

Group Exam 2



Date of Submission:
March 6, 2019

Grace Day

JiYoung Hwang

Aaron Scott

Thorne Wolfenbarger

Submitted to:
Professor Kaela Martin
College of Engineering

In Partial Fulfillment
of the Requirements of

AE 313
Space Mechanics
Spring, 2019

Contents

1	Current Position And Velocity	1
2	Time To Send The Message	1
3	New Position And Velocity	2
4	Latitude And Longitude	2
5	Ground Track	4
6	Can You Send The Message?	6
7	Why?	8

Knowns

$$\begin{aligned} a &= 42,170 \text{ km} & e &= 0.01053 & \Omega &= 14.89^\circ \\ \omega &= 318.4^\circ & i &= 14.37^\circ & \theta_1^* &= 184.2^\circ \end{aligned}$$

1 Current Position And Velocity

Find the current position and velocity (\vec{r}_1, \vec{v}_1) in ECI and perifocal coordinates.

$$\begin{aligned} \vec{r}_1 &= (-3.9158\hat{x} + 1.5532\hat{y} + 0.6423\hat{z}) \cdot 10^4 \text{ km} = (-4.2498\hat{e} - 0.3121\hat{p}) \cdot 10^4 \text{ km} \\ \vec{v}_1 &= -1.1820\hat{x} - 2.7383\hat{y} - 0.6002\hat{z} \text{ km/s} = 0.2252\hat{e} - 3.0340\hat{p} \text{ km/s} \end{aligned}$$

```
1 % 1. Find the current position and velocity in ECI and perifocal
  coordinates
2 p = a*(1-e^2);
3 h = sqrt(MU*p);
4 r1 = p/(1+e*cosd(true_a1));
5
6 vr1_rth = [r1 0 0]';
7 vv1_rth = [h*e/p*sind(true_a1) h/r1 0]';
8
9 AOL1 = AOP + true_a1;
10 rth_eci = rot_rth_eci(RAAN,inc,AOL1);
11
12 vr1_eci = rth_eci*vr1_rth;
13 vv1_eci = rth_eci*vv1_rth;
14
15 vr1_peri = r1*[cosd(true_a1) sind(true_a1) 0]';
16 vv1_peri = [-h/p*sind(true_a1) h/p*(e+cosd(true_a1)) 0]';
```

2 Time To Send The Message

You need to send the message by the time $E = 15^\circ$. How long do you have until t_2 ?
Time until $E = 15^\circ$: $t = 4.5618 \cdot 10^4 \text{ sec} = 12.6716 \text{ hours} = 0.528 \text{ days}$

```
1 % 2. You need to send the message by the time E=15deg. How long do you
  have
2 % until t2?
3 period = 2*pi*sqrt(a^3/MU);
4 E2 = 15; %deg
5 E2_rad = E2*pi/180;
6 E1 = 2*atand(tand(true_a1/2)/sqrt((1+e)/(1-e)));
7 E1_rad = E1*pi/180;
8
```

```
9 dt = sqrt(a^3/MU)*(E2_rad - E1_rad - (e*sin(E2_rad) - e*sin(E1_rad)));
```

3 New Position And Velocity

What is the new position and velocity (\vec{r}_2, \vec{v}_2) in ECI and perifocal coordinates?

$$\begin{aligned}\vec{r}_2 &= (4.0746\hat{x} - 0.7798\hat{y} - 0.4613\hat{z}) \cdot 10^4 \text{ km} = (4.0289\hat{e} + 1.0914\hat{p}) \cdot 10^4 \text{ km} \\ \vec{v}_2 &= 0.6526\hat{x} + 2.9573\hat{y} + 0.6892\hat{z} \text{ km/s} = -0.8039\hat{e} + 3.0000\hat{p} \text{ km/s}\end{aligned}$$

```
1 % 3. What is the new position and velocity in ECI and perifocal
  coordinates?
2 r2 = a*(1-e*cos(E2_rad));
3 true_a2 = 2*atan2d(sqrt((1+e)/(1-e))*tan(E2_rad/2),1);
4 vr2_rth = [r2 0 0]';
5 vv2_rth = [h*e/p*sind(true_a2) h/r2 0]';
6
7 AOL2 = AOP + true_a2;
8 rth_eci = rot_rth_eci(RAAN,inc,AOL2);
9
10 vr2_eci = rth_eci*vr2_rth;
11 vv2_eci = rth_eci*vv2_rth;
12
13 vr2_peri = r2*[cosd(true_a2) sind(true_a2) 0]';
14 vv2_peri = [-h/p*sind(true_a2) h/p*(e*cosd(true_a2)) 0]';
```

