AE4878 - Mission Geometry and Orbit Design
Part 10
Disaster management via Earth observation mission for Indonesia

Ali Nawaz
3ME, LR, Delft University of Technology.

September 2, 2018

1 Introduction

Indonesia lies on one of the most geologically active plates in the world. A local newspaper published in 2012 indicated that Indonesia has 127 active volcanoes with about 5 million people living within the danger zones[1]. Due to the geographic location of Indonesia, western islands of Indonesia are guarded by chains of volcanoes formed by the Sunda mega-thrust subduction zone. This is shown with the aid of Figures 1[2] and 2[3].

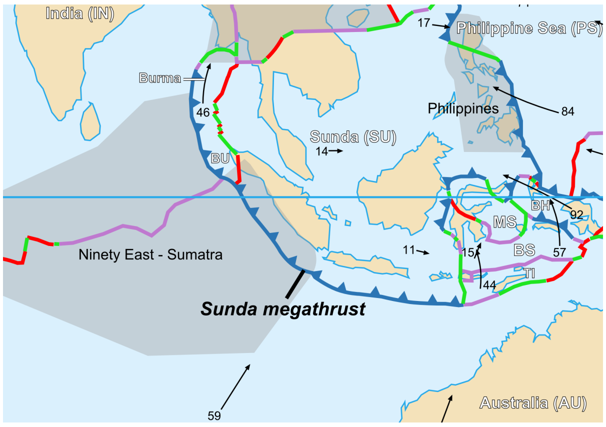

Figure 1: Sunda megathrust subduction zone on the western coastline of Indonesia.

Figure 2: Visual guide to some of the major volcanoes in Indonesia.

Since Indonesia lies close to the Equator, it is also covered by clouds majority of the time. Thus passive images taken from remote sensing satellites are not quite functional for immediate disaster management support. Due to high volcanic and plate tectonic activities in the region, Indonesia is exposed to earthquakes, Tsunami, floods, forest fires, landslides and typhoons. The damage done by all these natural disasters can be tracked via a Synthetic Aperture Radar (SAR) in orbit. SAR is an active imaging system it conduct 2D/3D topography observation both in night and cloudy environment. Given the geo-locational constraints of Indonesia and the environmental disasters under consideration, SAR makes the perfect instrument for disaster management instrument from space [4].

[1] Indonesia's 127 volcanoes: `http://www.pikiran-rakyat.com/node/186891`
[2] Sunda mega thrust: `https://en.wikipedia.org/wiki/File:Plate_setting_Sunda_megathrust.png`
[3] Indonesia Volcano map: `https://en.wikipedia.org/wiki/File:Map_indonesia_volcanoes.gif`
[4] SAR Applications in disaster management: `http://sac.gov.in/nisar/NISAR%20Science%20Workshop_Presentations_2015/P1-05.pdf`

Mission objective

The objective of this assignment is to select appropriate instrument or satellite solution and design an Earth observation mission to provide disaster management tools for Indonesia, for a period of 5 years. An assumption on payload mass of 300 kg is expected. However, after a literature study on the available SAR instruments and satellite solutions, this assumed value is updated. The assignment only addresses the orbital aspects such as orbital parameters, number of satellites, launch, maintenance, end-of-life relation with target area(s) etc.

Approach

A top down design approach is undertaken to achieve the desired final orbit solution. This is outlined with the aid of of the flow chart outlined in Figure 3. The approach is top down in the sense that first an instrument/spacecraft solution is determined, before designing the orbit and obtaining the ΔV budget. Once the ΔV budget is determined a launcher is selected. However, this is a iterative process, the location of the launch site plays a major role in the ΔV budgeting. ΔV budget might not necessarily indicate a launch cost, due to the recent developments in alternative low cost launches. Similar ΔV budget from two different launch site might result in different launch cost. This might be due to the different launchers accessible at different launch sites and their pricing options.

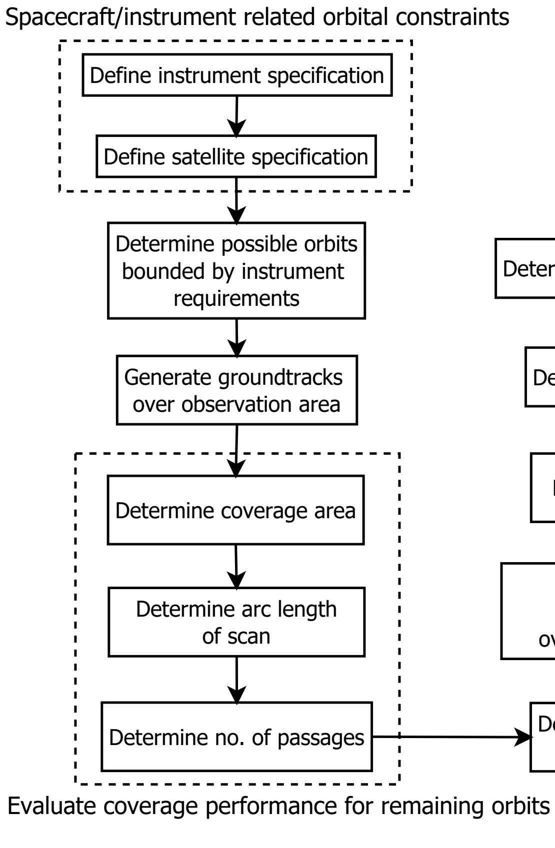

Figure 3: Brief overview of the approach undertaken to provide disaster management mission design solution.

Section 2 outlines the instrument and spacecraft solution chosen for the disaster management mission. This is followed by the sub-optimal orbit design and discussion on the obtained results. Section 3 provides an insight into the ΔV budget and launcher selection process. To conclude this assignment, the scripts are presented in Section 4

2 Mission design

This section first outlines the choice of payload and possible spacecraft solutions. Judging the spacecraft and payload operational constraints orbits are designed. Once the orbit is designed, the result of the optimum orbit design is presented.

Spacecraft and payload definition

Though it is recommended to select a payload mass of 300kg, some better alternatives were found during the literature study phase. Recent developments in SAR technology have lead to miniaturization and low cost SAR capabilities. Since the scope of this assignment is limited to orbit design, a SAR instrument is not designed/sized from scratch. However, Commercial Off The Shelf (COTS) solutions are sought after. Interested readers can refer to [5] for details into the sizing of SAR instrument. Table 1 refers to some of the recent developments in SAR satellite and instrumentation [7] [1].

Table 1: Overview of SAR satellite solutions.

Satellite solution	**Wet Mass [kg]**	**Resolution [m]**
NovaSAR-S	400	6
TecSAR	300	1
TerraSAR-X	1230	1
RadarSat-2	2200	3
MicroXSAR (proposal at JAXA based on SDS-4)	100	3 or 10
ICEYE	50-70	6 - 10 (aim for 3m by 2020)

Analysing Table 1, new space solution seems to offer better resolution with lower satellite mass. The solution chosen for this assignment is ICEYE satellite. Since the concept has been proven with ICEYE X1 being launched into space. The company aims to launch 18 satellites in distributed orbital planes by 2020 with average interval between imaging down to 2 hours [1]. Payload specifications required for orbit design are outlined in Table 2 [1] [2] [4].

Table 2: ICEYE satellite and instrument specification

Specification category	**Value**	**Unit**
Swath width	30-100	km
Ground resolution	6 - 10	m
Operational altitude	500-600	km
Lifetime	2 - 3	Earth years
FOV (based on the above swath width and operational altitude)	2.9-11.4	deg
FOV for 60km swath	5.7248 - 6.8673	deg

To estimate the FOV, the simplified trigonometric approach outlined in Equation 1 is used.

$$FOV = 2 \cdot \arctan\left(\frac{\text{swath}}{2 \cdot \text{altitude}}\right) \tag{1}$$

Orbit design

The steps undertaken to generate the optimum orbit design is outlined with the aid of flowchart in Figure 4. First the search space is defined with the aid of range selection for i, j and k parameter. Before selecting the ranges of orbital inclination angles, extreme boundary points of Indonesia must be defined. The extreme points of Indonesia is demonstrated with the aid of Table 3[5]. Judging the extreme points, the inclination of the orbit should be greater than or at least equal to 11°0'27". The range of inclination for search region is thus chosen as [11°0'27", 2·11°0'27"]. j is chosen in the range [10:1:100] and k is chosen in the range [1:1:7].

[5]Indonesia extreme points: `https://en.wikipedia.org/wiki/Extreme_points_of_Indonesia`

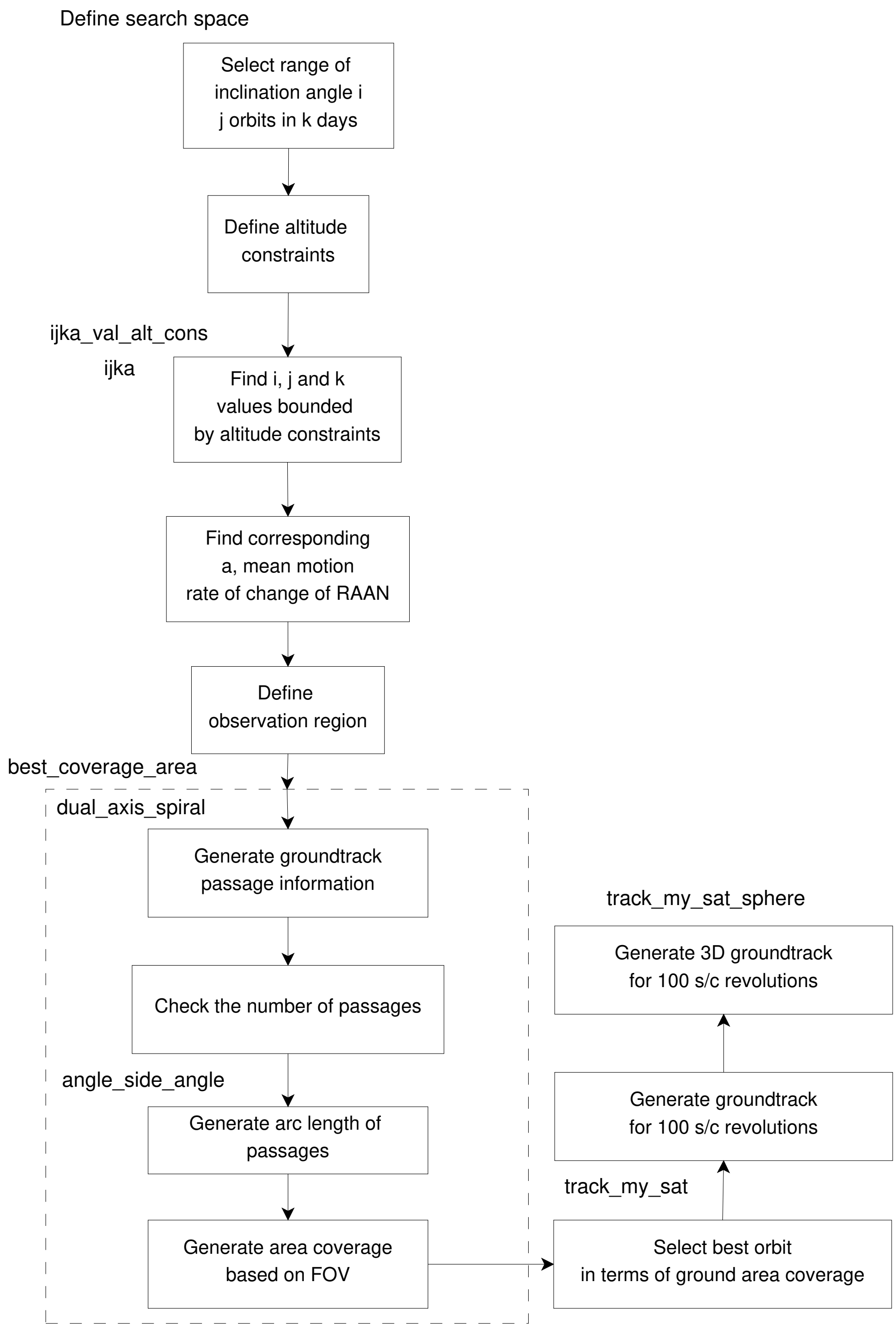

Figure 4: Outline of the approach to select the best orbit and the underlying functions used.

