Orbit Mechanics
Exercise 1: Keplerian Orbits in Space-fixed, Earth-fixed and Topocentric systems

Hsin-Feng Ho
03770686

January 6, 2024

1 Introduction

In this exercise, we will study the Keplerian orbits in different coordinate systems. The Keplerian obital elements for 5 satellites are given in the table below. The orbital elements are given in the following order: semi-major axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of perigee ω, and perigee passing time T_0 on Nov.06,2023.

Satellite	**a[km]**	**e**	**i[deg]**	**Ω[deg]**	**ω[deg]**	**T_0[s]**
GOCE	6629	0.004	96.6	210	144	01:00
GPS	26560	0.001	55	30	30	11:00
Molniya	26554	0.7	63	200	270	05:00
GEO	42164	0	0	0	50	00:00
Michibiki	42164	0.075	41	200	270	19:00

Orbit in 2D plane

In this first task we will calculate the polar coordinates of stellite postion. The Keplerian elements of the semi-major axis and eccentricity are given. These two parameters determine the shape of the orbit. Then we have to calculate the current position of the satellite on this orbit plane with the time of perigee passage, where $\nu = E = M = 0$. This allows us to calculate the orbit evolution over time.

$$\boldsymbol{r}_f(r,\nu) = \begin{bmatrix} r\cos\nu \\ r\sin\nu \\ 0 \end{bmatrix} = \begin{bmatrix} a\cos E - e \\ a\sqrt{1-e^2}\sin E \\ 0 \end{bmatrix}$$

r is the distance from the center of the ellipse to the satellite, and ν is the true anomaly. The true anomaly is the angle between the perigee and the satellite position. The eccentric anomaly E is the angle between the perigee and the position of the satellite on the ellipse. The eccentric anomaly can be calculated by solving the Kepler equation.

$$M = E - e\sin E$$

The mean anomaly M is the angle between the perigee and the position of the satellite on the circle. This is a fictitious angle and it evolves linearly in time. The mean anomaly can be calculated by the following equation.

$$M = n(t - T_0)$$

The mean motion n is the angular velocity of the satellite on the orbit. The mean motion can be calculated by the following equation.

$$n = \sqrt{\frac{\mu}{a^3}} = \sqrt{\frac{GM}{a^3}}$$

The five satellite orbit in 2D plane are shown in Figure 1. The orbit of GOCE is a circular orbit, and the orbit of GPS is a nearly circular orbit. The orbit of Molniya is a highly eccentric orbit. The orbit of GEO is a circular orbit, and the orbit of Michibiki is a nearly circular orbit.

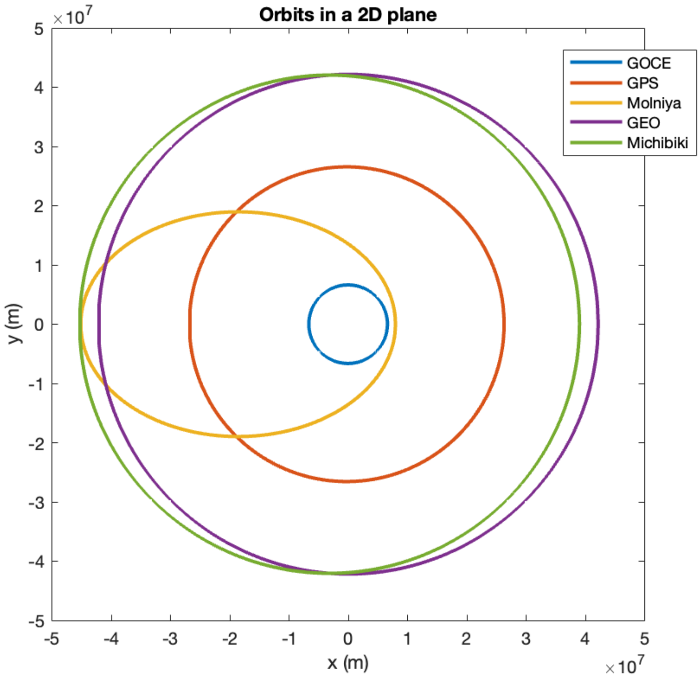

Figure 1: 2D orbit

We also have to plot the mean anomaly M, the eccentric anomaly E and the true anomaly ν over 12 hours for the GPS and Molniya satellite.

