```
clear all; clc; close all;

rp = 500+6378;
mu = 3.9860044189e5;
e = 1.5;
i = 35*pi/180;
W = 150*pi/180;
w = 115*pi/180;
true_ana = 0*pi/180;
h = sqrt(rp*mu*(1+e*cos(true_ana)));

dt = 3600*2;
Me = mu^2/h^3*(e^2-1)^1.5*dt;
F_guess = 1.8;
func = @(F) e*sinh(F)-F-Me;
F = fzero(func,F_guess);
true_ana = 2*atan(sqrt((e+1)/(e-1))*tanh(F/2));

[ro,vo,r,v,QXx] = OrbElem2StateVec(h,e,i,w,W,true_ana);

fprintf('Position Vector: [%0.0f %0.0f %0.0f] km \n', ro)
fprintf('Velocity Vector: [%0.3f %0.3f %0.3f] km/s \n', vo)


function [RX,VX,r,v,QXx] = OrbElem2StateVec(h,e,i,w,W,true_ano)
%This function will take orbital elements and compute two state vectors r and
v
% r and v must be 3-D vectors
mu = 3.9860044189e5;
r = (h^2/mu)/(1+e*cos(true_ano))*[cos(true_ano) sin(true_ano) 0]';
v = mu/h*[-sin(true_ano) (e+cos(true_ano)) 0]';
R3_W = [cos(W)  sin(W) 0;
      -sin(W) cos(W) 0;
      0       0      1;];
R1_i = [1 0 0;
        0 cos(i) sin(i);
        0 -sin(i) cos(i);];
R3_w = [ cos(w) sin(w) 0;
        -sin(w) cos(w) 0
        0 0 1;];
QXx = (R3_w) *(R1_i) *(R3_W);

RX = QXx.'*r;
VX = QXx.'*v;


end
```

Position Vector: [46011 13466 -24274] km

Velocity Vector: [4.850 2.893 -3.452] km/s