Table 3: Extreme points of Indonesia.

	Western-most point	Eastern-most point	Northern-most point	Southern-most point
Angle [deg]	94°58'22" East	141°1'10" East	6°4'30" North	11°0'27" South

Most of the Earth observation satellites orbit at an altitude of 500km. The SAR satellites analysed during the literature study had an altitude range of 400 to 700km. The ICEYE test satellites were functional within this given range. Thus an altitude range of [400km,700km] is chosen as the boundary constraint, for appropriate i,j and k parameter selection. The function used to estimate semi-major axis a , orbital altitude h, satellite mean motion n and rate of change of Right Ascension of Ascending Node (RAAN) is "`ijka(i,j,k)`". The function was verified in the previous assignments. A script outlining the function is presented in Section 4. The function "`ijka_val_alt_cons(i,j,k,alt_min, alt_max)`" uses the maximum and minimum altitude range and "`ijka(i,j,k)`" to constraint the choice of i,j and k.

Once the orbit options bounded by altitude constraints are available, the "`best_coverage_area(i,j,k,a,raandot,` mean_motion, e,rev, rho2, phi1_0,phi2_0,dt,east1_deg,east2_deg,north_deg,south_deg,fov)" function is used to select the orbit resulting in the best coverage area. The function "best_coverage_area()" uses the dual axis spiral function "dual_axis_spiral(rho1, rho2,omega1,omega2, phi1_0, phi2_0, phi1, phi2, t)" and the Angle-Side-Angle algorithm. Both of which are previously verified in earlier assignments. e is the eccentricity of the orbit, which is assumed to be 0 (a circular orbit). Parameter rev selects the number of spacecraft revolutions to be simulated. A value of 100 is chosen for rev. Parameter ρ_1 (angular distance of S from C) is taken as the inclination angle of the orbit and ρ_2(angular distance of P from S) is taken as 90 degrees to be able to produce the necessary ground track. Parameters ϕ_{1_0} (initial azimuth S about C) is taken as 90 degrees and ϕ_{2_0} (initial azimuth P about S) is taken as 0 degrees. The time step for simulation dt, is taken as 1 second. And the FOV is taken as the conservative estimate of 5.72 degrees as outlined in Table 2. For further information regarding the parameters, the reader is referred to [8][Table 8-8, pg.403]. For generating the arc length of passage a and right angled spherical triangle is used. Where, the difference in longitude of passage over the observation region represents the base while the difference in latitude represents the height of the circular triangle. The arc length of passage is given by the hypotenuse of the spherical triangle.This arc length is estimated by the Angle-Side-Angle rule. A quick sketch of the spherical triangle is presented with the aid of Figure 5. Once the arc length of ground passage is determined, the length of passage can be estimated with simple trigonometry relation $R_{Earth} \cdot \theta$. Where, θ represent the estimated arc length side. Length of passage multiplied with the ground swath width gives the scanned ground area. Assumption is made that push-broom scanning technique is used.

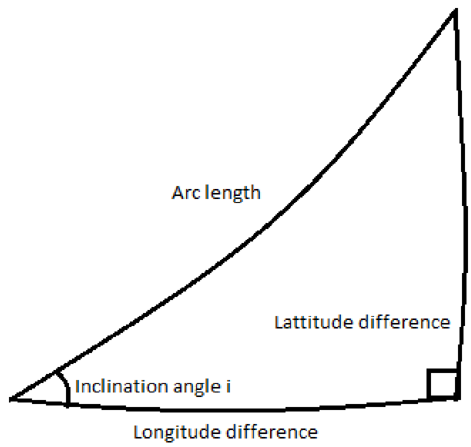

Figure 5: Quick sketch outlining the Angle Side Angle strategy to estimate the arc length of ground passage.

The orbital parameters are selected based on the maximum ground area scanned. Visualisations are generated with the aid of functions "trackmysat(i,j,k,a,raandot,mean_motion,e,rev,rho1,rho2,phi1_0,phi2_0,dt)" and "trackmysat_3D(i,j,k,a,raandot,mean_motion,e,rev,rho1,rho2,phi1_0,phi2_0,dt)". The above mentioned function and the main run file are outlined in Section 4.

Results

The resulting orbital parameters are outlined with the aid of Table 4.

Table 4: Resulting performance of orbital parameters for altitude bound [400km to 700 km].

Parameter	**Value**	**Unit**
inclination angle	11.0075	[deg]
j	43 or 86	orbits
k	3 or 6	days
semi major axis, a	7075.022067	km
orbital altitude, h	696.8861	km
rate of change of RAAN $\dot{\Omega}$	-7.874783e-05	deg/s
Satellite mean motion n	0.060785	deg/s
eccentricity, e	0	-
no. of satellite revolutions analysed	100	-
ground swath	139.608336	km
total no. of passages	76	-
total scanned length	342062.984228	km
total area scanned	47754844.066586	km^2

Figure 6 outlines a 3D view of the zone under observation. While Figure 7 outlines a 3D view of the zone under observation with passage ground tracks after 100 satellite revolutions. Figure 8 outlines the ground track for simulated orbit used for orbit selection. While Figure 9 demonstrates a zoomed in view of Indonesia to demonstrate the orbit passage undertaken.

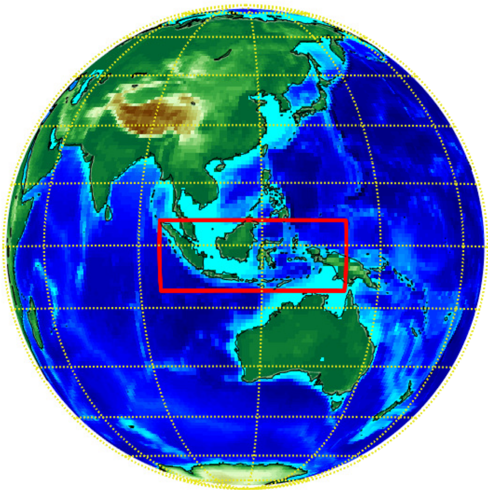

Figure 6: 3D overview of the region under observation.

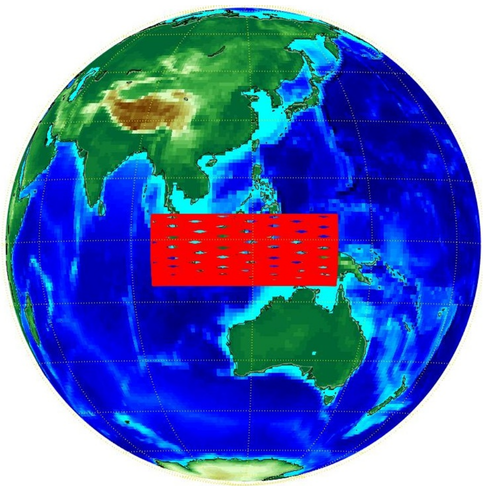

Figure 7: 3D overview of the region under observation, with passages after 100 satellite revolutions.

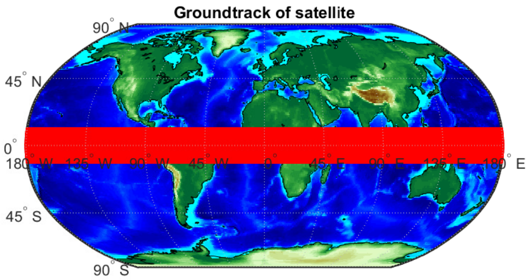

Figure 8: Global overview of the satellite ground track for 100 satellite revolutions.

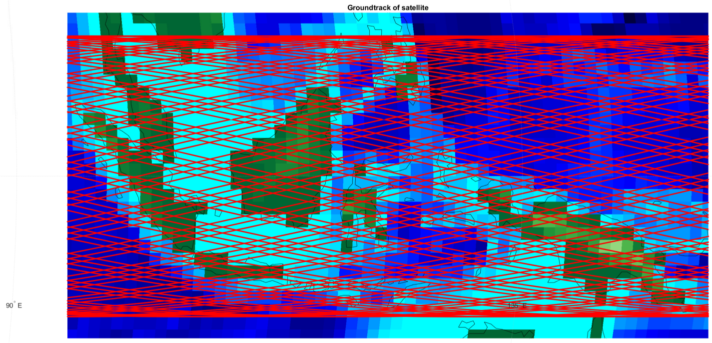

Figure 9: Overview of ground track over Indonesia for 100 satellite revolutions.

If the altitude bounds are reduced further, i.e. [500 km to 650 km], then the results outlined in Table 5.

Table 5: Resulting performance of orbital parameters for altitude bound [550km to 650 km].

Parameter	Value	Unit
inclination angle	11.0075	[deg]
j	29,58,87	orbits
k	2,4,6	days
semi major axis, a	7018.52370	km
orbital altitude, h	640.5237	km
rate of change of RAAN $\dot{\Omega}$	-8.098893e-05	deg/s
Satellite mean motion n	0.0615209	deg/s
eccentricity, e	0	-
no. of satellite revolutions analysed	100	-
ground swath	128.289926	km
total no. of passages	74	-
total scanned length	338669.317862	km
total area scanned	43447861.62016	km^2

Clearly a higher satellite orbit results in higher area coverage, but the swath width decreases. Since most of the SAR satellites in literature fly closer to 600 km altitude, the later option of 640.5237 km is chosen. However, it is important to note that this analysis completely excludes power budget, contact time and data transmission analysis. This is simply a first order estimate in terms of instrumental resolution and area coverage demands. The customer is strongly recommended to conduct further rounds of orbital design analysis, taking into account all the possible factors influencing orbit design. Figure 10 outlines a 3D view of the zone under observation with passage ground tracks after 100 satellite revolutions. Figure 11 outlines the ground track for simulated orbit used for orbit selection. While Figure 12 demonstrates a zoomed in view of Indonesia to demonstrate the orbit passage undertaken.

