ECE 6390 Homework 2: Look Angles

Solutions

Problem 1: Numerical Analysis of Orbits

The continuous time equations we start with are:

$$\ddot{r} = r\dot{\theta}^2 - \frac{GM_p}{r^2} \qquad \ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r}$$

First, let our discrete representations of r and θ be sampled versions of $r(t)$ and $\theta(t)$:

$$r_n = r\left(n\Delta t - \frac{1}{2}\Delta t\right) \qquad \theta_n = \theta\left(n\Delta t - \frac{1}{2}\Delta t\right)$$

The sampling offset $1/2\Delta t$ is there so that our estimate of the derivative at the points $n\Delta t$ is in terms of these offset points:

$$\dot{r}(n\Delta t) \approx \frac{r\left(n\Delta t + \frac{1}{2}\Delta t\right) - r\left(n\Delta t - \frac{1}{2}\Delta t\right)}{\Delta t} = \frac{r_{n+1} - r_n}{\Delta t} = \frac{\Delta r_n}{\Delta t} \tag{1}$$

$$\dot{\theta}(n\Delta t) \approx \frac{\theta\left(n\Delta t + \frac{1}{2}\Delta t\right) - \theta\left(n\Delta t - \frac{1}{2}\Delta t\right)}{\Delta t} = \frac{\theta_{n+1} - \theta_n}{\Delta t} = \frac{\Delta \theta_n}{\Delta t} \tag{2}$$

In the last equalities of each line, we have defined two new state variables of our discrete system:

$$\boxed{\Delta r_n = r_{n+1} - r_n} \qquad \boxed{\Delta\theta_n = \theta_{n+1} - \theta_n}$$

Now we need estimates of the samples of the second derivatives:

$$\ddot{r}(n\Delta t) \approx \frac{\dot{r}(n\Delta t + \Delta t) - \dot{r}(n\Delta t)}{\Delta t} = \frac{\Delta r_{n+1} - \Delta r_n}{\Delta t^2} \tag{3}$$

$$\ddot{\theta}(n\Delta t) \approx \frac{\dot{\theta}(n\Delta t + \Delta t) - \dot{\theta}(n\Delta t)}{\Delta t} = \frac{\Delta \theta_{n+1} - \Delta \theta_n}{\Delta t^2} \tag{4}$$

Finally, we'll sample the given continuous equations at the times $n\Delta t$, and then plug in the relationships we've found between the continuous and the discrete quantities:

$$\ddot{r}(n\Delta t) = r(n\Delta t)\dot{\theta}(n\Delta t)^2 - \frac{GM_p}{r(n\Delta t)^2} \tag{5}$$

$$\ddot{\theta}(n\Delta t) = -\frac{2\dot{r}(n\Delta t)\dot{\theta}(n\Delta t)}{r(n\Delta t)} \tag{6}$$

We have discrete expressions for most of the terms involving r and θ above by using (1),(2),(3), and (4); the exception is the term $r(n\Delta t)$. But notice that we can estimate this quantity based on the value of $r_n = r(n\Delta t - \frac{1}{2}\Delta t)$ by using a first order Taylor series approximation:

$$r(n\Delta t) \approx r\left(n\Delta t - \frac{1}{2}\Delta t\right) + \dot{r}\left(n\Delta t - \frac{1}{2}\Delta t\right)\frac{\Delta t}{2} = r_n + \frac{r(n\Delta t) - r(n\Delta t - \Delta t)}{\cancel{\Delta t}}\frac{\cancel{\Delta t}}{2}$$

Now, although the values of the function are not the same over a half sample, the value of the change can be assumed to be small over the window $\Delta t/2$. That is

$$r(n\Delta t) - r(n\Delta t - \Delta t) \approx r\left(n\Delta t + \frac{1}{2}\Delta t\right) - r\left(n\Delta t - \frac{1}{2}\Delta t\right) = \frac{\Delta r_n}{\Delta t}$$

So the term $r(n\Delta t)$ is:

$$r(n\Delta t) \approx r_n + \frac{1}{2}\Delta r_n \tag{7}$$

Now plug (1),(2),(3),(4), and (7) into (5) and (6):

$$\frac{\Delta r_{n+1} - \Delta r_n}{\Delta t^2} = \left(r_n + \frac{1}{2}\Delta r_n\right)\left(\frac{\Delta\theta_n}{\Delta t}\right)^2 - \frac{GM_p}{\left(r_n + \frac{1}{2}\Delta r_n\right)^2}$$

$$\frac{\Delta\theta_{n+1} - \Delta\theta_n}{\Delta t^2} = -\frac{2\frac{\Delta r_n}{\Delta t}\frac{\Delta\theta_n}{\Delta t}}{r_n + \frac{1}{2}\Delta r_n}$$