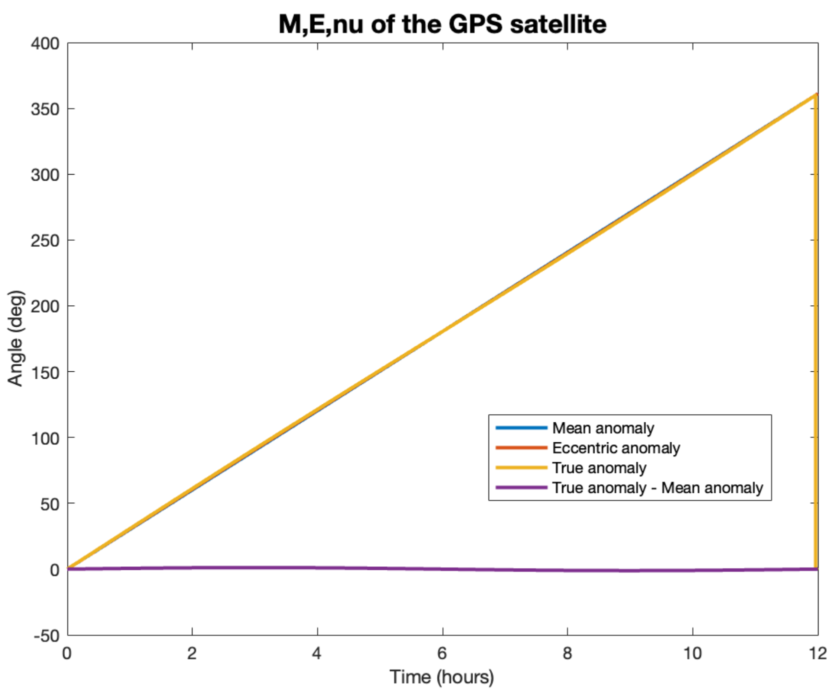

Figure 2: Mean, eccentric and true anomaly of GPS

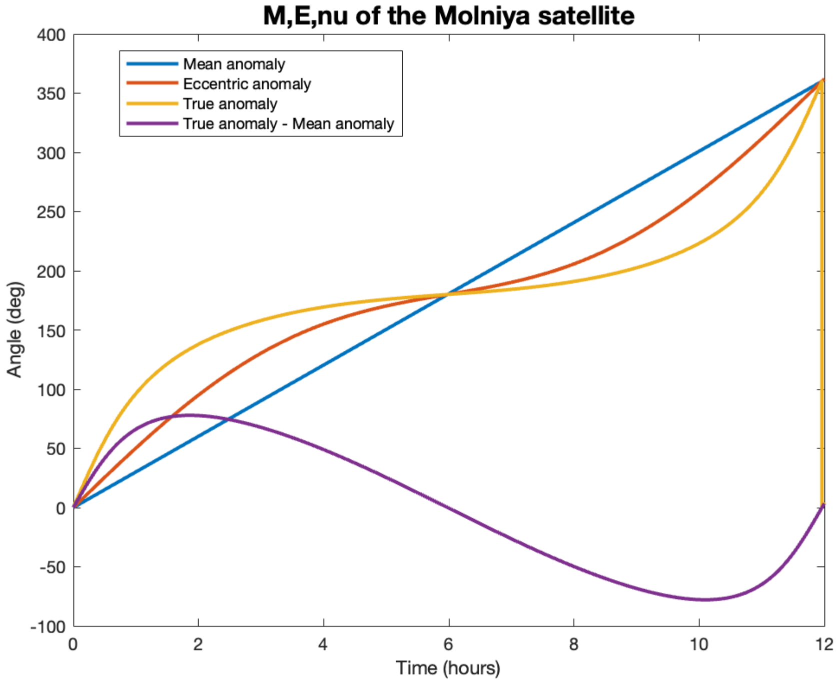

Figure 3: Mean, eccentric and true anomaly of Molniya

We can observe that the mean anomaly M is a linear function of time. The orbit of GPS is nearly circular, since the true anomaly ν increases similar with mean anomaly. On the other hand, the orbit of Molniya is highly eccentric, since the true anomaly ν increases in a periodic matter.