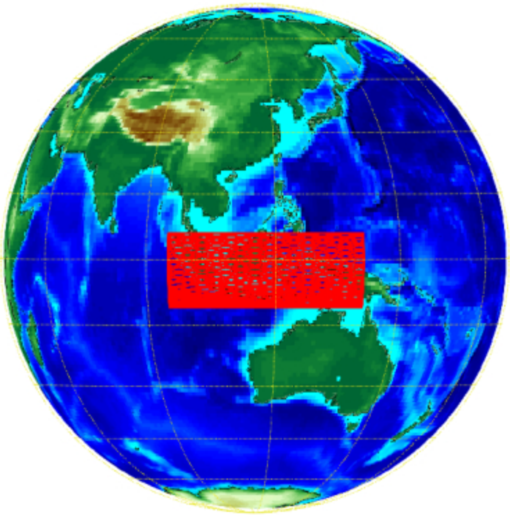

Figure 10: 3D overview of the region under observation, with passages after 100 satellite revolutions at an altitude of 640.5237 km.

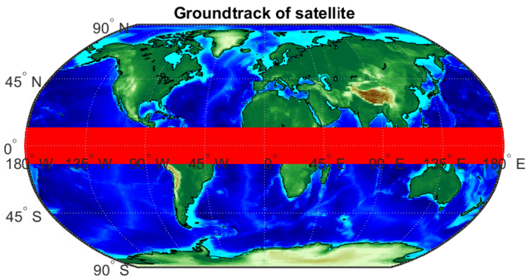

Figure 11: Global overview of the satellite ground track for 100 satellite revolutions and altitude of 640.5237 km.

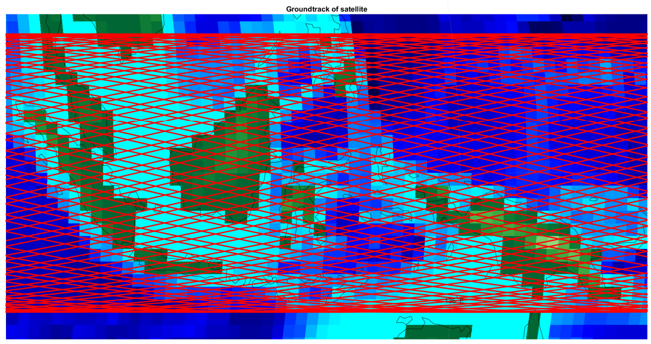

Figure 12: Overview of ground track over Indonesia for 100 satellite revolutions and altitude of 640.5237 km.

One might wonder what if the simulation time is set to the length of an Earth day, what would the results look like? The answer is presented with the aid of Table 6.

Table 6: Resulting performance of orbital parameters for altitude bound [550km to 650 km] for an Earth day.

Parameter	Value	Unit
inclination angle	11.0075	[deg]
j	29,58,87	orbits
k	2,4,6	days
semi major axis, a	7018.52370	km
orbital altitude, h	640.5237	km
rate of change of RAAN $\dot{\Omega}$	-8.098893e-05	deg/s
Satellite mean motion n	0.0615209	deg/s
eccentricity, e	0	-
time analysed	$24 \cdot 60 \cdot 60$	s
ground swath	128.289926	km
total no. of passages	10	-
total scanned length	40563.177698	km
total area scanned	5203847.052424	km^2

Every single day the satellite will make 10 passages over Indonesia. To provide a visual cue, the result of this is presented with the aid of Figures 13 and 14. Where, Figure 13 outlines a 3D overview of the passages over the extreme points of Indonesia. While Figure 14 shows a 2D ground track outlining the passages of the satellite track over Indonesia for one day.

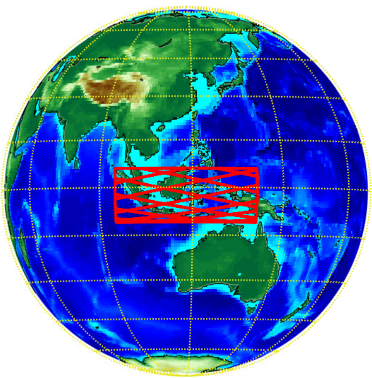

Figure 13: 3D overview of the passages over the extreme points of Indonesia for one Earth day.

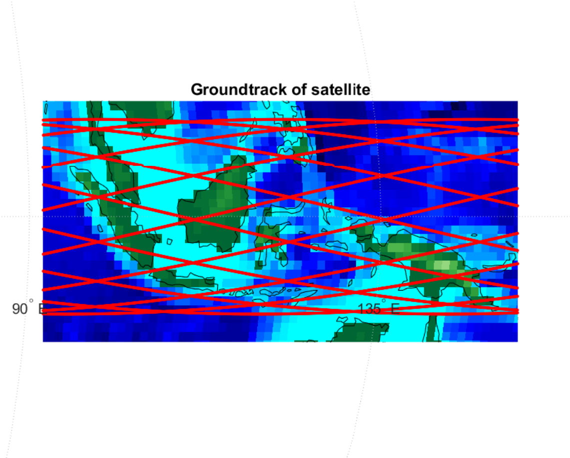

Figure 14: 2D overview of the passages over the extreme points of Indonesia for one Earth day.

3 Launcher selection and $\triangle$V budget

Since a top down approach is followed, where a satellite solution is presented first, the wet mass of the satellite is known as presented in Table 2. Since the satellite wet mass is below 100kg, most launchpads within or surrounding the latitude of observation posses launchers capable of launching the satellite into orbit. Section 3.1 conducts an analysis on ΔV required for orbit insertion for a chosen launch site. Section 3.2 conducts a brief analysis on the orbital decay behaviour of the satellite. Alongside orbital maintenance and EOL solutions are presented. The section concludes with total ΔV budget required for the specific choice of mission design. This chapter concludes with Section **??** where replenishment requirements are presented. Since launch cost is for piggyback options are not directly available for most of the launchers, only a back of of the envelope analysis of ΔV budget is presented here.

3.1 Orbit injection

As seen in Table 6, the orbital inclination for the satellite orbit is chosen as 11.0075 degrees. The objective is to first find a launch site close to a latitude of 11.0075 degrees North or South, in order to minimise the ΔV budget associated with orbital inclination change from transfer orbit to mission orbit. It is important to note that this analysis of cost is based on ΔV budget. Alternative options e.g. piggyback or re-usability of rocket launcher, will definitely effect the cost of launch for the customer. A detailed analysis is strongly recommended for final budgeting.
PSLV launcher is selected. PSLV-C37 successfully previously known to carry and deploy a record number of 104 satellites in sun-synchronous orbits. Furthermore previous PSLV bears previous records of CubeSat piggybacking options. PSLV makes a good choice because of it's launch site in Sriharikota, India. The launch site is located at 13.73740°N 80.23510°E [3].
Different schemes of orbit insertion exist. First scheme, involves inclination change at parking orbit followed by transfer orbit to the final orbit. Second scheme involves first conducting the transfer orbit and later conducting an inclination change. Second scheme is always recommended over the first scheme, since plane inclination is already expensive in terms of ΔV. Furthermore, an inclination change at a lower altitude requires higher ΔV than an inclination change at a higher altitude. Third scheme is to conduct plane change and transfer orbit initiation in one manoeuvre, followed by final orbit insertion. This scheme is more efficient than the first two[9][pg.151]. Fourth and final scheme requires makes use of an optimisation algorithm. The idea behind the fourth scheme is to have two transfer orbits before inserting into the final orbit. First transfer orbit sends the spacecraft beyond the actual orbital attitude with a combined ΔV and Δi manoeuvre, this is followed by

a second transfer orbit with combined ΔV and and Δi which brings the spacecraft to the desired orbit, this is followed by the final ΔV for orbit insertion.
Since it is known from literature [9][pg.151] that the first two schemes are inefficient. Third and fourth scheme will be the core of the analysis outlined in this Section.To provide the reader with a visual overview. Third scheme is presented with the aid of Figure 15 and the fourth scheme is presented with the aid of Figure 16.

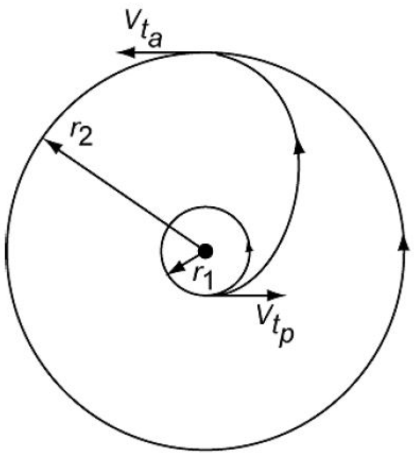

Figure 15: Third scheme of orbital transfer for ΔV budget estimation.

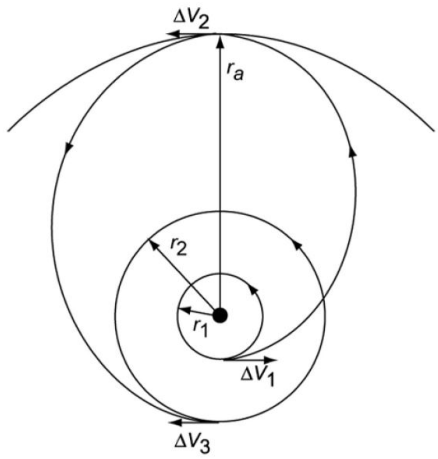

Figure 16: Fourth scheme of orbital transfer for ΔV budget estimation.