Finally, we use the approximation:

$$\left(r_n + \frac{1}{2}\Delta r_n\right)^2 = r_n^2 + r_n\Delta r_n + \frac{1}{4}\Delta r_n^2 \approx r_n^2$$

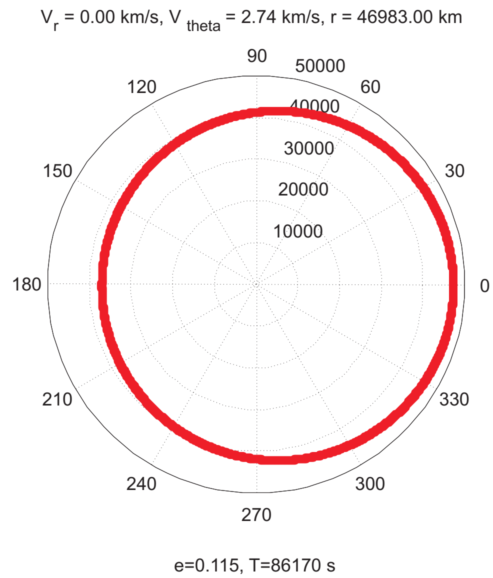
And rearrange to get the final two equations:

$$\boxed{\Delta r_{n+1} = \Delta r_n + \left(r_n + \frac{1}{2}\Delta r_n\right)\Delta\theta_n^2 - \frac{GM_p}{r_n^2}\Delta t^2}$$

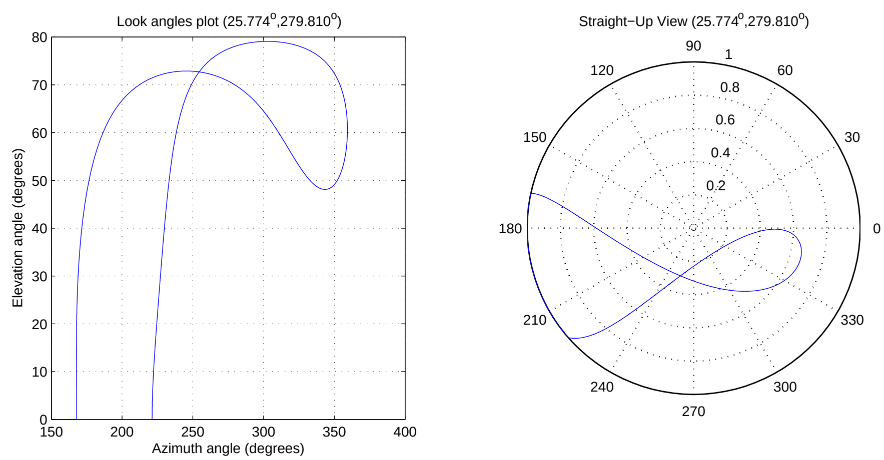
$$\boxed{\Delta\theta_{n+1} = \Delta\theta_n - \frac{2\Delta r_n \Delta\theta_n}{r_n + \frac{1}{2}\Delta r_n}}$$

Problem 2: Look angles simulation

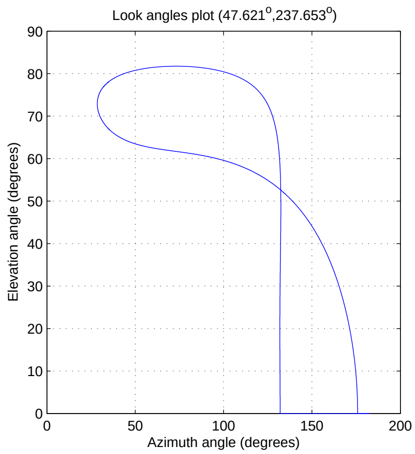
Below are the parameters and orbital sketch for a *Tundra* orbit:

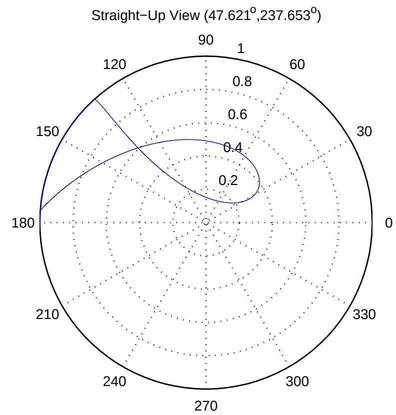


For the Miami location, the sky path is shown below in two fashions: an azimuth-elevation rectangular plot (left) and a polar plot of azimuth and cosine of elevation (to give a rough idea of what the path would look like if you were lying on your back in a field). This satellite is in view (ideally) almost exactly 18 hours.



