`sTopEuler` documentation

sTopEuler v0.5 written by Alexander Erlich (mailto:alexander.erlich@gmail.com)
Approved by MATLAB File Exchange: July 27th, 2010. Last update: July 2nd, 2010.
File Exchange URL: http://www.mathworks.com/matlabcentral/fileexchange/28309

General information

`sTopMain` shows an animated spinning top with Cardan mounting from raw animation data. An out of the box example call would be

```
>> sTopMain('animation_A.dat')
```

There is only one argument: a string (`fileToRead` in the source code), specifying the filename of an animation data file. Let (φ, ϑ, ψ) to be Cardan angles, see fig. 0.1. The data file must contain four columns: the current time t and the Cardan angles evaluated at the time t. The data file may not contain any header lines and its columns must be arranged like this (shown with some arbitrary example data):

t	φ	ϑ	ψ
⋮	⋮	⋮	⋮
2.6943	11.4475	1.5840	2.5266
2.7240	11.6341	1.5736	2.5279
2.7537	11.8207	1.5719	2.5282
⋮	⋮	⋮	⋮

The provided data files (`animation_A.dat` and `animation_B.dat`) result from a Runge-Kutta integration of a system of ODEs derived from the regular motion of an Euler top. The integration spans over a time interval of 20 seconds and the initial parameters are chosen according to motion on a separatrix (homoclinic orbit)

Explainations in the corresponding `m`-files (see below) will provide an understanding of how the animations are created. The key concept (in a nutshell) is: The rings and bars are created using the `surf` command. Rotation of the objects is performed by self-written individually computed rotation matrices (applied to the rings and bars). On each looped frame, the coordinates of the `surf` handles are updated using the `drawnow` command.

Future versions

In future versions, scripts both in Mathematica and Matlab to create animation files (depending on given parameters) shall be provided. The content of such files can be animated using the program presented here. Also, in future versions, it will be possible to create `avi` files from the animation. This feature will ensure the animation timing is always correct.

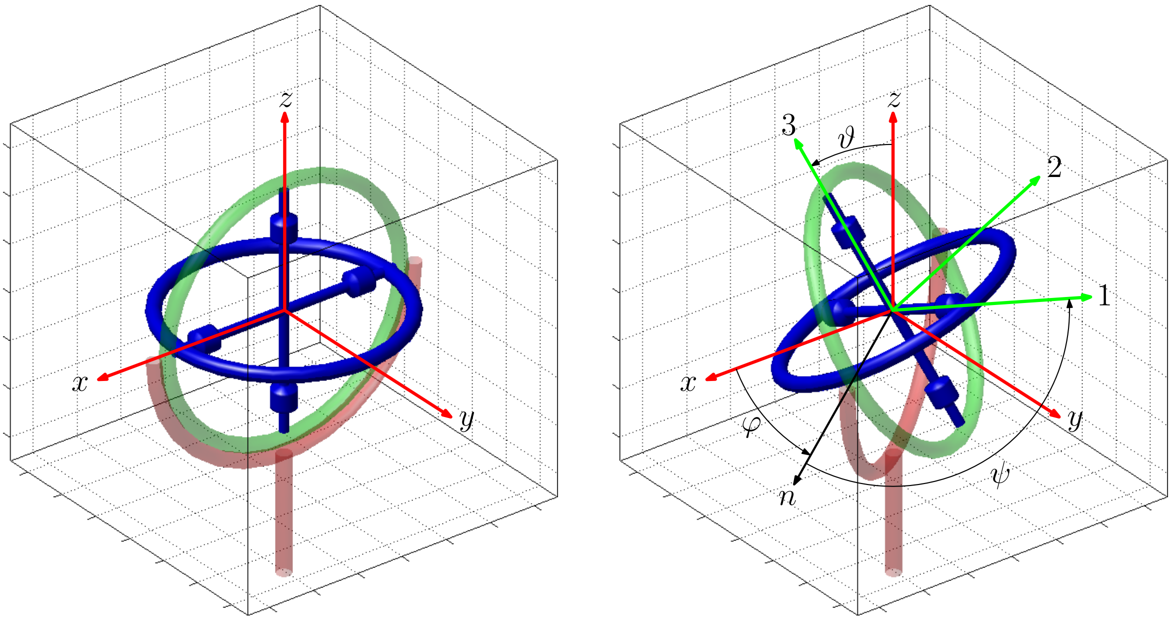

Abb. 0.1: Cardan angles (φ, ϑ, ψ) transforming a space-fixed system (x, y, z) into a body-fixed system (1, 2, 3) by subsequent rotations.