The underlying expression for the third scheme is presented in Equation 2. The manoeuvre comprises of two impulsive shots. As seen in Equation 2 the objective is to solve for two inclination plane change manoeuvre. Since the inclination of the parking and final orbit is known i.e. 13.73740° and 11.0075°, finding Δi_1 is the prime objective. Since this ultimately leads to finding Δi_2. Where, $\Delta i_2 = |i_{target} - i_{park}| - \Delta i_1$. A pre-verified Monte Carlo scheme is used to find the sub-optimal choice of Δi_1 and Δi_2. A copy of the script is presented in Section 4. Behaviour of ΔV for variable choices of Δi_1 and Δi_2 is presented with the aid of Figures 17 and 18. The Monte Carlo scheme is conducted for 3 random initialisation, with 1000 Δi_1 samples/initialisation. The lower bound for Δi_1 is set to 0°, while the upper bound is set to 13.73740° - 11.0075°.

$$\begin{aligned}
\Delta V &= \sqrt{V_{park}^2 + V_{trans,p}^2 - 2 \cdot V_{park} V_{trans,p} \cos\left(\Delta i_1\right)} + \sqrt{V_{final}^2 + V_{trans,a}^2 - 2 \cdot V_{final} V_{trans,a} \cos\left(|i_{target} - i_{park}| - \Delta i_1\right)} \\
a_{trans} &= \frac{a_{park} + a_{final}}{2} \\
V &= \sqrt{\mu_{Earth} \cdot \left(\frac{2}{r} - \frac{1}{a}\right)}
\end{aligned} \tag{2}$$

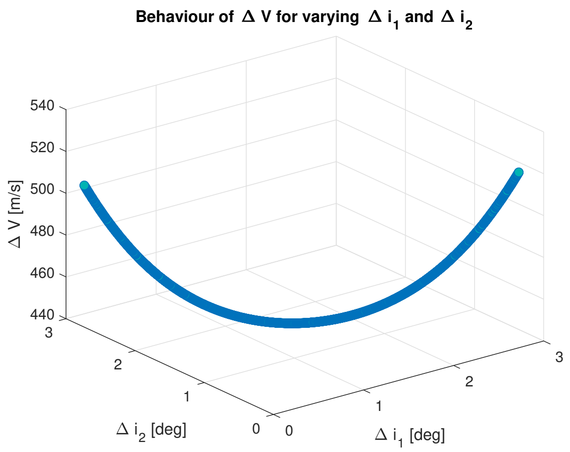

Figure 17: ΔV behaviour for different choice of Δi_1 and Δi_2. 3D view of Monte Carlo distribution.

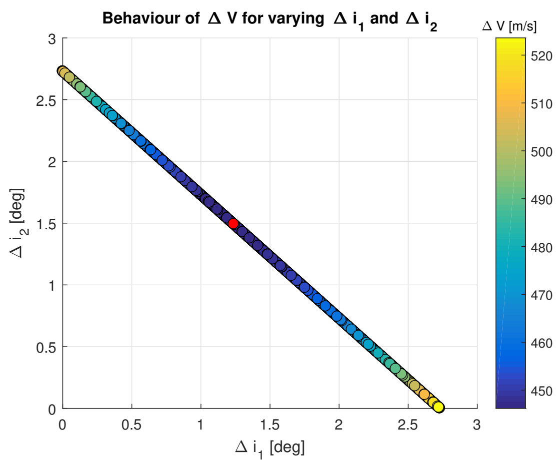

Figure 18: ΔV behaviour for different choice of Δi_1 and Δi_2. 2D view of Monte Carlo distribution with the minimum ΔV in red.

Similarly, the expressions for the fourth scheme is presented with the aid of Equation 3. The expressions represent a three impulsive shot manoeuvre with three unknowns. A pre-verified Monte Carlo scheme is used to find the sub-optimal choice of unknowns i.e. the intermediate semi major axis a_{int}, change in inclination angles Δi_1 and Δi_2, subsequently Δi_3. A copy of the script is presented in Section 4. Behaviour of ΔV for variable choices of a_{int}, Δi_1 and Δi_2 is presented with the aid of Figures 19 and 20. The Monte Carlo scheme is conducted for 3 random initialisation, with 1000 a_{int}, Δi_1 and Δi_2 samples/initialisation. The lower bound for Δi is set to 0°, while the upper bound is set to 13.73740° - 11.0075°. The lower bound for a_{int} is set to a_{final} and the upper bound to $2 \cdot a_{final}$.

$$
\begin{aligned}
\Delta V_1 &= \sqrt{V_{park}^2 + V_{trans,1,p}^2 - 2 \cdot V_{park} V_{trans,1,p} \cos\left(\Delta i_1\right)} \\
\Delta V_2 &= \sqrt{V_{trans,1,a}^2 + V_{trans,2,a}^2 - 2 \cdot V_{trans,1,a} V_{trans,2,a} \cos\left(\Delta i_2\right)} \\
\Delta V_3 &= \sqrt{V_{trans,2,p}^2 + V_{final}^2 - 2 \cdot V_{trans,2,p} V_{final} \cos\left(|i_{target} - i_{park}| - \Delta i_1 - \Delta i_2\right)} \\
\Delta V &= \Delta V_1 + \Delta V_2 + \Delta V_3 \\
a_{trans,1} &= \frac{a_{park} + a_{int}}{2} \text{ and } a_{trans,2} = \frac{a_{int} + a_{final}}{2} \\
V &= \sqrt{\mu_{Earth} \cdot \left(\frac{2}{r} - \frac{1}{a}\right)}
\end{aligned}
\tag{3}
$$

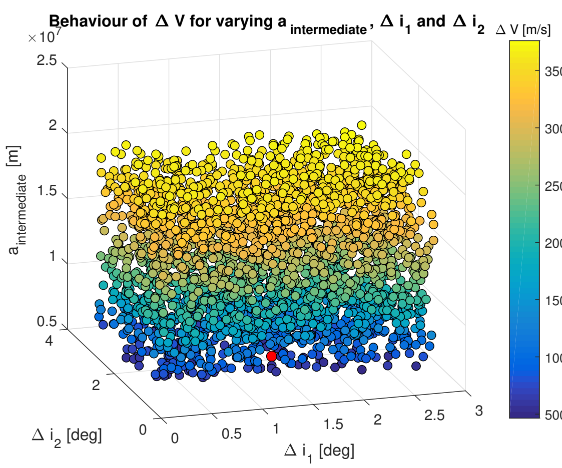

Figure 19: ΔV behaviour for different choice of a_{int}, Δi_1 and Δi_2. Red dot indicates the best choice.

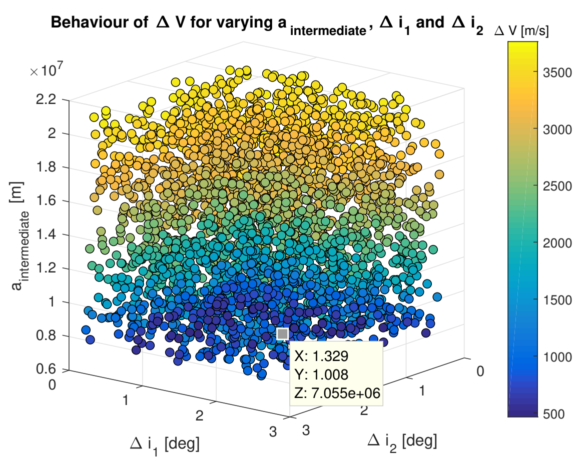

Figure 20: ΔV behaviour for different choice of a_{int}, Δi_1 and Δi_2. With minimum $DeltaV$ selected.

To conclude on the choice of orbit insertion manoeuvre from parking to final orbit, Table 7 outlines the outcomes of different orbital manoeuvres. For the case under observation, impulsive two shot manoeuvre is the best choice. Thus after the launcher inserts the satellite into a parking orbit at an altitude of 185km[6][Slide 51/131] and inclination equal to 13.73740°, ΔV of 446.1834 m/s is further required for insertion into nominal operational orbit. It is important to note that, this assumption of parking orbit characteristic is very likely to change depending on the launching criteria (e.g. piggy backing as a secondary payload.)

Table 7: Summary two and three impulsive shot, best choices of ΔV.

Transfer type	$a_{intermediate}$ [m]	Δi_1 [deg]	Δi_2 [deg]	Δi_3 [deg]	ΔV [m/s]
2 impulsive shot	-	1.2398	1.4901	-	446.1834
3 impulsive shot	7054835.8335	1.3286	1.0081	0.3932	464.7980

3.2 Orbital decay, maintenance, EOL behaviour and replenishment

Few other things that effect the ΔV budget include orbital decay, maintenance and EOL satellite behaviour. Due to the limited scope of this assignment, back of the envelope calculations are conducted to provide a quick

overview of estimations.
According to ICEYE satellite[1], the operational lifetime of the satellite is between 2 to 3 years. Judging the support requirement of 5 years, the customer is advised to plan for at least 2 replenishment missions. Worst case scenario, three mission replenishment might be required. Since the ICEYE satellites are in testing phase in orbit around Earth, more about accurate lifetime will be known soon. According to [8][pg.68] *"Above 600km the lifetimes due drag will typically exceed spacecraft operational lifetime of approximately 10 years, although drag may still be important for orbit maintenance or maintaining the structure of a constellation."*. Orbital decay in semi major axis after one orbital revolution can be expressed as shown in Equation 4[8][pg.69].

$$\Delta a_{2\pi} = -2\pi(C_D\frac{A}{m})\rho a^2$$

C_D= co-efficient of drag, A = S/C area, m = S/C mass, ρ = density at orbital altitude, a = semimajor axis. All in SI units. The density model is assumed to be simplified exponential.

$$\rho = \rho_0 exp(\frac{-hg_0}{RT})$$

Where, R = 287 J/kg/K, T = 240K, g_0 = 9.81 m/s^2, h = altitude [m], $\rho_0 = 1.225kg/m^{-3}$

(4)

Assuming 365 days/year with 86400s/day and no loss in orbital eccentricity, the total loss in semi-major axis can be estimated with the script provided in Section 4. According to [8] average C_D is 2.5. Area is taken as the largest cross sectional area of the satellite [1], 0.4· 3 = $1.2m^2$. The effect of solar radiation is not taken into account. And the exponential atmosphere model is no where close to perfection. It is used for a quick back of the envelope calculation purpose only. For mission duration of 2,3 and 5 years the total Δa degradation is -2.9527e-23m, -4.4291e-23m and -7.3818e-23m respectively. This does not degrade the mission performance. And hence no ΔV budget for orbital decay needs to be estimated. Altitude maintenance over 5 years is estimated to be 3.9630e-26 m/s. Which is negligible.
To de-orbit the satellite from $\approx$ 600km to a perigee of 50km, about 157m/s of ΔV is required[8][Table 2-21, pg.102]. Approximate lifetime after the manoeuvre is 0.033 Earth years.

To conclude this section after the satellite is inserted into the parking orbit, a two shot impulsive manoeuvre of ΔV = 446.1834 m/s is required to insert the satellite into nominal orbit. EOL de-orbit to 50km altitude requires a ΔV of 157m/s. Finally a safety factor of 1.2% is applied to account for unaccounted station keeping of the satellite. This gives a total ΔV budget of (446.1834 + 157)·1.2 $\approx$ 723.82008 m/s. The customer is recommended to conduct a detailed ΔV estimation. Since this analysis only provides a quick estimate.