Space-fixed inertial system

In this second task we want to transform the satellite postion and velocity from the perifocal f-frame of 2D plane to the cartesian coordinates of 3D space in an inertial space fixed system.

$$\boldsymbol{r}_i = \boldsymbol{R}_3(-\Omega)\boldsymbol{R}_1(-i)\boldsymbol{R}_3(-\omega)\boldsymbol{r}_f$$
$$\dot{\boldsymbol{r}}_i = \boldsymbol{R}_3(-\Omega)\boldsymbol{R}_1(-i)\boldsymbol{R}_3(-\omega)\dot{\boldsymbol{r}}_f$$

After transformation we can plot the 3D orbit and the projection of 3D orbit and velocity. The 3D orbit is shown in Figure 4. The projection of 3D orbit and velocity are shown in Figure 5. The orbits of GOCE and GPS are nearly circular. The orbit of Molniya is a highly eccentric orbit. The orbit of GEO is a circular orbit, and the orbit of Michibiki is a elliptical orbit. In the velocity plot we see that the velocity of Molniya changes periodicly over time, while the other satellites have only slight changes in velocity.

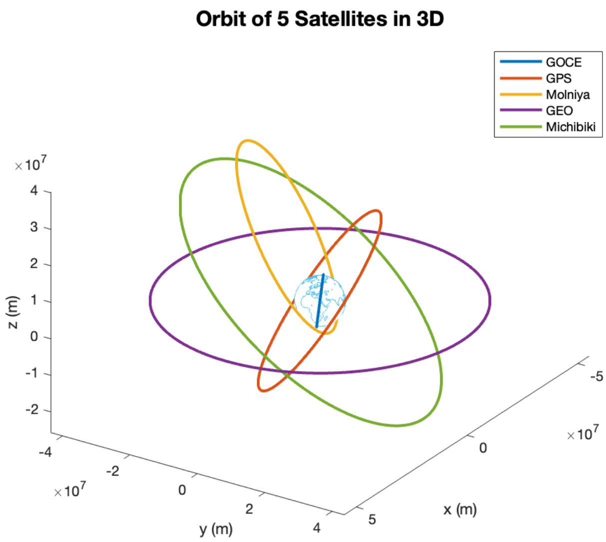

Figure 4: 3D orbit

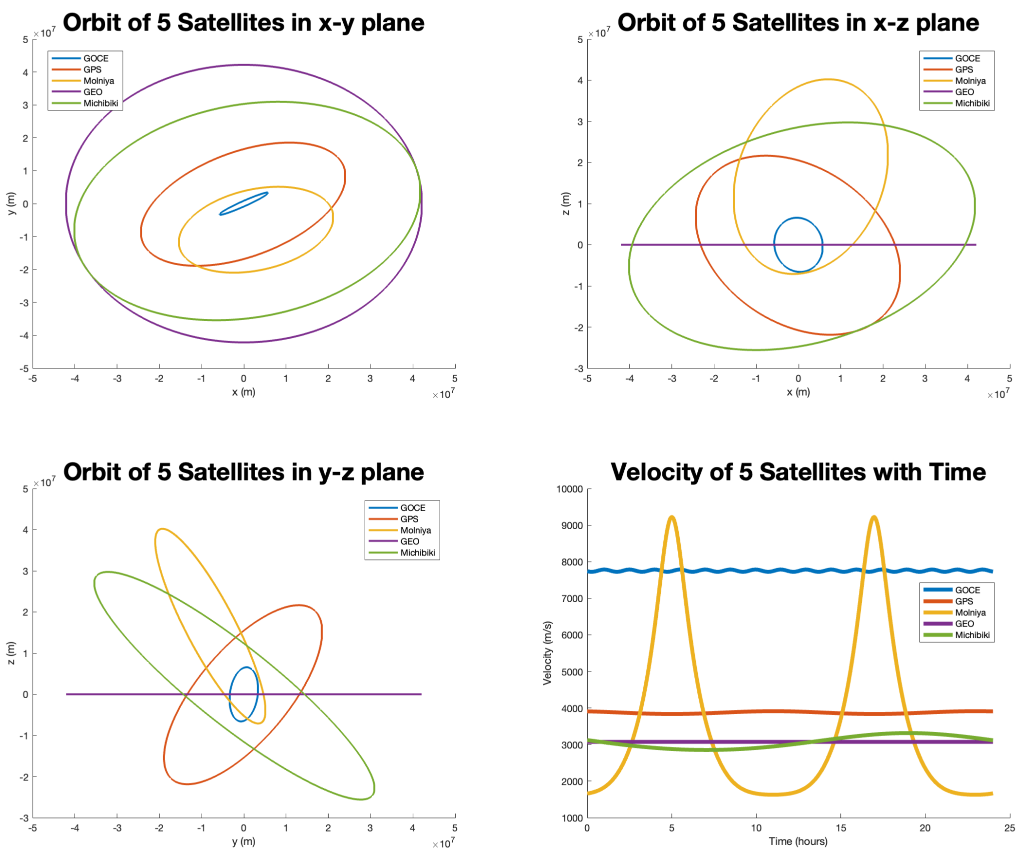

Figure 5: Projection of 3D orbit and velocity

Earth-fixed system

