ORBITAL MECHANICS - MIDTERM

Table of Contents

Ex. 8

```
clear all
close all
%initial conditions
a1=12000;           %semi major axis [km]
e1=.2;              %eccentricity
i1=45*pi/180;       %inclination [rad]
w1=0;               %argument of perigee [rad]
omega1=0;           %RAAN [rad]
f1=0;               %true anomaly [rad]
p1=a1*(1-e1^2);

%final conditions
a2=13000;
e2=.1;
i2=45*pi/180;
w2=0;
omega2=0;
f2=120*pi/180;
p2=a2*(1-e2^2);

dt=60*60;           %transfer time [sec]
mu=398600;          %gravitational parameter [km^3/s^2]
```

Getting position and velocity from orbital elements

```
[r1vec,v1vec] = RVFromCOE( a1,i1,omega1,w1,e1,f1, mu );
[r2vec,v2vec] = RVFromCOE( a2,i2,omega2,w2,e2,f2, mu );
```

Solving Lambert's Problem

```
[v1t,v2t,dTheta] = LambertSolver( r1vec, r2vec, dt, mu );
```

Orbital Elements for the transfer orbit

```
[at,it,omegat,wt,et,ft,ht] = OrbitalElementsFromRV( r1vec, v1t, mu );
```

Calculating delta v's

```
dv1=norm(v1t-v1vec);
dv2=norm(-v2t+v2vec);
fprintf('deltaV 1 = %.2f km/s\ndeltaV 2 = %.2f km/s\n\n',dv1,dv2)
fprintf('Orbital elements of the transfer orbit\n')
fprintf('Semi major axis: %.0f km\n',at)
fprintf('Inclination: %.0f°\n',it*180/pi)
fprintf('Right ascension of the ascending node: %.0f°\n',omegat*180/
pi)
fprintf('Argument of periapsis: %.2f°\n',wt*180/pi)
fprintf('Eccentricity: %.2f\n',et)
```

```
deltaV 1 = 0.24 km/s
deltaV 2 = 0.91 km/s

Orbital elements of the transfer orbit
Semi major axis: 12993 km
Inclination: 45°
Right ascension of the ascending node: 0°
Argument of periapsis: 5.95°
Eccentricity: 0.26
```

COMMENTS

```
%{
The "LambertSolver" function provides a solution for Lambert's problem
using universal variables. More details are given in the
sub-functions. The method described in class is used, just normalized
 to
universal variables so it is applicable to every conic section and it
 takes
into account of particular cases and singularity. I attach every
sub-function used for that.

VITTORIO BARALDI
%}
```

Published with MATLAB® R2019b