In conclusion a satellite mission is designed with a new space approach. The objective is provide disaster management support to Indonesia for 5 years. SAR is chosen as the core payload. This is primarily to support volcanic acitvity related disaster management. But can be utilised for other disasters as well, e.g. flood, earthquake etc. ICEYE satellite is chosen, detailed S/C and SAR specifications can be found in Table 2. Orbital is chosen such that, every day the satellite makes 10 passages over Indonesia. More about orbital parameters can be found in Table 6. PSLV series of launcher is selected with launch site at Sriharikota, India. The launch site is located at 13.73740°N 80.23510°E. After orbit insertion a ΔV budget of 723.82008 m/s is estimated, to support orbit insertion, maintenance and EOL manoeuvre. The mission design requires 2-3 satellite replenishment. The customer is strongly recommended to conduct an in-depth analysis on the budgets presented, since the budgets presented are simplified estimates. The customer is provided with Matlab scripts in Section 4. For replication and verification of the mission design process.

4 Matlab Scipt

All files can be found at the following Git source:
`https://github.com/Alixir/Mission-geometry-and-orbit-design`

Following is the main run file:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% AE4878 Mission Geometry and Orbit Design %%%%%%%%%%
%%%%%%% Assignment 10 - Week 3.7 - ORBDSGN-11 v2 %%%%%%%%%
%%%%% Author Info: Ali Nawaz; Student ID - 4276477 %%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Definite the geographic extreme points on Indonesia
east1_deg = 94+58*(1/60) + 22*(1/(60*60)); % Westernmost longitude [deg]
```

```
east2_deg = 141+1*(1/60) + 10*(1/(60*60)); % Easternmost longitude [deg]

north_deg = 6 + 4*(1/60) + 30*(1/(60*60)); % Northernmost lattitude [deg]
south_deg = 11 + 0*(1/60) + 27*(1/(60*60));% Southernmost lattitude [deg]

%%  Select i,j,k,a,raandot and mean motion for orbits within the given altitude range
disp('Selecting i, j, k,a,raandot and mean motion for orbits within the given altitude ...
    range...');
alt_min = 550; % minimum altitude [km]
alt_max = 650; % maximum altitude [km]

i = [south_deg:1:2*south_deg];
j = [10:1:100];
k = [1:1:7];
% i = [south_deg:1:15];
% j = [10:1:20];
% k = 1;
% res = i [deg],j [deg],k [deg],semi major axis [km] and altitude [km],
% rate of change of RAAN  and satellite mean motion in [deg/sidreal day];
% for given choice of altitude range [alt_min, alt_max]
res =  ijka_val_alt_cons(i,j,k,alt_min, alt_max );
disp('Step Complete');
%% Ground track and number of passages over the extreme points of Indonesia
disp('Generating the best area coverage option...');
i = res(:,1);
j = res(:,2);
k = res(:,3);
a = res(:,4);
raandot = res(:,6);
mean_motion = res(:,7);
e = 0; % circular orbit assumption
rev = 100;
rho2 = 90; % Angular distance of P from S [deg]
phi1_0 = 90; % initial azimuth S about C [deg]
phi2_0 = 0; % initial azimuth P about S [deg]
dt = 1; % Time step for groundtrack generation [s]
fov = 5.72; % minimum field of view [deg]
% Store the following best variables in result:
% [i(z), j(z), k(z), a(z), raandot(z), mean_motion(z), e,rev,rho1, ...
    rho2,phi1_0,phi2_0,dt,east1_deg,east2_deg,north_deg,south_deg,swath, max_passage, ...
    tot_scan_length,tot_scan_area];
result = best_coverage_area(i,j,k,a,raandot,mean_motion,e,rev, rho2, ...
    phi1_0,phi2_0,dt,east1_deg,east2_deg,north_deg,south_deg,fov);

%% Generate solution groundtrack with Eckert IV projection
trackmysat(result(1,1), result(1,2), result(1,3), result(1,4), result(1,5), result(1,6), ...
    result(1,7), result(1,8), result(1,9), result(1,10), result(1,11), result(1,12), ...
    result(1,13));
trackmysat_3D_passagebound(result(1,1), result(1,2), result(1,3), result(1,4), result(1,5), ...
    result(1,6), result(1,7), result(1,8), result(1,9), result(1,10), result(1,11), ...
    result(1,12), result(1,13), result(1,14),result(1,15), result(1,16), result(1,17),'on');

%% MONTE CARLO FOR SCHEME 3, 2 IMPULSIVE SHOTS DELTA V ESTIMATION
% Store the estimations of Delta i_1, Delta i_2 and correponding Delta V
ΔV_lst = [];
Δ_i1_lst = [];
Δ_i2_lst = [];

% Orbital Parameters
i_target = 11.0075; % Target inclination [deg]
i_park = 13.73740; % Parking orbit inclination angle [deg]
Δ_i = abs(i_park-i_target); % Total inclination change [deg]
mu_E = 398600.441*10^9; % Gravitational parameter of Earth [m^3/s^2]
R_E = 6378.136*10^3; % Radius of Earth [m]
a_final = R_E + 640.5237*10^3; % Satellite orbital altitude [m]
a_park = R_E + 185*10^3; % Satellite parking orbital altitude [m]

Vpark = sqrt(mu_E/a_park); % Satellite velocity at parking orbit [m/s]
Vfinal = sqrt(mu_E/a_final); % Satellite velocity at final orbit [m/s]

% Lower and upperbound on Δ_i1 [deg]
lb_d1 = 0 ;
ub_d1 = Δ_i;

N = 1000; % No. of samples
for run = 1:3 % Repeating each run three times
```

```
    % Random Delta i_{1} generation
    Δ_i1 = lb_d1 + (ub_d1).*rand(1,N);

    at1 = (a_final + a_park)/2 ; % semi major axis of transfer orbit 1

    Vt1a = sqrt(mu_E.*( (2./a_final) - (1./at1))); % Velocity at apogee for transfer orbit 1

    Vt1p = sqrt( mu_E*( 2./a_park - 1./at1)); % Velocity at perigee for transfer orbit 1

    % First and second impulsive shot manoeuver
    ΔV1 = sqrt ( Vpark.^2 + Vt1p.^2 - 2.*Vpark.*Vt1p.*cosd(Δ_i1));
    ΔV2 = sqrt ( Vt1a.^2 + Vfinal.^2 - 2.*Vt1a.*Vfinal.*cosd(Δ_i - Δ_i1));

    % Total ΔV
    ΔV = ΔV1 + ΔV2;

    % Appending Δ_i1, Δ_i2 and ΔV
    Δ_i1_lst = [Δ_i1_lst,Δ_i1];
    Δ_i2_lst = [Δ_i2_lst,Δ_i - Δ_i1];
    ΔV_lst = [ΔV_lst, ΔV];
end
% Find the index for minimum Δ V required
index = find(ΔV_lst == min(ΔV_lst));
result_scheme3 = [Δ_i1_lst(index),Δ_i2_lst(index), ΔV_lst(index)]; % best choice of Delta_i1, ...
    Delta_i2 and corresponding Delta_V
figure()
scatter(Δ_i1_lst,Δ_i2_lst,50,ΔV_lst,'filled','MarkerEdgeColor','k')
grid on
hold on
cmap = colorbar;
title(cmap,'\Delta V [m/s]')
cmap
scatter(Δ_i1_lst(index),Δ_i2_lst(index),50, Δ...
    V_lst(index),'MarkerFaceColor','r','MarkerEdgeColor','k');
xlabel({'\Delta i_{1} [deg]'});
ylabel({'\Delta i_{2} [deg]'});
title({'Behaviour of \Delta V for varying \Delta i_{1} and \Delta i_{2}'})
hold off

figure()
plot3(Δ_i1_lst,Δ_i2_lst,ΔV_lst,'o','MarkerFaceColor',[0 0.7 0.7],'LineWidth',0.5)
grid on
xlabel({'\Delta i_{1} [deg]'});
ylabel({'\Delta i_{2} [deg]'});
zlabel({'\Delta V [m/s]'});
title({'Behaviour of \Delta V for varying \Delta i_{1} and \Delta i_{2}'})

%% MONTE CARLO FOR SCHEME 4, 3 IMPULSIVE SHOTS DELTA V ESTIMATION
% Store the estimations of a_int, Delta i_1, Delta i_2 and correponding Delta V
a_int_lst  = [];
ΔV_lst = [];
Δ_i1_lst = [];
Δ_i2_lst  = [];

i_target = 11.0075; % Target inclination [deg]
i_park = 13.73740; % Parking orbit inclination angle [deg]
Δ_i = abs(i_park-i_target); % Total inclination change [deg]
mu_E = 398600.441*10^9; % Gravitational parameter of Earth [m^3/s^2]
R_E = 6378.136*10^3; % Radius of Earth [m]
a_final = R_E + 640.5237*10^3; % Satellite orbital altitude [m]
a_park = R_E + 185*10^3; % Satellite parking orbital altitude [m]

Vpark = sqrt(mu_E/a_park); % Satellite velocity at parking orbit [m/s]
Vfinal = sqrt(mu_E/a_final); % Satellite velocity at final orbit [m/s]

% Lower and upperbound on Δ_i1 [deg]
lb_d1 = 0 ;
ub_d1 = Δ_i;
% Lower and upperbound on Δ_i2 [deg]
lb_d2 = 0;
ub_d2 = Δ_i;
% Lower and upperbound on intermediate semi major axis [m]
lb_aint = a_final;
ub_aint = a_final*2;

```

```
N = 1000; % No. of samples
for run = 1:3 % Repeating each run three times

    % generate 1000 random samples of unknowns
    Δ_i1 = lb_d1 + (ub_d1).*rand(1,N);  % Delta i1 [deg]
    Δ_i2 = lb_d2 + (ub_d2).*rand(1,N);  % Delta i2 [deg]
    a_int = lb_aint + (ub_aint).*rand(1,N); % intermedita semi major axis [m]

    at1 = (a_int + a_park)/2 ; % semi major axis of transfer orbit 1 [m]
    at2 = (a_final + a_int)/2; % semi major axis of transfer orbit 2 [m]

    Vt1a = sqrt(mu_E.*( (2./a_int) - (1./at1))); % Velocity at apogee for transfer orbit 1 [m/s]
    Vt2a = sqrt(mu_E.* ( (2./a_int) - (1./at2))); % Velocity at apogee for transfer orbit 2 [m/s]

    Vt1p = sqrt( mu_E*( 2./a_park - 1./at1)); % Velocity at perigee for transfer orbit 1 [m/s]
    Vt2p = sqrt( mu_E*( 2./a_final - 1./at2)); % Velocity at perigee for transfer orbit 2 [m/s]