In this task we have to transform the satellite postion and velocity from the cartesian coordinates of 3D space in an inertial space fixed system to the cartesian coordinates of 3D space in an Earth-fixed system. The Earth-fixed system is a rotating system with respect to Earth rotation rate. The transformation can be calculated by the following equation.

$$\theta = \dot{\Omega}_E t + \text{sidereal angle}$$
$$\boldsymbol{r}_e = \boldsymbol{R}_3(-\theta)\boldsymbol{r}_i$$

which $\dot{\Omega}_E$ is the Earth rotation rate. The Earth rotation rate is calculated by $\frac{2\pi}{86164}$. The sidereal angle is the angle between the Greenwich meridian and the equinox. In this exercise we use a sidereal angle of 3 hours. The satellite position in Earth-fixed system is shown in Figure 6.

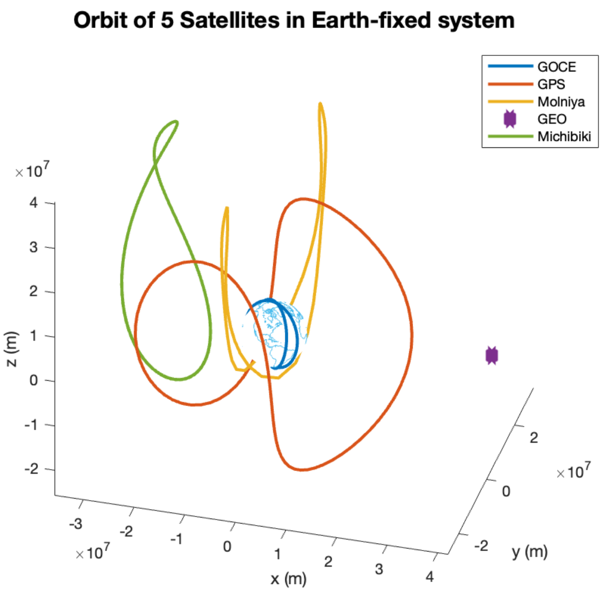

Figure 6: Earth-fixed system

The satellite ground-tracks on the Earth surface will also be calculated and shown in Figure 7.

$$\lambda = \arctan\left(\frac{y}{x}\right)$$
$$\phi = \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right)$$

After calculation we get the ground-track of the satellites. The GEO satellite stays at the same position on the Earth surface, since the GEO satellite is a geostationary satellite. And the ground track of the Michibiki satellite is a figure of eight.