4 Latitude And Longitude

Find the latitude and longitude at the two positions.

$$\begin{aligned}\phi_1 &= 8.6698^\circ & \lambda_1 &= -49.9545^\circ \\ \phi_2 &= -6.3452^\circ & \lambda_2 &= -49.7477^\circ\end{aligned}$$

```
1 % 4. Find the latitude and longitude at the two positions
2 dt_days = dt/60/60/24;
3 rot_earth = 7.2921151467*10^-5; %rad
4 tu1 = juliandate(datetime('2019-03-06 03:00:00')) - 2451545;
5 tu2 = juliandate(datetime('2019-03-06 03:00:00')) - 2451545;
6
7 theta_era1 = 2*pi*(0.7790572732640 + 1.00273781191135448*tu1);
8 theta_era1 = mod(theta_era1,2*pi)*180/pi;
9 theta_era2 = 2*pi*(0.7790572732640 + 1.00273781191135448*tu2);
10 theta_era2 = mod(theta_era2,2*pi)*180/pi;
```

```

11
12 eci_ecef1 = rot_eci_ecef(theta_era1);
13 eci_ecef2 = rot_eci_ecef(theta_era2);
14 vr1_ecef = eci_ecef1*vr1_eci;
15 vr2_ecef = eci_ecef2*vr2_eci;
16
17
18 alpha1 = atan2d(cosd(inc)*sind(AOL1),cosd(AOL1));
19 alpha2 = atan2d(cosd(inc)*sind(AOL2),cosd(AOL2));
20 theta_gr1 = theta_era1+180/pi*rot_earth*0;
21 theta_gr2 = theta_era2+180/pi*rot_earth*dt;
22
23 lat1 = asind(vr1_ecef(3)/r1);
24 long1 = alpha1 + RAAN - theta_gr1;
25 lat2 = asind(vr2_ecef(3)/r2);
26 long2 = alpha2 + RAAN - theta_gr2;
27 long2 = mod(long2,-360);

```

5 Ground Track

Create a ground track between the two position using MATLAB. Mark the starting location.