    % Delta V for three separate stages of impulsive shots [m/s]
    ΔV1 = sqrt ( Vpark.^2 + Vt1p.^2 - 2.*Vpark.*Vt1p.*cosd(Δ_i1));
    ΔV2 = sqrt ( Vt1a.^2 + Vt2a.^2 - 2.*Vt1a.*Vt2a.*cosd(Δ_i2));
    ΔV3 = sqrt ( Vt2p.^2 + Vfinal.^2 - 2.*Vt2p.*Vfinal.*cosd(Δ_i - Δ_i1 - Δ_i2));

    % Total Delta V
    ΔV = ΔV1 + ΔV2 + ΔV3;

    % Store results
    Δ_i1_lst = [Δ_i1_lst,Δ_i1];
    Δ_i2_lst = [Δ_i2_lst,Δ_i2];
    a_int_lst = [a_int_lst,a_int];
    ΔV_lst = [ΔV_lst, ΔV];
end
% Find the index for minimum Δ V required
index = find(ΔV_lst == min(ΔV_lst));
% Store the best choice a_int [m], Delta_i1 [deg], Delta_i2 [deg], Delta_i3
% [deg], Delta_V [m/s]
result_scheme4 = [a_int_lst(index),Δ_i1_lst(index), Δ_i2_lst(index),Δ_i - Δ_i1_lst(index) - Δ...
    _i2_lst(index),ΔV_lst(index)];
figure()
scatter3(Δ_i1_lst, Δ_i2_lst,a_int_lst,35,ΔV_lst,'filled','MarkerEdgeColor','k')
hold on
cmap = colorbar;
title(cmap,'\Delta V [m/s]')
cmap
scatter3(Δ_i1_lst(index), Δ_i2_lst(index),a_int_lst(index),50, Δ...
    V_lst(index),'MarkerFaceColor','r','MarkerEdgeColor','k');
xlabel({'\Delta i_{1} [deg]'});
ylabel({'\Delta i_{2} [deg]'});
zlabel({'a_{intermediate} [m]'});
title({'Behaviour of \Delta V for varying a_{intermediate}, \Delta i_{1} and \Delta i_{2}'})
hold off

%% Orbital decay
a_final = R_E + 640.5237*10^3; % Satellite orbital altitude [m]
T_2pi = 2*pi*sqrt((a_final^3)/mu_E); % [s] Orbital period [s]
mission_lifetime = 5; % [years]
total_rev = (365*mission_lifetime*86400)/(T_2pi); % Total number of orbital revolutions over ...
    given lifetime.
CD = 2.5; % Drag co-efficient
A = 0.3*4; % Max cross sectional area [m/s^2]
rho_0 = 1.225; %[kg/m^3]
R = 287; % ideal gas constant[J/kg/K]
g0 = 9.81; % gravitational constant [m/s^2]
T = 240; % base temperature [K]
mass = 100; % max S/C mass [kg]
a_current = a_final;
v_current = sqrt(mu_E*(1/a_current));
a_list = []; % accumulate a at every instance
da_list = []; % accumulate the decrease in altitude da at every instance
dv_list = []; % accumulate Δ v required
for z = 1:total_rev
    da = -2*pi*CD*(A/(mass))*rho_0*exp(-(a_current-R_E)*g0/(R*T))*a_current^2; % loss due to ...
        atmospheric drag
    dv = pi*CD*(A/(mass))*rho_0*exp(-(a_current-R_E)*g0/(R*T))*a_current*v_current;
    a_current = a_current + da;
    v_current = sqrt(mu_E*(1/a_current));
    a_list = [a_list;a_current];
```

```
    da_list = [da_list;da];
    dv_list = [dv_list;dv];
end
a_decrease = sum(da_list); % total decrease in semi major axis [m]
dv_decay = sum(dv_list); % total Δ v required for orbital decay compensation [m/s]
```

Following is the function “ijka” used to estimate Semi-major axis and altitude in [km], Rate of change of RAAN and mean motion of the satellite in [deg/s]. Taking inclination angle i [deg], no. of orbits [j] in [k] days as inputs.

```
function [semi_major_axis, altitude,RAANdot,mean_motion] = ijka(i,j,k)
    % ijka function
    % INPUT PARAMETERS : inclination angle i [deg], no. of orbits [j] in [k] days
    % OUTPUT PARAMETERS: Semi-major axis and altitude in [km], Rate of change of RAAN
    % and mean motion of the satellite in [deg/s]
    % Standard parameters

    % Script Author: Ali Nawaz, Delft University of Technology.
    % Aerospace Engineering Faculty [LR]
    % Mechanical, Maritime and Materials Engineering Faculty [3ME]

    mu = 3.98600441*10^(14); % [m^3 s^-2] standard gravitational parameter of Earth from ...
        Wikipedia
    D = 86164.10035; % sidereal rotation period of the Earth [s]. Source OCDM pg.81
    RE = 6378.136*10^(3); % radius of Earth [m]. Source: OCDM parameter list, intro lecture, ...
        slide 16..
    J2 = 1082.63*10^(-6); % Dimensionless J2 parameter. Source: Lecture intro, slide 16
    ev = 0; % eccentricity = 0 for circular orbit
    repeat_pattern = j./k; % repeat pattern, j orbits in k days
    sidereal_day = 86164.1004; % [sec]
    a = [];
    a0_list = [];
    RAANdot = [];
    mean_motion=[];
    for z = 1:length(j)
        % initialize semi major axis a0
        a0 = (mu^(1/3) )* ( 2*pi*j(z)/D*k(z))^(-2/3); % [m]
        % List of a0
        a0_list = [a0_list,a0];
        % Avoiding -ve altitude initialisation for non-integar repeat patterns
        if (a0-RE) < 0
            a0 = RE;
        end

        acc = 10^-5; % accuracy [m];
        err = 10000; % Initializes while loop.
        iter = 0;
        a(iter+1,z) = [a0];
        iv_rad = deg2rad(i); % inclination value in radians

        while err > acc
            iter = iter +1;
            L_dot_deg = 360; % Rotation rate of the Earth in deg/day
            L_dot_rad = deg2rad(L_dot_deg); % Rotation rate of the Earth in rad/day

            Omega_dot = ...
                -(3/2)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*cos(iv_rad)*((1-ev^2)^(-2))*(180/pi)*D; ...
                %  Rate of change of Right ascension of ascending node in deg/sidereal day
            omega_dot = ...
                (3/4)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*(5*(cos(iv_rad)^2)-1)*((1-ev^2)^(-2))*(180/pi)*D; .
                % Rate of change of argument of pericenter in deg/siderealday
            M_dot = (3/4)*J2*sqrt(mu)*(RE^2)*(a(iter,z)^-3.5)*(3*(cos(iv_rad)^2) ...
                -1)*((1-ev^2)^(-1.5))*(180/pi)*D; % Rate of change of Mean anomaly ...
                [deg/siderealday]

            n = (j(z)/k(z)) * (L_dot_deg - Omega_dot) - (omega_dot + M_dot); % Mean motion of ...
                the satellite in orbit [deg/sidreal day]

            a(iter+1,z) = ( mu/ ( ( (n/D)*(pi/180))^2 ))^(1/3); % Updated semi-major axis [m];

            if iter> 100000
                % Infinite loop avoidance criteria
                break;
            end
```

```
%          err = a(iter+1,z) - a(iter,z);
        end
        RAANdot(z) = Omega_dot; % rate of change of RAAN [deg/sidreal day]
        mean_motion(z) = n; % mean motion of the satellite in orbit [deg/sidereal day]
    end
    % Orbit altitude in km
    semi_major_axis = (a(end,:)')*10^(-3); % [km]
    altitude = (a(end,:) - RE)'*10^(-3); % [km]
    RAANdot = RAANdot'./sidereal_day; % [deg/s]
    mean_motion = mean_motion'./sidereal_day; % [deg/ s]
end
```

Following is the function “`ijka_val_alt_cons(i,j,k,alt_min, alt_max )`” used to obtain possible orbits constrained by the altitude ranges.

```
function res =  ijka_val_alt_cons(i,j,k,alt_min, alt_max )
    %% INPUT PARAMETERS
    % i = range of preferred inclination angles [deg]
    % j,k = range of preferred j orbits in k days
    % alt_min = minimum desired altitude [km]
    % alt_max = maximum desired altitude [km]

    %% OUTPUT PARAMETERS : Combinations bounded by the altitude constraints
    % [i [deg],j [no. of orbits],k [no. of days], semi major axis sma [km],
    % orbital altitude alt [km], rate of change of RAAN raandot [deg/ sidereal day],
    % mean motion n [deg/sidereal day] ]

    % Script Author: Ali Nawaz, Delft University of Technology.
    % Aerospace Engineering Faculty [LR]
    % Mechanical, Maritime and Materials Engineering Faculty [3ME]

    ijka_val = [];
    res = [];
    entry =0;
    for z = 1:length(i)
        for zz = 1:length(j)
            for zzz = 1:length(k)
                entry = entry +1;
                [sma, alt,raandot,mean_motion] = ijka(i(z),j(zz),k(zzz)); % Semi major axis ...
                    and altitude in [km]
                ijka_val = [ijka_val;[i(z),j(zz),k(zzz),sma,alt,raandot,mean_motion]]; % ...
                    store i [deg],j [deg],k [deg],semi major axis [km] and altitude [km], ...
                    rate of change of RAAN  and satellite mean motion in [deg/sidreal day]
                if ijka_val(entry,5) ≥ alt_min && ijka_val(entry,5) ≤ alt_max
                    res = [res; ijka_val(entry,:)];
                end
            end
        end
    end
end
```

Following is the function '`best_coverage_area`' used to estimate orbital parameters resulting in maximum area scanned.

