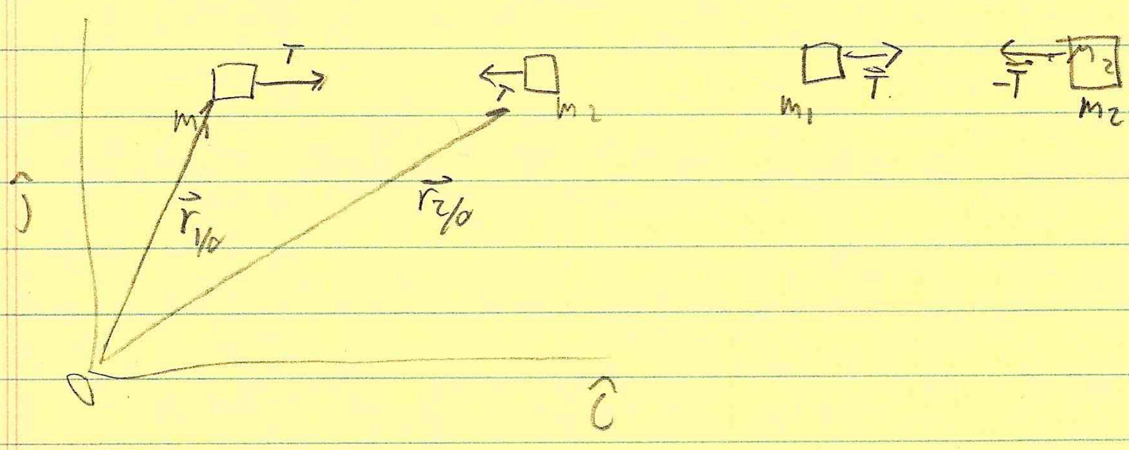
3 hr 14. 2D Dumbell

Two equal masses $m=1$ are constrained by a rod to be a distance $l=1$ apart. At $t=0$, they have equal and opposite velocity $\perp$ to the rod.

Use a set of 3 DAEs w/ numerical integration to find the subsequent motion. Use plots and animation to help debug, and quantify as many errors as possible.

First, a FBD



For m_1,

$$m_1 \vec{a}_1 = \frac{T(\vec{r}_{2/0} - \vec{r}_{1/0})}{|\vec{r}_{2/0} - \vec{r}_{1/0}|} \quad (A)$$ ✓

For m_2

✓

$$m_2 \vec{a}_2 = \frac{T(\vec{r}_{1/0} - \vec{r}_{2/0})}{|\vec{r}_{2/0} - \vec{r}_{1/0}|} \quad (B)$$

For the constraint equation,

$$(x_2-x_1)^2+(y_2-y_1)^2=\ell^2$$

we must differentiate this to be able to use it in our system of DES.

One differentiation,

$$\Rightarrow 2(x_2-x_1)(\dot{x}_2-\dot{x}_1)+2(y_2-y_1)(\dot{y}_2-\dot{y}_1)=0$$

$$\Rightarrow 2x_2\dot{x}_2-2x_2\dot{x}_1-2x_1\dot{x}_2+2x_1\dot{x}_1$$
$$+2y_2\dot{y}_2-2y_2\dot{y}_1-2y_1\dot{y}_2+2y_1\dot{y}_1=0$$

Differentiating again,

$$2\dot{x}_2\dot{x}_2+2x_2\ddot{x}_2-2\dot{x}_2\dot{x}_1-2x_2\ddot{x}_1-2\dot{x}_1\dot{x}_2-2x_1\ddot{x}_2$$
$$+2\dot{x}_1\dot{x}_1+2x_1\ddot{x}_1$$

$$+2\dot{y}_2\dot{y}_2+2y_2\ddot{y}_2-2\dot{y}_2\dot{y}_1-2y_2\ddot{y}_1-2\dot{y}_1\dot{y}_2-2y_1\ddot{y}_2$$
$$+2\dot{y}_1\dot{y}_1+2y_1\ddot{y}_1=0$$

$$\Rightarrow \dot{x}_2^2+\dot{x}_1^2-2\dot{x}_2\dot{x}_1+\ddot{x}_2(x_2-x_1)+\ddot{x}_1(x_1-x_2)$$
$$+\dot{y}_2^2+\dot{y}_1^2-2\dot{y}_2\dot{y}_1+\ddot{y}_2(y_2-y_1)+\ddot{y}_1(y_1-y_2)=0$$

(1)

So, Equation (1) will be one of the DAEs we use to solve our system. We can get four more.