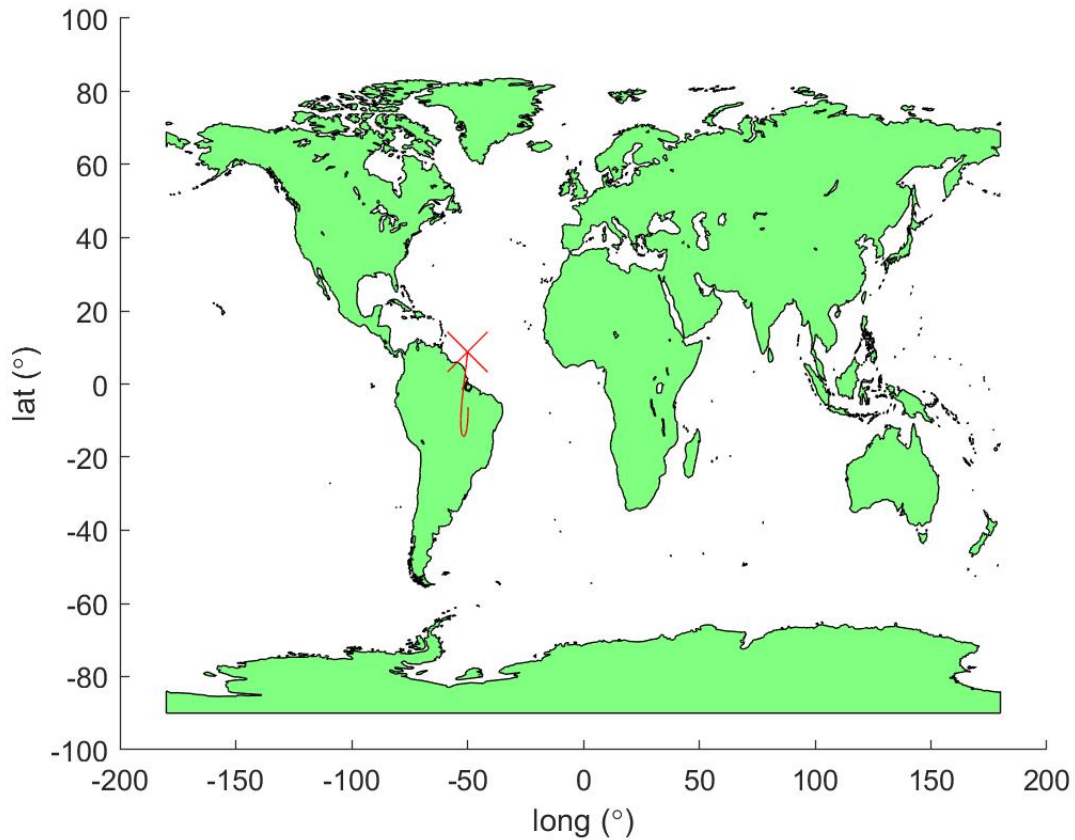


Figure 1: Latitude vs. Longitude

```
1 % 5. Create a ground track between the two positions using MATLAB. Mark
  the
2 % starting location.
3
4 true_a_array = linspace(true_a1, true_a2+360, 1000);
5 r_array = p./(1+e*cosd(true_a_array));
6 % E_array = acosd(-(r_array/a-1)/e).*sign(true_a_array);
7 E_array = 2*atand(tand(true_a_array/2)/sqrt((1+e)/(1-e)));
8 E_array_rad = E_array*pi/180;
9
10 dt_array = sqrt(a^3/MU)*(E_array_rad - E1_rad - (e*sin(E_array_rad) - e*
  sin(E1_rad)));
11
```

```

12 theta_era = theta_era1;
13 theta_gr_array = theta_era+180/pi*rot_earth*dt_array;
14
15
16 theta_array = AOP + true_a_array;
17 lat_array = asind(sind(inc)*sind(theta_array));
18
19 alpha_array = atan2d(cosd(inc)*sind(theta_array),cosd(theta_array));
20 long_array = alpha_array + RAAN - theta_gr_array;
21
22 figure; hold on;
23 geoshow("landareas.shp","FaceColor",[0.5 1.0 0.5]);
24 geoshow(lat_array, long_array,'Color','red');
25 geoshow(lat_array(1),long_array(1),'DisplayType','Point','Marker','x','
    Markersize',20);

```

6 Can You Send The Message?

Assume that you have a ground station in Prescott (34.54 N, 112.4685 W, 1.64 km above mean equatorial). Plot the elevation angle versus time passed. If the ground station needs a minimum elevation angle of 10 degrees, how much time is the TDRS visible? Can you send your urgent message to the TDRS?