For the Seattle location, the sky path is shown below. This satellite is in view (ideally) 15 hours and 36 minutes. Of course, the satellite probably does not become audible below 30 degrees of elevation because of the oblique, longer path and the likelihood of terrain and object blockage on the horizon. Still, these orbits obviously keep the satellite high overhead for a long period of time.


Look angles plot (47.621°,237.653°)
Elevation angle (degrees)
Azimuth angle (degrees)
0
10
20
30
40
50
60
70
80
90
0
50
100
150
200


Straight−Up View (47.621°,237.653°)
90
1
120
60
0.8
0.6
150
30
0.4
0.2
180
0
210
330
240
300
270

```
% Code by Uzoma Onunkwo, Greg Durgin August 2004
% Modified by Raj Bhattacharjea September 2009
clear all
close all

%Step 1: Generate the orbit--------------------------------------------------

% Physical Constants
G = 6.672e-11; % Gravitational Constant (Newton meters^2 kilograms^-2)
Mp = 5.974e24; % Mass of the Planet Earth (kilograms)
MAX_ITERATION = 1e5; % Largest number of iterations
CF = pi/180; % multiplication conversion factor from degrees to radians

% Initial Conditions
dt = 10; % increment (seconds)
r = 46.983e6; r_o = r; % initial radius (meters)
theta = 0; % initial angle (radians)
V_r = 0; % r-component of velocity (meters/second)
V_theta = 2.741e3; % theta-component of velocity (meters/second)

% Convert to Discrete Conditions
dr = V_r*dt; % meters
dtheta = V_theta/r*dt; % radians

% Initialize values for looped calculation
r_0 = r;
theta_0 = theta;
all_theta = theta;
all_r = r;
while abs(theta-theta_0) <= 2*pi && length(all_theta) < MAX_ITERATION,

% compute new state information
  r_new = r + dr;
  theta_new = theta + dtheta;
  dr_new = dr + ((r+0.5*dr)*dtheta^2 - G*Mp/(r^2)*(dt^2));
  dtheta_new = dtheta - (2*dr*dtheta/(r+0.5*dr));

% save and print new state information
  all_theta = [all_theta; theta_new ];
  all_r = [all_r; r_new ];

% reset variables for new calculation
  r = r_new; dr = dr_new;
  theta = theta_new; dtheta = dtheta_new;
end;
```

```matlab
% calculate period and eccentricity and put it on the plot
ecc = (max(all_r)-min(all_r))/(min(all_r)+max(all_r));
Period = dt*length(all_theta); % unit in seconds (same as unit of dt)
figure(1), polar( all_theta , all_r/1000, 'r.' );
title( sprintf('V_r = %1.2f km/s, V_{theta} = %1.2f km/s, r = %1.2f km', ...
     V_r/1000,V_theta/1000,r_o/1000));
if(length(all_theta) < MAX_ITERATION)
   xlabel( sprintf('e=%1.3f, T=%1.0f s',ecc, Period) );
else
   xlabel( 'Escape Velocity Reached' );
end;

%Step 2: Perform inclination -----------------------------------------
phi_i = 63.4*CF; %inclincation angle in radians
%go to cartesian
x = all_r .* cos(all_theta);
y = all_r .* sin(all_theta);

%now incline the orbit
xp = x .* cos(phi_i);
zp = x .* sin(phi_i);
yp = y;

%Step 3: Calculate Satellite Lat and Lon ------------------------------
Tsid = 23*3600+56*60 +4;
Re = 6378.14e3; % Radius of the Planet Earth (meters)
aLong = -96*CF; %apogee longitude

s_Lat = atan(zp./(xp.^2+yp.^2).^.5); %cartesian->polar in one coordinate
%this shifts the apogee to -96 degrees long, and accounts for earth
%rotation
s_Long = aLong + atan2(yp,xp) - (2*pi)*([0:length(xp)-1]')*dt/Tsid;

%Step 4: look angles (Azimuth and Elevation angles) --------------------
%Miami
Lat = 25.774252*CF; Long = 2*pi-80.190262*CF;

%use look angle formulas from formula sheet