createInnerBars.m

```
function [ Xbar1,Ybar1,Zbar1,Xbar2,Ybar2,Zbar2 ] = ...
    createInnerBars( bar1R1,bar1R2,bar2R1,bar2R2)
%CREATEINNERBARS Creates the bars of the inner (blue) ring
%
%   bar[i]R[j]: There are two bars which are perpendicular to each other.
%   They are melted with the inner ring, so the ring together with the bars
%   form the inner part of the top.
%   R1 always describes the radius of the bar itself. Additionally, both
%   bars have two attatched weigths each. Both of the attatched weigths are
%   cylinders of R2.

[Zbar1,Ybar1,Xbar1]=cylinderWeights(bar1R1,bar1R2);
Xbar1=Xbar1-0.5;
Xbar1=Xbar1*10;
[Xbar2,Ybar2,Zbar2]=cylinderWeights(bar2R1,bar2R2);
Zbar2=Zbar2-0.5;
Zbar2=Zbar2*10;
```

createOuterBars.m

```
function [ Xupper,Yupper,Zupper,Xlower,Ylower,Zlower ] = ...
    createOuterBars( radius )
%CREATEOUTERBARS Creates the red motionless supporting bar

[Xupper,Yupper,Zupper] = cylinder(radius);
Zupper=Zupper.*5;
Zupper=Zupper+5.9;
[Xlower,Ylower,Zlower] = cylinder(radius);
Zlower=Zlower.*5;
Zlower=Zlower-10.9;
```

cylinderWeights.m

```
function [X,Y,Z] = cylinderWeights(r1,r2)
%CYLINDERWEIGHTS creates the innermost blue bars with heavy looking cylinders
%representing attatched weights.
t=0:0.01:1;
x=t;

for i=1:numel(t)
    if ((i>10 && i<20) || (i>80 && i<90))
        x(i)=r2;
    else
        x(i)=r1;
    end
end
[X,Y,Z]=cylinder(x,30);
```

torus.m

```
function [x, y, z] = torus (a, n, r, kpi)
% TORUS Generate a torus.
% torus (r, n, a, kpi) generates a plot of a
% torus with central radius a and
% lateral radius r.
% n controls the number of facets
% on the surface.
% kpi makes it possible to draw a whole torus,
% or e.g. half of it.
%
% This script is a modification of a
% program from:

% MATLAB Primer, 6th Edition
% Kermit Sigmon and Timothy A. Davis
% Section 11.5, page 65.

theta = -pi * (0:2:kpi*n)/n ;
phi = 2*pi* (0:2:n)'/n ;
x = (a + r*cos(phi)) * cos(theta) ;
y = (a + r*cos(phi)) * sin(theta) ;
z = r * sin(phi) * ones(size(theta)) ;
```

EuMat.m

```
function [ ret_mat ] = EuMat( phi, theta, psi )
%EUMAT returns a rotation matrix rotating an object by the Cardan angles
%(phi, theta, psi)

    Dphi = [cos(phi),-sin(phi),0;sin(phi),cos(phi),0;0,0,1];
    Dtheta = [1,0,0;0,cos(theta),-sin(theta);0,sin(theta), cos(theta)];
    Dpsi = [cos(psi),-sin(psi),0;sin(psi),cos(psi),0;0,0,1];

    ret_mat = Dphi*Dtheta*Dpsi;
```

multiplyEuMat.m

```
function [ Xt,Yt,Zt ] = multiplyEuMat( EuMat, X,Y,Z )
%MULTIPLYEUMAT takes the X, Y, Z coordinates of an object and returns the
%coordinates Xt, Yt, Zt of same object rotated using a rotation matrix

Xt=X;
Yt=Y;
Zt=Z;

resvec=[1;1;1];
for i=1:numel(X)
   temp=[X(i);Y(i);Z(i)];
   resvec=EuMat*temp;
   Xt(i)=resvec(1);
   Yt(i)=resvec(2);
   Zt(i)=resvec(3);
end
```