```
function result = best_coverage_area(i,j,k,a,raandot,mean_motion,e,rev, rho2, ...
    phi1_0,phi2_0,dt,east1_deg,east2_deg,north_deg,south_deg,fov)
RE = 6378.136; % [km]

final_info_all = [];
altitude =[];
for z = 1:length(i)
         disp('Best coverage area : generating passage groundtrack');
         rho1 = i(z); % Angular distance of S from C [deg]
        [longval, latval,combi] = ...
            passage_groundtrack(i(z),j(z),k(z),a(z),raandot(z),mean_motion(z),e,rev,rho1,rho2,phi1_0,phi2_0,d
        n = combi(:,1);
        % Entrance and exit points for
        enter_exit = [combi(1,:)];
        entry = 1;
        diff_enter_exit = [];
        pass_no = n(1,1);
        % Estimate the arc difference between the longitude/lattitude of
```

```
        % entrace and exit in the observation zone
        disp('Best coverage area : generating arc difference');
        for zz = 1:length(n)
            if n(zz,1) ≠ pass_no
                pass_no = n(zz,1);
                entry = entry +1;
                enter_exit = [enter_exit; combi(zz-1,:); combi(zz,:)];
                diff_enter_exit = [diff_enter_exit; [combi(zz-1,1), abs( ...
                    abs(enter_exit(entry,2)) - abs(enter_exit(entry-1,2))), abs( ...
                    abs(enter_exit(entry,3)) - abs(enter_exit(entry-1,3)) )]];
            else
                continue
            end
        end

        passage_no = []; % Passage number [-]
        scan_arc = []; % Arc of scan over the passage [deg]
        scan_length = []; % Length of scan over the passage [km]
        area_scan = []; % Area of scan over the passage [km^2]

        disp('Best coverage area : estimate area coverage');
        for zzz = 1:length(diff_enter_exit(:,1))
            theta_a = diff_enter_exit(zzz,3); % Lattitude difference of passage [deg]
            theta_b = diff_enter_exit(zzz,2); % Longitude difference of passage [deg]

            % Angle Side Angle rule
                A = i(z); % Angle [deg]
                B = 90; % Angle [deg]
                c = theta_b; % Side angle [deg]
                %  Solution 1
            % Define Hc
            if 0 ≤ wrapTo360(c) && wrapTo360(c)<180
                Hc = 1;
            elseif 180 ≤ wrapTo360(c) && wrapTo360(c)< 360
                Hc = -1;
            end
            % Define HA
            if 0 ≤ wrapTo360(A) && wrapTo360(A)<180
                HA = 1;
            elseif 180 ≤ wrapTo360(A) && wrapTo360(A)< 360
                HA = -1;
            end
            % Define HB
            if 0 ≤ wrapTo360(B) && wrapTo360(B)<180
                HB = 1;
            elseif 180 ≤ wrapTo360(B) && wrapTo360(B)< 360
                HB = -1;
            end
            % Solution 1
            C1 = acos2(-cosd(A)*cosd(B) + sind(A)*sind(B)*cosd(c), Hc);
%             a1 = acos2( ( cosd(A) + cosd(B)*cosd(C1) )/( sind(B)*sind(C1) ), HA);
            b1 = acos2( ( cosd(B) + cosd(A)*cosd(C1))/(sind(A)*sind(C1)), HB);
%             % Solution 2
%             C2 = 360-C1; % [deg]
%             a2 = wrapTo360(a1 + 180); % [deg]
%             b2 = wrapTo360(b1 + 180); % [deg]
            h = b1;
            swath = 2*(a(z) - RE)*tand(fov); % swath [km]
            scan_arc = [scan_arc;h]; % [deg]
            scan_length = [scan_length; deg2rad(h)*RE]; % [km]
            area_scan = [area_scan; deg2rad(h)*RE*swath]; %   Area of scan in [km^2]
            passage_no = [passage_no; diff_enter_exit(zzz,1)]; % Passsage number
        end
        max_passage = passage_no(end); % [-]
        tot_scan_length = sum(scan_length); % [km]
        tot_scan_area = sum(area_scan); % [km^2]
        final_info_all(z,:) = [i(z), j(z), k(z), a(z), raandot(z), mean_motion(z), ...
            e,rev,rho1, rho2,phi1_0,phi2_0,dt,east1_deg,east2_deg,north_deg,south_deg,swath, ...
            max_passage, tot_scan_length,tot_scan_area];
        message = strcat('Progress ', num2str((z/length(i))*100),'%');
        disp(message);
end
        row_index = find( final_info_all(:,21) ==  max(final_info_all(:,21)) ); % Find the ...
            row index with maximum total area scanned
%          row_index = find( final_info_all(:,20) ==  max(final_info_all(:,20)) ); % Find the ...
    row index with maximum total scanned length
```

```
%         row_index = find( final_info_all(:,19) ==  max(final_info_all(:,19)) ); % Find the ...
    row index with maximum number of passages
        result = final_info_all(row_index,:);
        disp('Best coverage analysis complete.');
end
```

Following is the function '`passage_groundtrack`' used to generate the groundtrack passages.

```
function [longval,latval,combi] = ...
    passage_groundtrack(i,j,k,a,raandot,mean_motion,e,rev,rho1,rho2,phi1_0,phi2_0,dt,...
westlong,eastlong,northlat,southlat);
    % INPUT PARAMETERS:

    % i = inclination angle [deg]
    % e = eccentricity of the orbit
    % rev = number of orbital revolutions
    % a = semi-major axis in [km]
    % rho1 = angular distance of Spacecraft from the rotational axis of the
    % Core. [deg]
    % rho2 = angular distance of the Projected point from the Spacecraft
    % [deg]
    % phi1_0 = initial azimuth S about C [deg]
    % phi2_0 = initial azimuth P from S [deg]
    % dt = resolution of time step for simulation [s]

    % Definition of observation region
    % Enter longitude values in the range -180 deg to 180 deg
    % westlong = western most longitude value of the region under
    % observation [deg]
    % eastlong = eastern most longitude value of the region under
    % observation [deg]
    % Enter lattitude values in the range -90 deg to 90 deg
    % northlat = northern most lattitude value of the region under
    % observation [deg]
    % southlat = southern most lattitude value of the region under
    % observation [deg]

    % OUTPUT PARAMETERS:
    % longitude and lattitude of passage in the observation region [deg]
    % longval [deg]
    % latval [deg]

    RE= 6378.136; % Radius of Earth[km]
    mu_E = 398600.441; % Gravitational parameter of Earth[ km^3/s^2]
    J2 = 1082.63*10^(-6); % Dimensionless J2 parameter. Source: Lecture intro, slide 16
    sidereal_day = 86164.1004; % [sec]
    phi1 = [];
    phi2 = [];
%     t = [0:dt:rev*sqrt( ((a^3)/(mu_E))*(4*pi^2)) ]; % Time of given S/C revolutions
    t = [0:dt:24*60*60]; % Time of given S/C revolutions
    alpha = []; % stores propagated azimuth values
    ∆ = []; % stores propagated elevation values
%     v = []; % stores propagated velocity
%     Psi = []; % Store propagated direction of motion of P
    omega1 = rad2deg((-2*pi)/(86164.1004)) + raandot/sidereal_day; % [deg/s]
    omega2 = mean_motion;% [deg/s]
    for k = 1:length(t)
        % run the dual_axis_spiral function for given time stamp
%         [azimuth, elevation, velocity, direction_of_motion] = dual_axis_spiral(rho1, ...
    rho2,omega1,omega2, phi1_0, phi2_0,phi1,phi2, t(k));
        [azimuth, elevation, ¬, ¬] = dual_axis_spiral(rho1, rho2,omega1,omega2, phi1_0, ...
            phi2_0,phi1,phi2, t(k));
        alpha = [alpha,azimuth];
        ∆ = [∆, elevation];
%         v = [v, velocity];
%         Psi = [Psi, direction_of_motion];
    end

    longval = []; % initialising the longitude values of passage in the observation region [deg]
    latval = []; %initialising the lattitude values of passage in the observation region [deg]
    combi = []; % initialising passage number and the corresponding longitutde and lattitude ...
        values of the passage
    n = 0;
    flag = 1;

```

```
    for z = 1:length(alpha)
        if (alpha(z) ≥ westlong && alpha(z)≤ eastlong ) && (Δ(z) ≥ southlat && Δ(z)≤ northlat)
            longval = [longval;alpha(z)];
            latval = [latval;Δ(z)];
            if flag == 0
                n = n+1;
            end
            combi = [combi; [n, alpha(z), Δ(z)]];
            flag = 1;
        else
            flag = 0;
            continue
        end
    end
%     for z = 1:length(Δ)
%         if Δ(z) ≥ southlat || Δ(z)≤ northlat
%             latval = [latval;Δ(z)];
%         else
%             continue
%         end
%     end
%     longval = azimuth;
%     latval = elevation;
end
```

Following is the function 'dual_axis_spiral' used to generate ground tracks.

```
function [azimuth, elevation, velocity, direction_of_motion] = dual_axis_spiral(rho1, ...
    rho2,omega1,omega2, phi1_0, phi2_0, phi1, phi2, t)
    % INPUT PARAMETERS [deg]:[ rho1, rho2,omega1,omega2, phi1_0, phi2_0, t ]
    % Point P with coordinates (alpha, Δ) is rotating with angular
    % velocity omega2 [deg/s] about the secondary axis S, which in turn is rotating
    % with angular velocity omega1 [deg/s] about the primary or central axis C
    % located at (0 deg, 90 deg)
    % rho1: angular distance of S from C
    % rho2: angular distance of P from S
    % phi1_0 = initial azimuth of S about C relative to alpha = 0
    % phi2_0 = initial azimuth of P about S relative to C
    % t = time vector [s]
    % OUTPUT PARAMETERS : [azimuth, elevation, velocity, direction_of_motion]
    % alpha = Azimuth of P about C [deg]
    % Δ = Elevation of P relative to C [deg]
    % v = Velocity of P [m/s]
    % Psi = Direction of motion of P [deg]
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Script constructed with the aid of Table 8-8 OCDM
    % Orbit Constellation Design and Management [OCDM] by James R Wertz 2nd
    % Printing.
    % Script Author: Ali Nawaz, Delft University of Technology.
    % Aerospace Engineering Faculty [LR]
    % Mechanical, Maritime and Materials Engineering Faculty [3ME]
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if rho1<0 || rho1>180 || rho2<0 || rho2>180
        msg = ('Please enter 0≤ rho1, rho2 ≤180 degrees');
    end
    if isempty(phi1_0) == 1
        phi1 = phi1 ;% [deg] Azimuth of S about C relative to alpha = 0
    else
        phi1 = phi1_0 + omega1.*t; % [deg] Azimuth of S about C relative to alpha = 0
    end

    if isempty(phi2_0) == 1
        phi2 = phi2 ;% [deg] Azimuth of P about S relative to C
    else
        phi2 = phi2_0 + omega2.*t; % [deg] Azimuth of P about S relative to C
    end
    % Equations of motion for dual axis spiral
    Δ = 90 - acosd( cosd(rho1)*cosd(rho2) + sind(rho1)*sind(rho2)*cosd(phi2) ); % [deg] ...
        Elevation of P relative to C, -90≤ Δ < 90 deg
    Δ_alpha = acos2( ( cosd(rho2) - cosd(rho1)*sind(Δ) )/( sind(rho1)*cosd(Δ) ), ...
        -hemisphere(phi2) );% change in alpha [deg] 0 deg ≤ Δ_alpha <360 deg
    alpha = wrapTo360( phi1 + Δ_alpha); % [deg] alpha parameter, Azimuth of P about C. 0 deg ≤...
         alpha < 360 deg

    % Determining the rotation and orientation of Euler axis, E
```

```
    Δ_Ebar = myatan2d(deg2rad(omega2)*sind(rho1),( deg2rad(omega1) + ...
        deg2rad(omega2)*cosd(rho1) )) ; % [deg] co-elevation Δ_E' of Euler axis 0 deg ≤Δ...
        _E'≤180 deg
    omegaE = omega2* ( sind(rho1)/sind(Δ_Ebar) ); % Rotation about Euler axis, E [deg/s]
%     omegaE = sqrt( omega1^2 +omega2^2 + 2*omega1*omega2*cos(rho1)); % [deg/s] Alternate ...
    approach to above solution
    rhoE = acosd( cosd(Δ_Ebar)*sind(Δ) + sind(Δ_Ebar)*cosd(Δ)*cosd(Δ_alpha));% [deg] Angle ...
        from P to Euler axis E (Instantaneous Rotation Axis)
    Δ_Psi = acos2( ( cosd(Δ_Ebar) - cosd(rhoE)*sind(Δ))/( sind(rhoE)*cosd(Δ)), ...
        hemisphere(Δ_alpha));% [deg] Change in direction of Motion of P
    v = deg2rad(omegaE)*sind(rhoE); % [m/s] Velocity of P 0≤ v ≤ omegaE in radians
    Psi = wrapTo360(Δ_Psi -90); % [deg] 0 deg ≤ Psi < 360 deg
    % Results
    azimuth = alpha;
    elevation = Δ;
    velocity = v;
    direction_of_motion = Psi;
end
```

Following are the functions 'acos2' and 'mytan2d' used in the function 'dual_axis_spiral'.