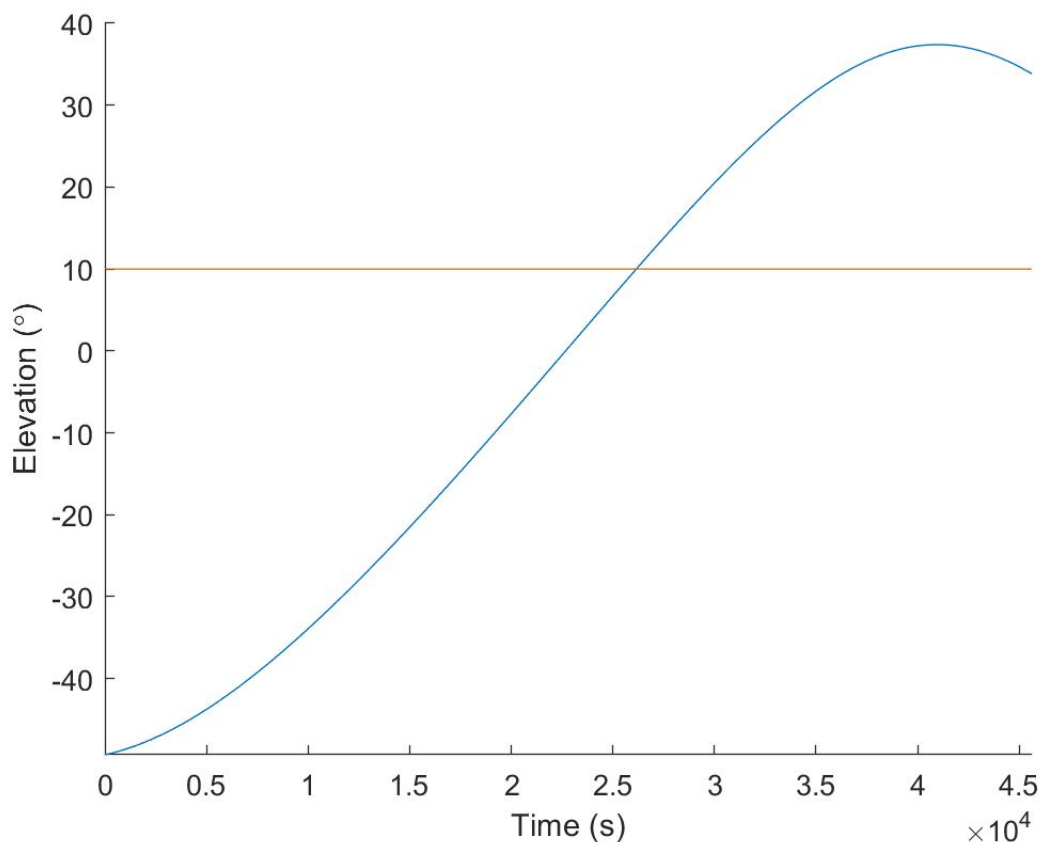


Figure 2: Elevation vs. Time

In Part 2, "Time To Send The Message", we were informed that we have until $E = 15$ degrees to send our message. We found that it will take 12.6716 hours to reach our $E = 15$ degrees limit, beginning on March 6, 2019 at 03:00:00.000. As shown in the above plot, it will take only 7.2751 hours for TDRS to become visible to our ground station. Once visible, the TDRS will remain visible for 5.3965 hours. Therefore, because TDRS becomes and remains visible for well within our allowable range, we will be able to send our urgent message as early as 10:16:30.36.

-
- 1 % 6. Assume you have the ground station in Prescott. Plot the elevation angle
 - 2 % versus time passed. If the ground station needs a minimum elevation angle


```

3 % of 10 degrees, how much time is the TDRS visible? Can you send your
4 % urgent message to the TDRS?
5 min_elevation = 10; %deg
6 radius_e = 6378; %km
7 lat_gs = 34.54; %deg
8 long_gs = 112.4685; %deg
9 h_gs = 1.64; %km
10 vr_e_gs = [0 0 radius_e+h_gs]'; %km
11
12 elevation_array = zeros(1,length(r_array));
13 for i = 1:length(r_array)
14     vr_rth = [r_array(i) 0 0]';
15     rth_eci = rot_rth_eci(RAAN, inc, theta_array(i));
16     eci_ecef = rot_eci_ecef(theta_era);
17     vr_ecef = eci_ecef*rth_eci*vr_rth;
18
19     ecef_sez = rot_ecef_sez(long_gs, lat_gs);
20     vr_sez = ecef_sez*vr_ecef;
21
22     vr_e_sc = vr_sez;
23
24     vr_gs_sc = vr_e_sc - vr_e_gs;
25
26     elevation_array(i) = asind(vr_gs_sc(3)/norm(vr_gs_sc));
27 end
28
29 figure(2)
30 hold on;
31 plot(dt_array, elevation_array)
32 line([0 dt_array(end)], [10 10], 'Color', [0.78 0.43 0.02])
33 xlabel('Time (s)')
34 ylabel('Elevation (\circ)')
35 q = 1;
36 while elevation_array(q)<10
37     q = q + 1;
38 end
39 time_start = dt_array(q);
40 time_end = dt_array(end);
41 time_above_minimum = time_end - time_start;
42 time_above_minimum_hours = time_above_minimum/60/60;

```

7 Why?

Why would the TDRS be in such an orbit?