Dotting (A) with $\hat{i}$,

$$\Rightarrow m_1 \ddot{x}_1 = T \frac{(x_2 - x_1)}{l_0} \quad (2)$$

We get our 2nd equation.

Dotting with $\hat{j}$,

$$\Rightarrow m_1 \ddot{y}_1 = \frac{T(y_2 - y_1)}{l_0} \quad (3)$$

Dotting (B) with $\hat{i}$ and $\hat{j}$, respectively, we get

$$m_2 \ddot{x}_2 = \frac{T(x_1 - x_2)}{l_0} \quad (4) \quad ✓$$

$$m_2 \ddot{y}_2 = \frac{T(y_1 - y_2)}{l_0} \quad (5) \quad ✓$$

we can now put all of these equations in a huge matrix:

NICE

$$\begin{bmatrix} m_1 & 0 & 0 & 0 & \frac{-(x_2-x_1)}{l_0} \\ 0 & m_1 & 0 & 0 & \frac{-(y_2-y_1)}{l_0} \\ 0 & 0 & m_2 & 0 & \frac{-(x_1-x_2)}{l_0} \\ 0 & 0 & 0 & m_2 & \frac{-(y_1-y_2)}{l_0} \\ -(x_2-x_1) & -(y_2-y_1) & -(x_1-x_2) & -(y_1-y_2) & 0 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -(\dot{x}_2-\dot{x}_1)^2 - (\dot{y}_2-\dot{y}_1)^2 \end{bmatrix}$$

And then solve this using ODE23, using the attached code.

One error we can quantify is the error in the constraint distance, plotted as $l_0 - \text{sqrt}(\text{abs}(x_1-x_2)^2 + \text{abs}(y_1-y_2)^2)$

Interestingly, it seems the distance between the two masses keeps getting smaller we can also plot kinetic energy, or $\frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2$, which should be constant. However, we see this kinetic energy dropping.

Does this make sense? If the KE is dropping and the radius is dropping, then angular momentum won't be conserved. Performing $\vec{r}_a \times m\vec{v}$ for each yields, since $\vec{r}_a$ and $\vec{v}$ are perpendicular,