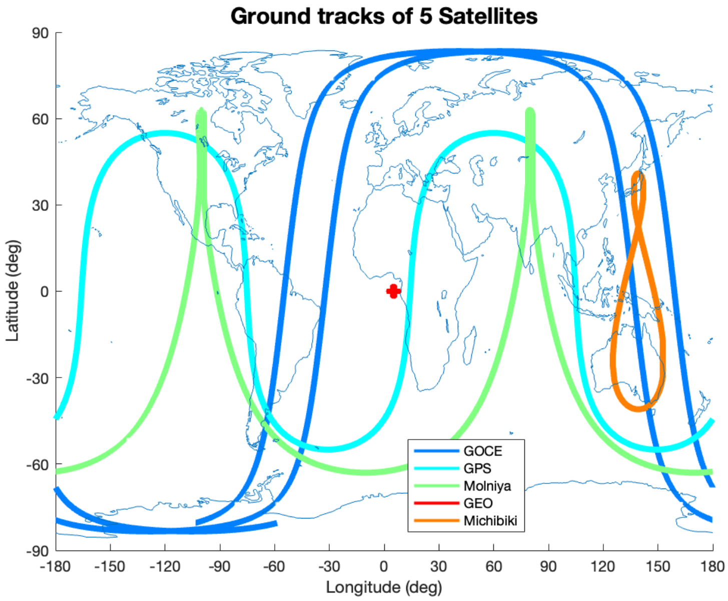

Figure 7: Ground-track

Topocentric system

The position vector of the station Wettzell is given in the Earth-fixed system: $\boldsymbol{r}_w = (4075.53022, 931.78130, 4801.61819)^T$ km. The position vector of the satellite in the topocentric system is calculated by the following equation.

$$\boldsymbol{r}_t = \boldsymbol{r}_e - \boldsymbol{r}_w$$
$$\boldsymbol{r}_t = \boldsymbol{Q}_1 \boldsymbol{R}_2(90 - \Phi_w)\boldsymbol{R}_3(\lambda_w)\boldsymbol{r}_t$$
$$Q_1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The Q_1 matrix is used to transfrom from right- to left-handed system.

It's also useful to compute the azimuth and elevation of the satellites in the topocentric system. The azimuth and elevation are calculated by the following equation.

$$\text{azimuth} = \arctan\left(\frac{y_t}{x_t}\right)$$
$$\text{elevation} = \arctan\left(\frac{z_t}{\sqrt{x_t^2 + y_t^2}}\right)$$

And now we can plot the trajectory of the satellites that can be observed by Wettzell. The trajectory of the satellites that can be observed by Wettzell is shown in Figure 8.

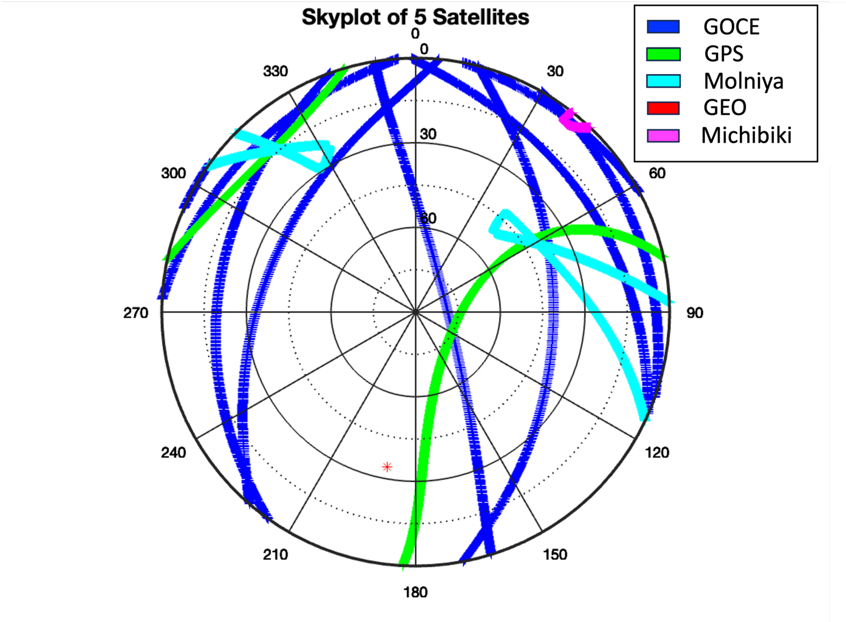

Figure 8: Topocentric system

Finally we can calculate the visibility of the satellites at the station Wettzell. The visibility of the satellites at the station Wettzell is shown in Figure 9.