The TDRS is a relay satellite intended to be in a geosynchronous orbit. The purpose of such an orbit is to maximize the amount of time that signals can be transmitted through TDRS from a well positioned ground station.

```
1 %% AE 313 Group Exam 2
2 % Team 3 – Jiyoung Hwang, Grace Day, Thorne Wolfenbarger, Aaron Scott
3
4 clc; clear all; close all;
5
6 % Knowns
7 MU = 398600;
8 a = 42170; %km
9 e = 0.01053;
10 RAAN = 14.89; %deg
11 AOP = 318.4; %deg
12 inc = 14.37; %deg
13 true_a1 = 184.2; %deg
14 energy = -MU/(2*a);
15
16 % 1. Find the current position and velocity in ECI and perifocal
    coordinates
17 p = a*(1-e^2);
18 h = sqrt(MU*p);
19 r1 = p/(1+e*cosd(true_a1));
20
21 vr1_rth = [r1 0 0]';
22 vv1_rth = [h*e/p*sind(true_a1) h/r1 0]';
23
24 AOL1 = AOP + true_a1;
25 rth_eci = rot_rth_eci(RAAN,inc,AOL1);
26
27 vr1_eci = rth_eci*vr1_rth;
28 vv1_eci = rth_eci*vv1_rth;
29
30 vr1_peri = r1*[cosd(true_a1) sind(true_a1) 0]';
31 vv1_peri = [-h/p*sind(true_a1) h/p*(e+cosd(true_a1)) 0]';
32 % 2. You need to send the message by the time E=15deg. How long do you
    have
33 % until t2?
34 period = 2*pi*sqrt(a^3/MU);
35 E2 = 15; %deg
36 E2_rad = E2*pi/180;
37
38 % syms E1
39 % eq = r1 == a*(1-e*cosd(E1));
40 % E1 = double(solve(eq,E1))
41 E1 = 2*atand(tand(true_a1/2)/sqrt((1+e)/(1-e)));
```

```

42 % E1 = acosd((a*e+r1*cosd(true_a1))/a);
43 % E1 = acosd(-(r1/a-1)/e)*sign(true_a1);
44 E1_rad = E1*pi/180;
45
46 dt = sqrt(a^3/MU)*(E2_rad - E1_rad - (e*sin(E2_rad) - e*sin(E1_rad)));
47
48 % 3. What is the new position and velocity in ECI and perifocal
    coordinates?
49 r2 = a*(1-e*cos(E2_rad));
50 true_a2 = 2*atan2d(sqrt((1+e)/(1-e))*tan(E2_rad/2),1);
51
52 % f = 1 - a/r1*cos(E2_rad - E1_rad);
53 % g = dt - sqrt(a^3/MU)*((E2_rad - E1_rad) - sin(E2_rad - E1_rad));
54 % fdot = sqrt(MU*a)/(r2*r1)*sin(E2_rad - E1_rad);
55 % gdot = 1-a/r2*(1-cos(E2_rad - E1_rad));
56 % vr2_eci = f*vr1_eci + g*vv1_eci;
57 % vv2_eci = fdot*vr1_eci + gdot*vv1_eci;
58 vr2_rth = [r2 0 0]';
59 vv2_rth = [h*e/p*sind(true_a2) h/r2 0]';
60
61 AOL2 = AOP + true_a2;
62 rth_eci = rot_rth_eci(RAAN,inc,AOL2);
63
64 vr2_eci = rth_eci*vr2_rth;
65 vv2_eci = rth_eci*vv2_rth;
66
67 vr2_peri = r2*[cosd(true_a2) sind(true_a2) 0]';
68 vv2_peri = [-h/p*sind(true_a2) h/p*(e+cosd(true_a2)) 0]';
69
70
71 % 4. Find the latitude and longitude at the two positions
72 dt_days = dt/60/60/24;
73 rot_earth = 7.2921151467*10^-5; %rad
74 tu1 = juliandate(datetime('2019-03-06 03:00:00')) - 2451545;
75 tu2 = juliandate(datetime('2019-03-06 03:00:00')) - 2451545;
76
77 theta_era1 = 2*pi*(0.7790572732640 + 1.00273781191135448*tu1);
78 theta_era1 = mod(theta_era1,2*pi)*180/pi;
79 theta_era2 = 2*pi*(0.7790572732640 + 1.00273781191135448*tu2);
80 theta_era2 = mod(theta_era2,2*pi)*180/pi;
81
82 eci_ecef1 = rot_eci_ecef(theta_era1);
83 eci_ecef2 = rot_eci_ecef(theta_era2);
84 vr1_ecef = eci_ecef1*vr1_eci;