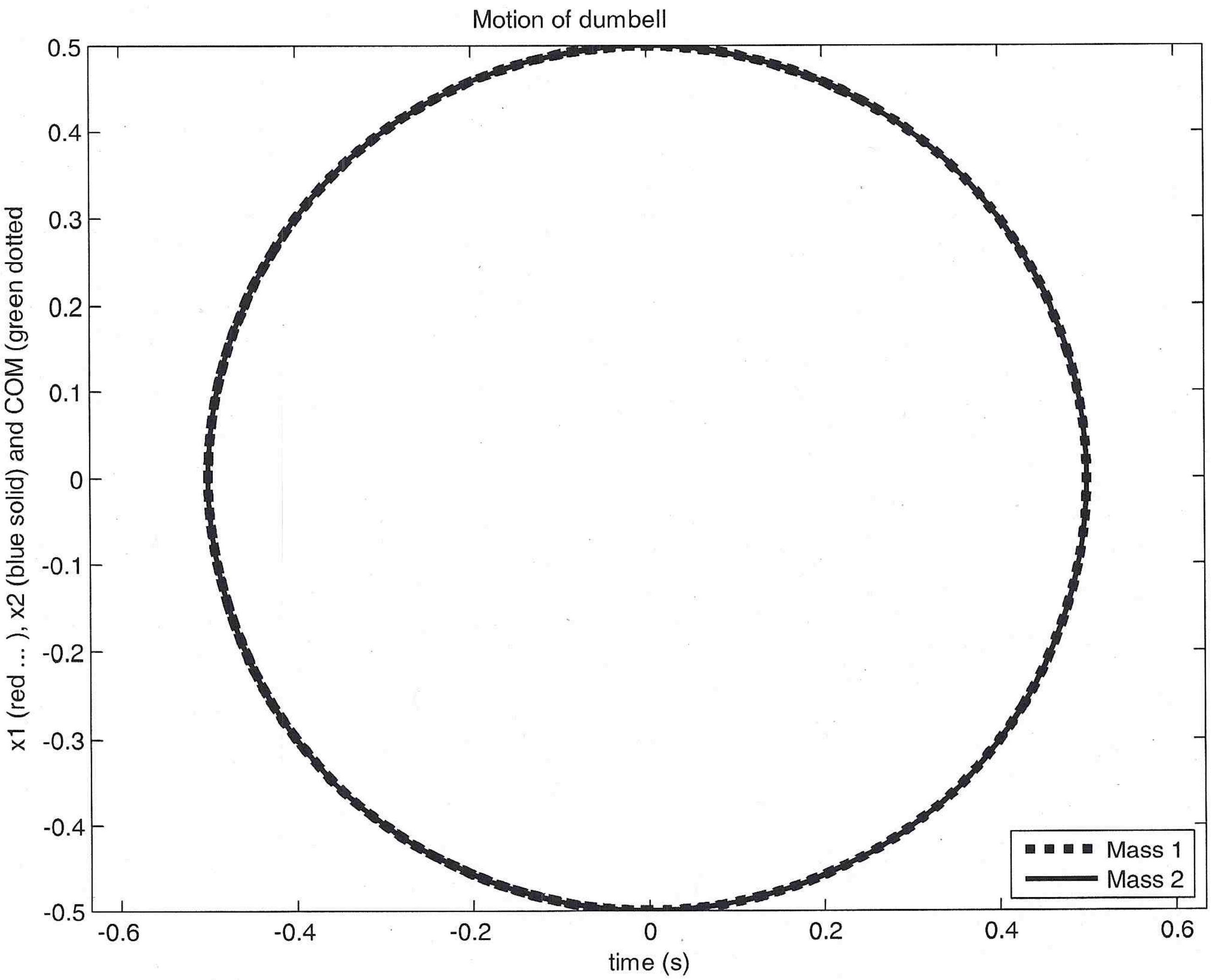
$$|\vec{r}_a||\vec{v}|m$$

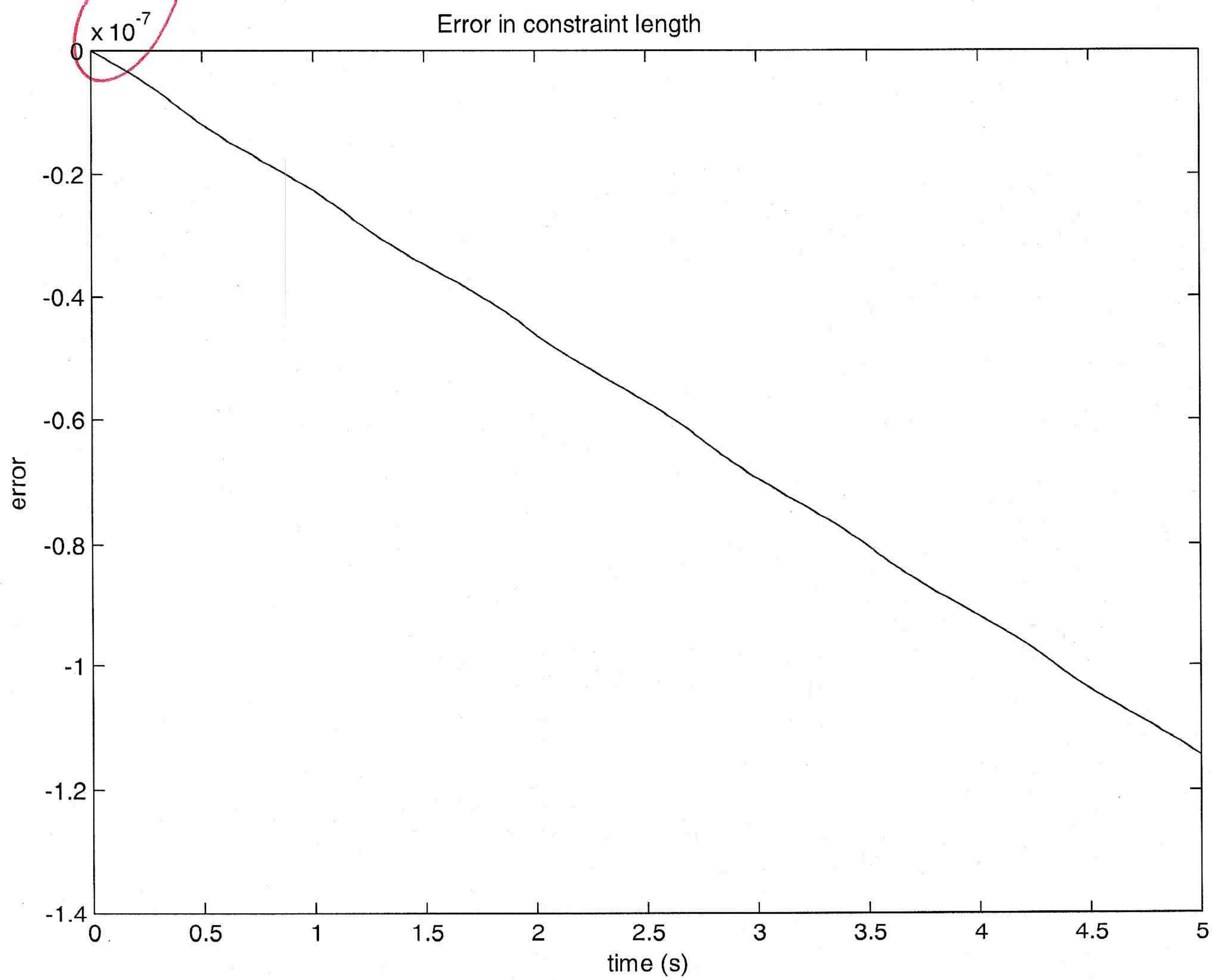
And we can see that, indeed, the sum of angular momentum is not conserved