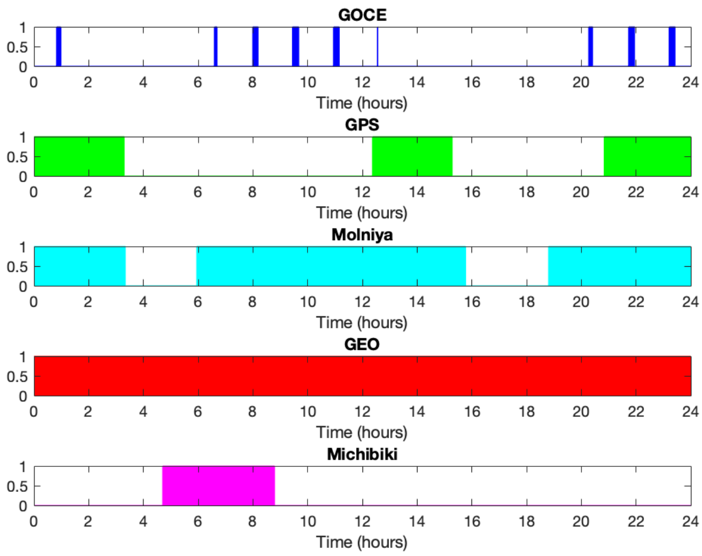

Figure 9: Visibility

Source code

```
function [r,nu,GM]=kep2orb(a,e,t,T)
    % kep2orb(a,e,t,T,Omega,omega)
    %
    % Using 6 kepler elements to calculate the polar coordinates of
    % a satellite and true anomaly.
    % In this case will use time since perigee passage as input,
    % instead of mean anomaly.
    %
    % IN:
    % 4 kepler elements a,e,T
    % a(m) semi-major axis
    % e eccentricity
    % t(s) time since perigee pasasage
    % T(s) time through perigee
    % I(rad) inclination
    %
    % OUT:
    % r radius(m)
    % nu true anomaly(rad)
    % ==========================================================
    % author:           Hsin-Feng Ho
    % Martikelnummer:   03770686
    % created at:       26.11.2023
    % last modification:26.11.2023
    % project:          Exercise 1: Keplerian Orbits
    % ==========================================================

    % gravitational cosntant
    GM=3.986005e14;
    % calculate the mean motion
    n=sqrt(GM./a.^3);
    % calculate the mean anomaly
    M=n.*(t-T);
    % iterate to calculate the eccentric anomaly
    E = M;
    delta = 1;
    while abs(delta)>10^-10
        E_new = M+e.*sin(E);
        delta = E_new-E;
        E = E_new;
    end
    % calculate the true anomaly
    nu=atan2(sqrt((1+e)./(1-e)).*tan(E./2),1)*2;
    % calculate the radius
    r=a.*(1-e.*cos(E));
end
```