sTopEuler.m

```
% coordinates of the three rings (blue, red and green)
[X1,Y1,Z1]=torus(5-0.3,30,0.3,2); % the top (blue)
[X2,Z2,Y2]=torus(5+0.3,30,0.3,2); % inner gimbal (green, semi-transparent)
[X3,Z3,Y3]=torus(5+0.9,30,0.3,1); % outer gimbal (red, semi-transparent)

% handles to surf plots of the three rings
H1=surf(X1,Y1,Z1,'EdgeColor','none','FaceColor','blue'); hold on;
H2=surf(X2,Y2,Z2,'EdgeColor','none','FaceColor','green','FaceAlpha',0.3);
H3=surf(X3,Y3,Z3,'EdgeColor','none','FaceColor','red','FaceAlpha',0.3);

% creates the bars inside the inner (blue) ring. The wide cylinders
% represent weights.
[Xbar1,Ybar1,Zbar1,Xbar2,Ybar2,Zbar2]=createInnerBars(0.2,0.5,0.2,0.5);
bar1=surf(Xbar1,Ybar1,Zbar1,'EdgeColor','none','FaceColor','blue');
bar2=surf(Xbar2,Ybar2,Zbar2,'EdgeColor','none','FaceColor','blue');

% the red motionless supporting bar
[ Xupper,Yupper,Zupper,Xlower,Ylower,Zlower ] = createOuterBars( 0.3 );
surf(Xlower,Ylower,Zlower,...
    'EdgeColor','none','FaceColor','red','FaceAlpha',0.3);

hold off;
 camlight left;
 lighting gouraud; % phong is more demanding but gives nicer results
 bb=7;
 axis([-bb bb -bb bb -bb bb]);
axis square;
xlabel('x'); ylabel('y'); zlabel('z');
```

sTopMain.m

```
function [F] = sTopMain(fileToRead)
%STOPMAIN Plots an animated spinning top with Cardan mounting from raw
%animation data.
%
%   example call: sTopMain('animation_A.dat')
%
%   See PDF documentation for details.
%
%   Alexander Erlich (alexander.erlich@gmail.com)

rawData = importdata(fileToRead);
time = rawData(:,1);
phi = rawData(:,2);
theta = -1.*rawData(:,3);
psi = rawData(:,4);

sTopEuler;  % creating surf plots for the spinning Top in its initial position
            % (phi, theta, psi) = (0, 0, 0)

for i=1:length(phi)-1
    if (ishandle(H1)~= 0)

        % computing the new position
        [Xtemp1,Ytemp1,Ztemp1]=...
            multiplyEuMat(EuMat(phi(i),theta(i),psi(i)),X1,Y1,Z1);
        [Xtemp2,Ytemp2,Ztemp2]=...
            multiplyEuMat(EuMat(phi(i),theta(i),0),X2,Y2,Z2);
        [Xtemp3,Ytemp3,Ztemp3]=...
            multiplyEuMat(EuMat(phi(i),0,0),X3,Y3,Z3);
        [Xtempbar1,Ytempbar1,Ztempbar1]=...
            multiplyEuMat(EuMat(phi(i),theta(i),psi(i)),Xbar1,Ybar1,Zbar1);
        [Xtempbar2,Ytempbar2,Ztempbar2]=...
            multiplyEuMat(EuMat(phi(i),theta(i),psi(i)),Xbar2,Ybar2,Zbar2);

        % updating the new position
        set(H1,'XData',Xtemp1,'YData',Ytemp1,'ZData',Ztemp1);
        set(H2,'XData',Xtemp2,'YData',Ytemp2,'ZData',Ztemp2);
        set(H3,'XData',Xtemp3,'YData',Ytemp3,'ZData',Ztemp3);
        set(bar1,'XData',Xtempbar1,'YData',Ytempbar1,'ZData',Ztempbar1);
        set(bar2,'XData',Xtempbar2,'YData',Ytempbar2,'ZData',Ztempbar2);

        drawnow % drawing the new position
        pause(time(i+1)-time(i)); % pausing until next frame is shown
    end
end
```