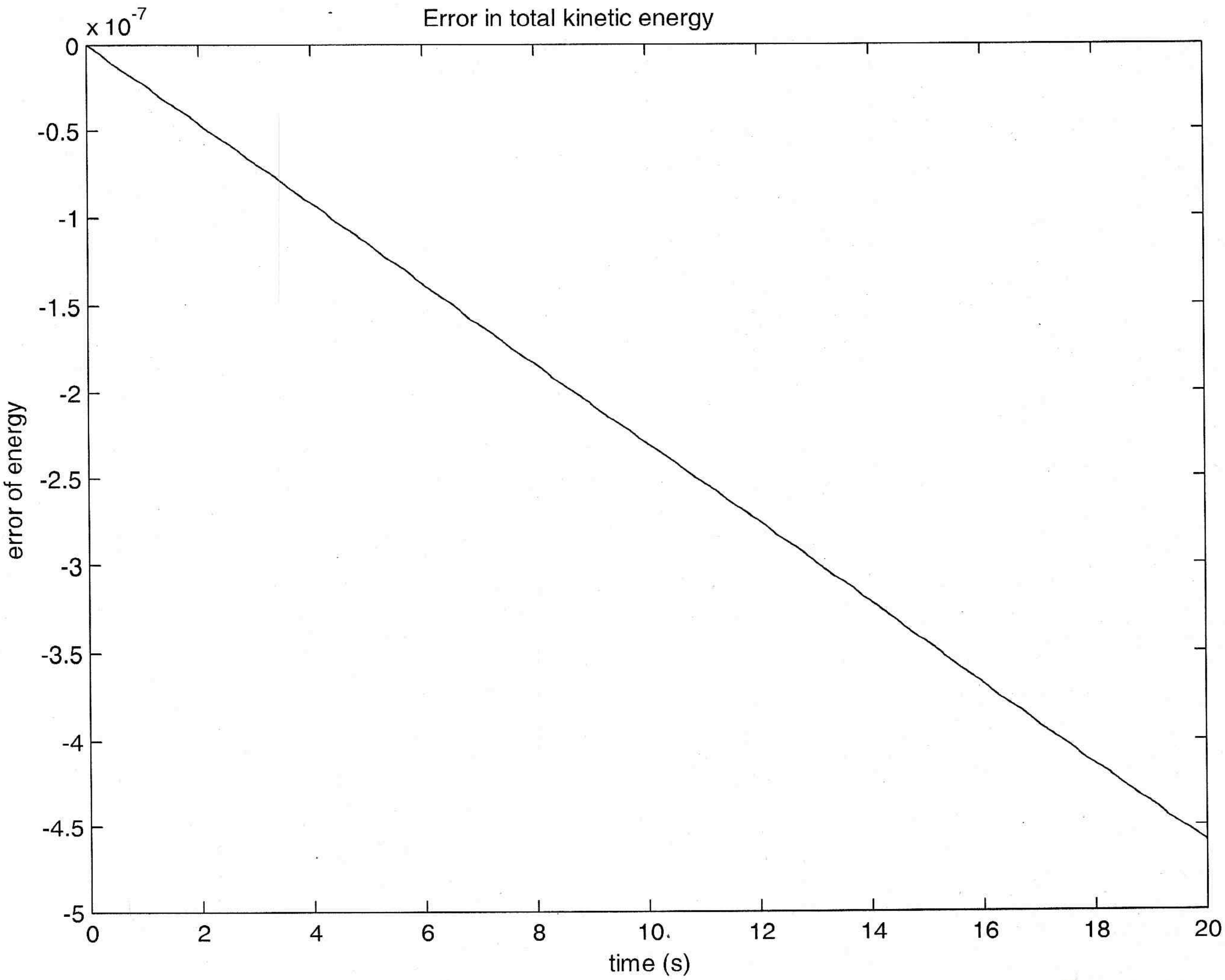
```
function [t,x1,y1,x2,y2,Statearray]=Dumbell() %by Ethan Ritz

%this function calculates the trajectory of two masses connected by a
%rigid bar, subject to velocity ICs that make it spin around

% Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%Parameters

p.m1=1;                               %the first mass
p.m2=1;                               %the second mass

p.L0=1;                               %the rest length of the spring

totalTime=20;
startTime=0;
t=(startTime:.01:totalTime)';

%Initial Conditions
pos1x_0=-.5;                              %initial positions...
pos1y_0=0;
pos2x_0=.5;
pos2y_0=0;

pos_0=[pos1x_0;pos1y_0;pos2x_0;pos2y_0]; %...put into vector

v1x_0=0;                              %initial velocities...
v1y_0=-1;
v2x_0=0;
v2y_0=1;
v_0=[v1x_0;v1y_0;v2x_0;v2y_0];            %...put into vector

%Define Initial State
State0=[pos_0;v_0];
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Call Ode23 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


options=odeset('reltol',1e-8,'abstol',1e-8);
[t, Statearray] = ode23(@eqOfMot,t, State0,options,p);

size(Statearray)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Parse Answer ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

x1 = Statearray(:,1);
y1 = Statearray(:,2);

x2 = Statearray(:,3);
```

