
N = (tN-t0)/h;
t = linspace(t0,tN,N);

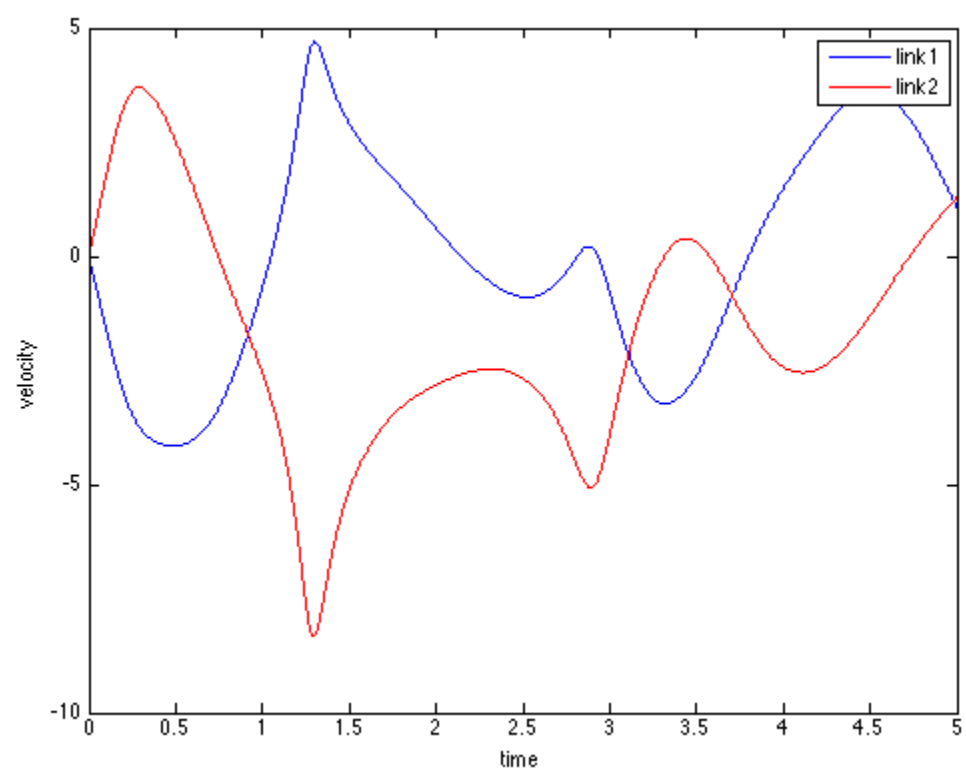
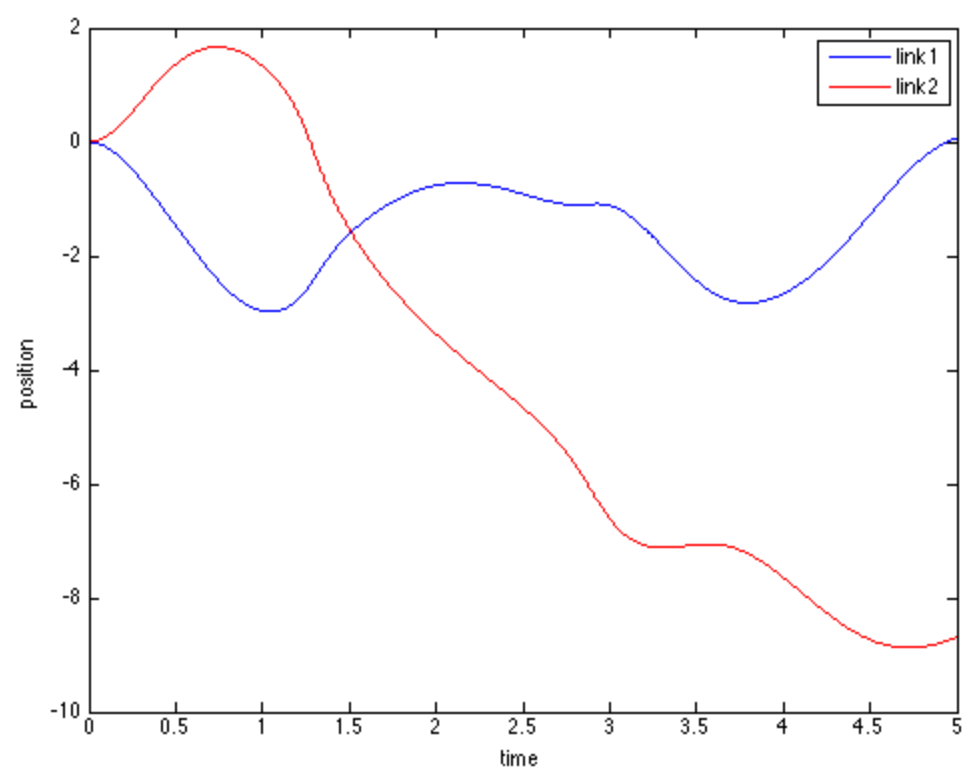
%initial conditions
theta1 = 0;
thetaldot = 0;
theta2 = 0;
theta2dot = 0;
x0=[theta1 thetaldot theta2 theta2dot]';

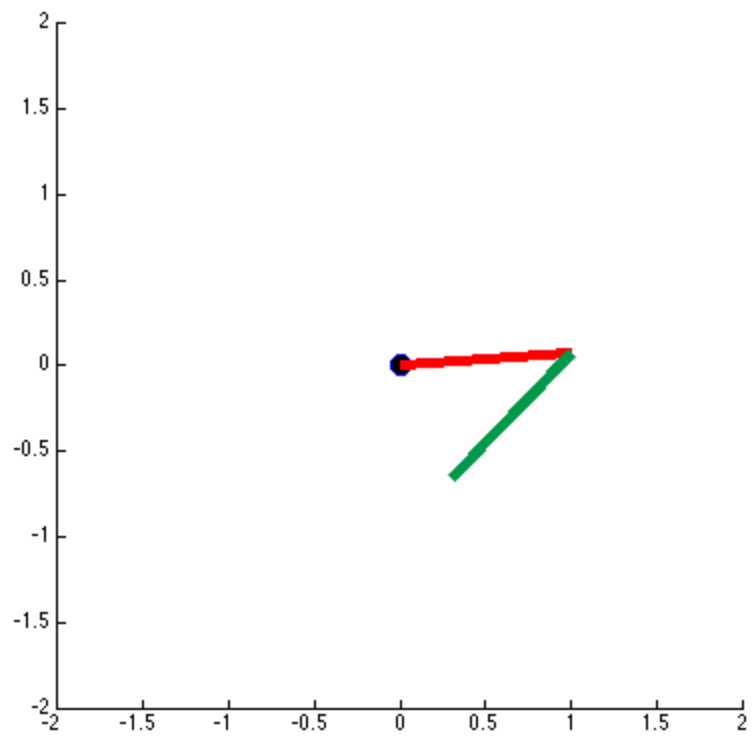
%integrate equations of motion
x = ode4('twolink_rhs',t,x0,parms);

%do some plots
figure(1)
plot(t,x(:,1),'b',t,x(:,3),'r');
xlabel('time');
ylabel('position');
legend('link1','link2');

figure(2)
plot(t,x(:,2),'b',t,x(:,4),'r');
xlabel('time');
ylabel('velocity');
legend('link1','link2');

figure(3) %animation
twolink_animation(t,[x(:,1), x(:,3)],parms);
```





Published with MATLAB® R2013b