Gamma = ...
  acos(sin(s_Lat).*sin(Lat) + cos(s_Lat).*cos(Lat).*cos(s_Long - Long));
Alpha = ...
  asin(sin(abs(Long - s_Long)).*(cos(s_Lat)./sin(Gamma)));

% Elevation angle in degrees
El_angle = (acos(sin(Gamma)./sqrt(1+(Re./all_r).^2 - ...
     2*(Re./all_r).*cos(Lat).*cos(s_Long - Long).*cos(s_Lat))));

%imaginary elevation means its below the horizon, zero these out
I = imag(El_angle)~=0; El_angle(I) = 0;
```

```
% Azimuth angle in degrees
% make sure and wrap the satellite longitude values to the correct range
I = s_Long < -2*pi; s_Long(I) = s_Long(I) + 2*pi;
Az_angle = zeros(size(s_Long));
% SSP is NE of ES
I = s_Long >= Long & s_Lat >= Lat; Az_angle(I) = Alpha(I);
% SSP is SE of ES
I = s_Long >= Long & s_Lat <  Lat; Az_angle(I) = pi - Alpha(I);
% SSP is NW of ES
I = s_Long <  Long & s_Lat >= Lat; Az_angle(I) = 2*pi - Alpha(I);
% SSP is SW of ES
I = s_Long <  Long & s_Lat <  Lat; Az_angle(I) = Alpha(I) + pi;

%make sure to wrap the azimuth angles to the right range
I = Az_angle> 2*pi; Az_angle(I) = Az_angle(I) - 2*pi;

%plot the rectangular and straight up views
figure('OuterPosition', [0 0 900 480]);
subplot(1,2,1), plot(Az_angle/CF, El_angle/CF);
xlabel('Azimuth angle (degrees)');
ylabel('Elevation angle (degrees)');
title(sprintf('Look angles plot (%1.3f^{o},%1.3f^{o})', Lat/CF, Long/CF));
grid on;

subplot(1,2,2), polar( Az_angle, cos(El_angle) );
title(sprintf('Straight-Up View (%1.3f^{o},%1.3f^{o})', Lat/CF, Long/CF));
grid on;

%output the time above the horizon
tVisible = length(El_angle(El_angle~=0))*dt


%Seattle
Lat = 47.620973*CF; Long = 2*pi-122.347276*CF;

%use look angle formulas from formula sheet
Gamma = ...
  acos(sin(s_Lat).*sin(Lat) + cos(s_Lat).*cos(Lat).*cos(s_Long - Long));
Alpha = ...
  asin(sin(abs(Long - s_Long)).*(cos(s_Lat)./sin(Gamma)));

% Elevation angle in degrees
El_angle = (acos(sin(Gamma)./sqrt(1+(Re./all_r).^2 - ...
    2*(Re./all_r).*cos(Lat).*cos(s_Long - Long).*cos(s_Lat))));

%imaginary elevation means its below the horizon, zero these out
I = imag(El_angle)~=0; El_angle(I) = 0;
```

```
% Azimuth angle in degrees
% make sure and wrap the satellite longitude values to the correct range
I = s_Long < -2*pi; s_Long(I) = s_Long(I) + 2*pi;
Az_angle = zeros(size(s_Long));
% SSP is NE of ES
I = s_Long >= Long & s_Lat >= Lat; Az_angle(I) = Alpha(I);
% SSP is SE of ES
I = s_Long >= Long & s_Lat <  Lat; Az_angle(I) = pi - Alpha(I);
% SSP is NW of ES
I = s_Long <  Long & s_Lat >= Lat; Az_angle(I) = 2*pi - Alpha(I);
% SSP is SW of ES
I = s_Long <  Long & s_Lat <  Lat; Az_angle(I) = Alpha(I) + pi;

%make sure to wrap the azimuth angles to the right range
I = Az_angle> 2*pi; Az_angle(I) = Az_angle(I) - 2*pi;

%plot the rectangular and straight up views
figure('OuterPosition', [0 0 900 480]);
subplot(1,2,1), plot(Az_angle/CF, El_angle/CF);
xlabel('Azimuth angle (degrees)'), ylabel('Elevation angle (degrees)');
title(sprintf('Look angles plot (%1.3f^{o},%1.3f^{o})', Lat/CF, Long/CF));
grid on;

subplot(1,2,2), polar( Az_angle, cos(El_angle) );
title(sprintf('Straight-Up View (%1.3f^{o},%1.3f^{o})', Lat/CF, Long/CF));
grid on;

%output the time above the horizon
tVisible = length(El_angle(El_angle~=0))*dt

%for previous years, use the following locations
%Van Leer
%Lat = 33.775968*CF; Long=2*pi-84.397131*CF;
% Bangor Maine's coordinates (degrees)
%Lat = 44.8*CF; Long = 2*pi-68.8*CF;
```