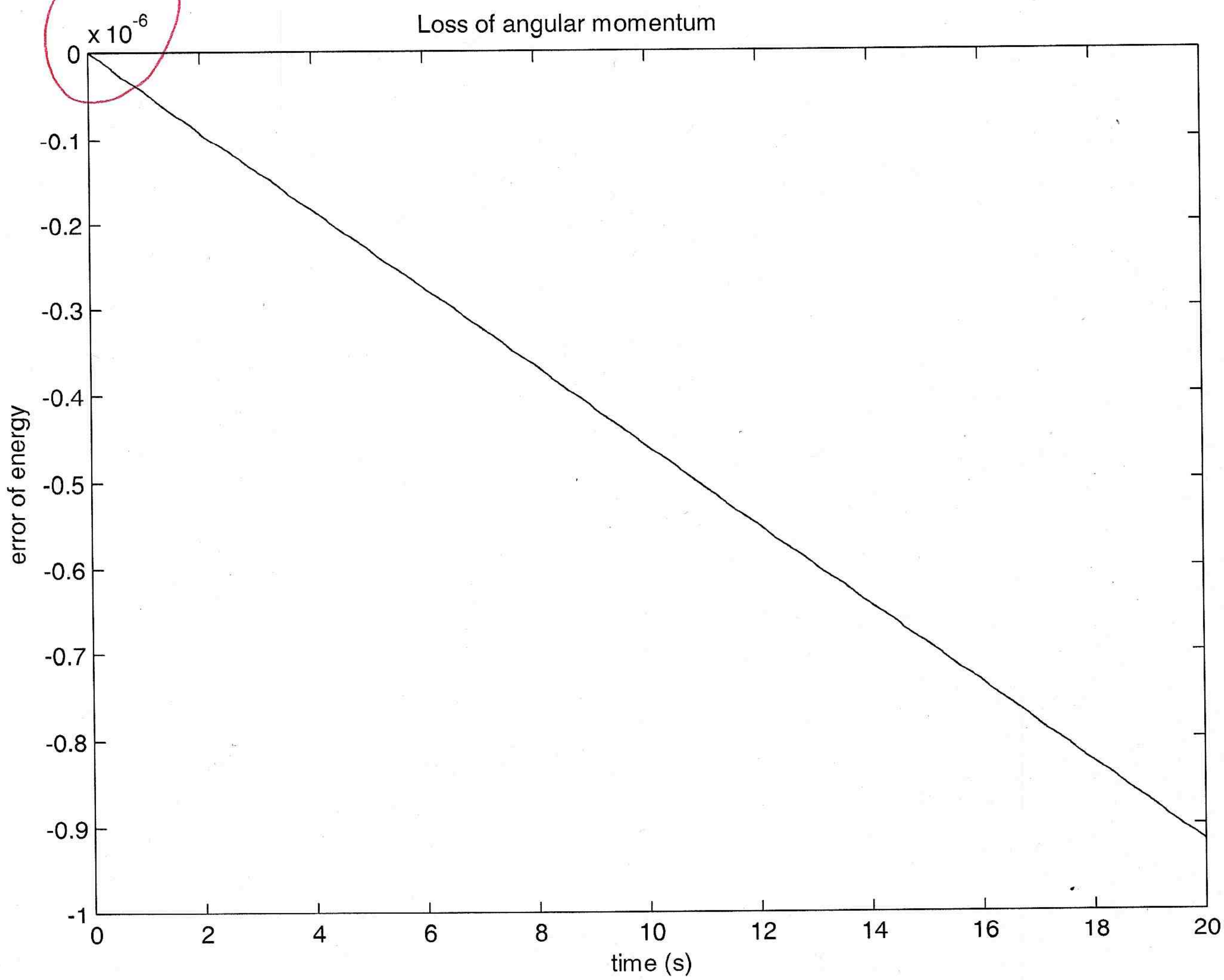
```
y2 = Statearray(:,4);

%xCOM=(x1*m1+x2*m2)/(m1+m2);          %center of mass motion

figure
plot(y1,x1,'r.')
title('Motion of dumbell');

xlabel('time (s)')
ylabel('x1 (red ... ), x2 (blue solid) and COM (green dotted')
hold on
plot(y2,x2,'b')
axis equal

legend('Mass 1','Mass 2', 'Location', 'SouthEast');

% hold off
% figure
% plot(t,x1-xCOM,'r');
% title(['Deviation of the masses from COM for k=', num2str(p.k)]);
% xlabel('time (s)')
% ylabel('|x1-xCOM| (red solid ), |x2-xCOM| (blue --)')
% hold on
% plot(t,x2-xCOM,'b--');
% legend('Mass 1','Mass 2', 'COM','Location', 'SouthEast');
% hold off
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
end

function Statedot=eqOfMot(t,xbar,p)     %the idea of a data structure 'p' was
                                       %something I leared from Ruina.
                                       %xbar is a vector where
   %xbar=[px1,py1,px2,py2,vx1,vy1,vx2,vy2]

%The goal is to get 'Statedot', which is the derivative of the state


% Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
x1=xbar(1);
y1=xbar(2);
x2=xbar(3);
y2=xbar(4);
vx1=xbar(5);
vy1=xbar(6);
vx2=xbar(7);
vy2=xbar(8);

Statedot=zeros(8,1); %this will ultimately be out output

% Say that position dot is velocity ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Statedot(1)=vx1;
```

```
Statedot(2)=vy1;
Statedot(3)=vx2;
Statedot(4)=vy2;

% Make matrix N ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
N=zeros(5);

N(1,:)=[p.m1,0,0,0,-(x2-x1)/p.L0];
N(2,:)=[0,p.m1,0,0,-(y2-y1)/p.L0];
N(3,:)=[0,0,p.m2,0,-(x1-x2)/p.L0];
N(4,:)=[0,0,0,p.m2,-(y1-y2)/p.L0];
N(5,:)=[-(x2-x1),-(y2-y1),-(x1-x2),-(y1-y2),0];

% Make the vector of knowns, "K" ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

K=[0;0;0;0;-(vx2-vx1)^2-(vy2-vy1)^2];

%find our vector of unknowns "U" and assign them ~~~~~~~~~~~~~~~~~~~~~~~~~

U=N\K; %multiplication by inv(N)
x1dd=U(1);
y1dd=U(2);
x2dd=U(3);
y2dd=U(4);
T=U(5); %but we never actually need this
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%Load them into statedot, and we are done
Statedot(5)=x1dd;
Statedot(6)=y1dd;
Statedot(7)=x2dd;
Statedot(8)=y2dd;
end
```


Motion of dumbell
x1 (red ...), x2 (blue solid) and COM (green dotted
time (s)
Mass 1
Mass 2
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.4
-0.2
0
0.2
0.4
0.6


Error in constraint length
x 10^-7
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
error
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
time (s)


Error in total kinetic energy
x 10^-7
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
error of energy
0
2
4
6
8
10
12
14
16
18
20
time (s)


Loss of angular momentum
x 10^-6
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
-0.9
-1
error of energy
0
2
4
6
8
10
12
14
16
18
20
time (s)

V. good