```

```

85 vr2_ecef = eci_ecef2*vr2_eci;
86
87
88 alpha1 = atan2d(cosd(inc)*sind(AOL1),cosd(AOL1));
89 alpha2 = atan2d(cosd(inc)*sind(AOL2),cosd(AOL2));
90 theta_gr1 = theta_era1+180/pi*rot_earth*0;
91 theta_gr2 = theta_era2+180/pi*rot_earth*dt;
92
93 lat1 = asind(vr1_ecef(3)/r1);
94 long1 = alpha1 + RAAN - theta_gr1;
95 lat2 = asind(vr2_ecef(3)/r2);
96 long2 = alpha2 + RAAN - theta_gr2;
97 long2 = mod(long2,-360);
98
99 % 5. Create a ground track between the two positions using MATLAB. Mark
    the
100 % starting location.
101
102 true_a_array = linspace(true_a1, true_a2+360, 1000);
103 r_array = p./(1+e*cosd(true_a_array));
104 % E_array = acosd(-(r_array/a-1)/e).*sign(true_a_array);
105 E_array = 2*atand(tand(true_a_array/2)/sqrt((1+e)/(1-e)));
106 E_array_rad = E_array*pi/180;
107
108 dt_array = sqrt(a^3/MU)*(E_array_rad - E1_rad - (e*sin(E_array_rad) - e*
    sin(E1_rad)));
109
110 theta_era = theta_era1;
111 theta_gr_array = theta_era+180/pi*rot_earth*dt_array;
112
113
114 theta_array = AOL1 + true_a_array;
115 lat_array = asind(sind(inc)*sind(theta_array));
116
117 alpha_array = atan2d(cosd(inc)*sind(theta_array),cosd(theta_array));
118 long_array = alpha_array + RAAN - theta_gr_array;
119
120 figure; hold on;
121 geoshow("landareas.shp","FaceColor",[0.5 1.0 0.5]);
122 geoshow(lat_array, long_array,'Color','red');
123 geoshow(lat_array(1),long_array(1),'DisplayType','Point','Marker','x','
    Markersize',20);
124
125

```

```

126 % 6. Assume you have the ground station in Prescott. Plot the elevation
    angle
127 % versus time passed. If the ground station needs a minimum elevation
    angle
128 % of 10 degrees, how much time is the TDRS visible? Can you send your
129 % urgent message to the TDRS?
130 min_elevation = 10; %deg
131 radius_e = 6378; %km
132 lat_gs = 34.54; %deg
133 long_gs = 112.4685; %deg
134 h_gs = 1.64; %km
135 vr_e_gs = [0 0 radius_e+h_gs]'; %km
136
137 elevation_array = zeros(1,length(r_array));
138 for i = 1:length(r_array)
139     vr_rth = [r_array(i) 0 0]';
140     rth_eci = rot_rth_eci(RAAN, inc, theta_array(i));
141     eci_ecef = rot_eci_ecef(theta_era);
142     vr_ecef = eci_ecef*rth_eci*vr_rth;
143
144     ecef_sez = rot_ecef_sez(long_gs, lat_gs);
145     vr_sez = ecef_sez*vr_ecef;
146
147     vr_e_sc = vr_sez;
148
149     vr_gs_sc = vr_e_sc - vr_e_gs;
150
151     elevation_array(i) = asind(vr_gs_sc(3)/norm(vr_gs_sc));
152 end
153
154 figure(2)
155 hold on;
156 plot(dt_array, elevation_array)
157 line([0 dt_array(end)], [10 10], 'Color',[0.78 0.43 0.02])
158 xlabel('Time (s)')
159 ylabel('Elevation (\circ)')
160 q = 1;
161 while elevation_array(q)<10
162     q = q + 1;
163 end
164 time_start = dt_array(q);
165 time_end = dt_array(end);
166 time_above_minimum = time_end - time_start;
167 time_above_minimum_hours = time_above_minimum/60/60;

```

```

168 % 7. Why would the TDRS be in such an orbit?
169
170
171
172 function A = rot_rth_eci(o,i,t)
173
174 % o : Omega, Longitude of the Ascending Node (RAAN)
175 % i : i, inclination
176 % t : theta, Argument of Latitude
177
178 A = [cosd(o)*cosd(t)-sind(o)*cosd(i)*sind(t), -cosd(o)*sind(t)-sind(o)*
      cosd(i)*cosd(t), sind(o)*sind(i); ...
179      sind(o)*cosd(t)+cosd(o)*cosd(i)*sind(t), -sind(o)*sind(t)+cosd(o)*
      cosd(i)*cosd(t), -cosd(o)*sind(i); ...
180      sind(i)*sind(t), sind(i)*cosd(t), cosd(i)];
181
182 end
183
184 function A = rot_eci_ecef(t)
185
186 % t : theta_era, Earth rotation angle
187
188 A = [cosd(t), sind(t), 0; -sind(t), cosd(t), 0; 0, 0, 1];
189 end
190
191 function A = rot_ecef_sez(l,p)
192
193 % l : lamda_gs, longitude of the ground station
194 % p : phi_gs, latitude of the ground station
195
196 A = [sind(p)*cosd(l), sind(p)*sind(l), -cosd(p);
197      -sind(l), cosd(l), 0;
198      cosd(p)*cosd(l), cosd(p)*sind(l), sind(p)];
199
200 end

```