```
function res = acos2( cos_phi, H_phi)
    % Takes in degrees, outputs in degrees. Enter cos(theta) and
    % hemisphere function H(theta).
%     res = wrapTo360( rad2deg(H_phi*acos(cos_phi)) );
%     res = wrapTo360( H_phi*acosd(cos_phi) );
    if abs(cos_phi)>1
        cos_phi = 1;
    end
    res = mod(H_phi*acosd(cos_phi),360);
end


function rest = myatan2d(y,x)
    % Outputs degrees.
    % According to definition of OCDM A-41
    % Orbit Constellation Design and Management [OCDM] by James R Wertz 2nd
    % Printing.
    % Script Author: Ali Nawaz, Delft University of Technology.
    % Aerospace Engineering Faculty [LR]
    % Mechanical, Maritime and Materials Engineering Faculty [3ME]
    if ( 0 ≤ y ) && ( y ≤ abs(x) ) && x > 0
        rest = atand(y/x);
    elseif -abs(x) ≤ y && y ≤0 && x > 0
        rest = 360 + atand(y/x);
    elseif abs(y) ≤ abs(x) && x<0
        rest = 180 + atand(y/x);
    elseif y≥ abs(x)
        rest = 90 - atand(x/y);
    elseif y< -abs(x)
        rest = 270 - atand(x/y);
    end
end
```

Following is the function 'trackmysat' used to generate 2D Eckert IV projected ground tracks.

```
function res =  trackmysat(i,j,k,a,raandot,mean_motion,e,rev,rho1,rho2,phi1_0,phi2_0,dt);
    % Input parameters:
    % i = inclination angle [deg]
    % e = eccentricity of the orbit
    % rev = number of orbital revolutions
    % a = semi-major axis in [km]
    % rho1 = angular distance of Spacecraft from the rotational axis of the
    % Core. [deg]
    % rho2 = angular distance of the Projected point from the Spacecraft
    % [deg]
    % phi1_0 = initial azimuth S about C [deg]
    % phi2_0 = initial azimuth P from S [deg]
    % dt = resolution of time step for simulation [s]
    RE= 6378.136; % Radius of Earth[km]
    mu_E = 398600.441; % Gravitational parameter of Earth[ km^3/s^2]
    J2 = 1082.63*10^(-6); % Dimensionless J2 parameter. Source: Lecture intro, slide 16
    sidereal_day = 86164.1004; % [sec]
```

```
    phi1 = [];
    phi2 = [];
%     t = [0:dt:rev*sqrt( ((a^3)/(mu_E))*(4*pi^2)) ]; % Time of given S/C revolutions
    t = [0:dt:24*60*60]; % Time of given S/C revolutions
    alpha = []; % stores propagated azimuth values
    Δ = []; % stores propagated elevation values
    v = []; % stores propagated velocity
    Psi = []; % Store propagated direction of motion of P
    omega1 = rad2deg((-2*pi)/(86164.1004)) + raandot/sidereal_day; % [deg/s]
    omega2 = mean_motion;% [deg/s]
    for k = 1:length(t)
        % run the dual_axis_spiral function for given time stamp
        [azimuth, elevation, velocity, direction_of_motion] = dual_axis_spiral(rho1, ...
            rho2,omega1,omega2, phi1_0, phi2_0,phi1,phi2, t(k));
        alpha = [alpha,azimuth];
        Δ = [Δ, elevation];
        v = [v, velocity];
        Psi = [Psi, direction_of_motion];
    end
    % Generate fancy ground track of GPS satellite on Earth map
    groundtrack(Δ, alpha,'fancy','Groundtrack of satellite');
    % Generate simple ground track of GPS satellite on Earth map
    groundtrack(Δ, alpha,'simple','Groundtrack of satellite');
end
```

Following is the function 'trackmysat_3D_passagebound' used to generate 3D spherical groundtracks.

```
function res = trackmysat_3D_passagebound(i,j,k,a,raandot,mean_motion,e,rev,rho1,rho2,phi1_0,...
phi2_0,dt,westlong,eastlong,northlat,southlat,show_passages);
    % INPUT PARAMETERS:

    % i = inclination angle [deg]
    % e = eccentricity of the orbit
    % rev = number of orbital revolutions
    % a = semi-major axis in [km]
    % rho1 = angular distance of Spacecraft from the rotational axis of the
    % Core. [deg]
    % rho2 = angular distance of the Projected point from the Spacecraft
    % [deg]
    % phi1_0 = initial azimuth S about C [deg]
    % phi2_0 = initial azimuth P from S [deg]
    % dt = resolution of time step for simulation [s]

    % Definition of observation region
    % Enter longitude values in the range -180 deg to 180 deg
    % westlong = western most longitude value of the region under
    % observation [deg]
    % eastlong = eastern most longitude value of the region under
    % observation [deg]
    % Enter lattitude values in the range -90 deg to 90 deg
    % northlat = northern most lattitude value of the region under
    % observation [deg]
    % southlat = southern most lattitude value of the region under
    % observation [deg]

    % OUTPUT PARAMETERS:
    % longitude and lattitude of passage in the observation region [deg]
    % longval [deg]
    % latval [deg]

    % generate passage ground tracks
    if strcmp(show_passages,'on')== 1
        [longval,latval,combi] = ...
            passage_groundtrack(i,j,k,a,raandot,mean_motion,e,rev,rho1,rho2,phi1_0,phi2_0,dt,...
        westlong,eastlong,northlat,-southlat);
    end
    latrange = [-southlat:0.1:northlat];
    longrange =  [westlong:0.1:eastlong];
    list1 = [latrange',westlong*ones(size(latrange'))];
    list2 = [latrange',eastlong*ones(size(latrange'))];
    list3 = [-southlat*ones(size(longrange')),longrange'];
    list4 = [northlat*ones(size(longrange')), longrange'];

    grs80 = referenceEllipsoid('grs80','km'); % Constructing grs80 or Geodetic Reference ...
        System 1980
```

```
    figure('Renderer','opengl');
    % Axis definition
    ax = axesm('globe','Geoid',grs80,'Grid','on', 'GLineWidth',1,'GLineStyle','.','Gcolor', ...
        [0.9 0.9 0.1], 'Galtitude',100); % Generating globe axis
    ax.Position = [ 0 0 1 1]; % defining axis position
    axis equal off % no background axis
    view(3) % 3D view
    load topo; % load topography information
    land = shaperead('landareas','UseGeoCoords',true);
    hold on

    geoshow(ax,land,'FaceColor',[0.5 0.7 0.5]); %show lands
    geoshow(ax, land, 'FaceAlpha', [0]);% Show coordinates
    geoshow(topo,topolegend,'Displaytype','texturemap') % show rough topography
    demcmap(topo) % add colormap appropriate to terrain elevation data, note it doesn't show ...
        the ice sheets!
    geoshow(list1(:,1),list1(:,2),'DisplayType','MultiPoint','Marker','.', 'MarkerEdgeColor', ...
        'r');
    geoshow(list2(:,1),list2(:,2),'DisplayType','MultiPoint','Marker','.', 'MarkerEdgeColor', ...
        'r');
    geoshow(list3(:,1),list3(:,2),'DisplayType','MultiPoint','Marker','.', 'MarkerEdgeColor', ...
        'r');
    geoshow(list4(:,1),list4(:,2),'DisplayType','MultiPoint','Marker','.', 'MarkerEdgeColor', ...
        'r');
    if strcmp(show_passages,'on')== 1
        geoshow(latval,longval,'DisplayType','MultiPoint','Marker','.', 'MarkerEdgeColor', 'r');
    end
    hold off
end
```

References

[1] ICEYE: Radar imaging with microsatellites - new source for reliable real-time data. http://c-sigma.space/c-sigma-VII/briefs/ICEYE_EMSA_C-SIGMA_VII.pdf.

[2] ICEYE X1, X2, X3 - Gunter's Space Page, howpublished = http://space.skyrocket.de/doc_sdat/iceye-x.htm, note =,.

[3] Wikipedia launch site chart. https://en.wikipedia.org/wiki/List_of_rocket_launch_sites. Date of access: 01 May 2018.

[4] Pekka Laurila and Rafal Modrzewski. Validation of ICEYE Small Satellite SAR Design for Ice Detection and Imaging. *Arctic Offshore Technology Conference*, 2016. ICEYE Ltd, https://www.onepetro.org/download/conference-paper/OTC-27447-MS?id=conference-paper%2FOTC-27447-MS.

[5] Alberto Moreira. Synthetic Aperture Radar (SAR): Principles and Applications. *4th Advanced Training Course in Land Remote Sensing*, 2013. German Aerospace Center (DLR) Microwaves and Radar Institute 82230 Oberpfaffenhofen, Germany, https://earth.esa.int/documents/10174/642943/6-LTC2013-SAR-Moreira.pdf.

[6] R. Noomen. *AE4878 Mission Geometry and Orbit Design, Orbit Design v4-14*. TU Delft, 2017.

[7] Hirobumi Saito. Compact X band synthetic aperture radar on 100 kg small satellite. *IAA Small Satellite Berlin*. Japan Aerospace Exploration Agency (JAXA), Institute of Space and Astronautical Science (ISAS), https://www.dlr.de/iaa.symp/Portaldata/49/Resources/dokumente/archiv10/pdf/0301.pdf.

[8] James R. Wertz. *Orbit Constellation Design Management*. Springer, 2009.

[9] James R. Wertz and Wiley J. Larson. *Space Mission Analysis and Design, Third Edition*. Springer Space Technology Library, 2010.