kep2orb.m

```
function [ri,ri_dot]=kep2cart(a,e,t,T,I,Omega,omega)
```

```
    % kep2cart(a,e,t,T,I,Omega,omega)
    %
    % Using kepler elements and time to calculate the cartesian
       coordites of
    % a satellite and its velocities in cartesian coordinates.
    %
    %
    % IN:
    % Kepler elements a,e,I,Omega,omega
    % a(m) semi-major axis
    % e eccentricity
    % I(rad) inclination
    % Omega(rad) right ascension of the ascending node
    % omega(rad) argument of perigee
    %
    % t(s) time to compute the satellite position
    % T(s) time of perigee
    %
    % OUT:
    % ri(m) 3-dimensional cartesian coordinates in inertial frame
    % ri_dot(m/s) 3-dimensional cartesian velocities in inertial
       frame
    % ===========================================================
    % author:           Hsin-Feng Ho
    % Martikelnummer:   03770686
    % created at:       26.11.2023
    % last modification:26.11.2023
    % project:          Exercise 1: Keplerian Orbits
    % ===========================================================

    % calculate the polar coordinates and true anomaly
    [r,nu,GM]=kep2orb(a,e,t,T);

    % calculate the coordinates and velocities in orbit reference
       frame
    rf=[r.*cos(nu);r.*sin(nu);zeros(1,length(r))];
    rf_dot=[-sqrt(GM./a./(1-e.^2)).*sin(nu);
    sqrt(GM./a./(1-e.^2)).*(e+cos(nu));zeros(1,length(r))];

    % transform to inertial frame
    ri=MatRot(-Omega,3)*MatRot(-I,1)*MatRot(-omega,3)*rf;
    ri_dot=MatRot(-Omega,3)*MatRot(-I,1)*MatRot(-omega,3)*rf_dot;
end
```

kep2cart.m

```
function [re,re_dot]=cart2efix(r,r_dot,t)
    % cart2efix(r,r_dot,t)
    %
    % Using cartesian coordinates and time to calculate the
       cartesian
    % coordites of a satellite and its velocities in cartesian
```

```
        coordinates
    % in Earth-fixed system
    %
    %
    % IN:
    % cartesian position(s) and velocity(s) of a satellite in
        inertial frame
    % and its corresponding time.
    %
    % r(m) 3 dimensional vector cartesian coordinates
    % r_dot(m/s) 3 dimensional vector cartesian velocities
    % t(s) corresponding time of the satellite
    %
    %
    % OUT:
    % re(m) 3-dimensional cartesian coordinates in Earth-fixed
        frame
    % re_dot(m/s) 3-dimensional cartesian velocities in
        Earth-fixed frame
    % ===========================================================
    % author:           Hsin-Feng Ho
    % Martikelnummer:   03770686
    % created at:       26.11.2023
    % last modification:26.11.2023
    % project:          Exercise 1: Keplerian Orbits
    % ===========================================================

    % Earth rotation rate
    Omega_dot=2*pi/86164;

    % rotation angle
    theta=Omega_dot.*t+3/12*pi;

    re=zeros(size(r,1),size(r,2));
    re_dot=zeros(size(r_dot,1),size(r_dot,2));
    % transformation in Earth fixed system
    for i=1:length(t)
        re(:,i)=MatRot(theta(i),3)*r(:,i);
        re_dot(:,i)=MatRot(theta(i),3)*r_dot(:,i);
    end

end
```

cart2efix.m

```
function [rt,rt_dot,az,elev]=efix2topo(r,r_dot)
    % efix2topo(r,r_dot)
    %
    % Using cartesian coordinates and velocities in Earth-fixed
    % system to calculate the cartesian coordites of a satellite
    % and its velocities in cartesian coordinates in topocentric
    % system. In addition to that, the azimuth and elevation of
```

```
% the satellite are also calculated.
%
%
% IN:
% cartesian position(s) and velocity(s) of a satellite
% in Earth-fixed frame
%
%
% r(m) 3 dimensional vector cartesian coordinates
% r_dot(m/s) 3 dimensional vector cartesian velocities
%
%
% OUT:
% rt(m) 3-dimensional cartesian coordinates in topocentric
   frame
% rt_dot(m/s) 3-dimensional cartesian velocities in
   topocentric frame
% az(rad) azimuth of the satellite
% elev(rad) elevation of the satellite
% =============================================================
% author:            Hsin-Feng Ho
% Martikelnummer:    03770686
% created at:        02.01.2024
% last modification:02.01.2024
% project:           Exercise 1: Keplerian Orbits
% =============================================================

% translation
rw=[4075.53022,931.78130,4081.61819]'*1000; % position vector
   in efix for the station Wettzell
r_trans=r-rw;

% latitude and longitude of the station Wettzell
% a=6378137;
% f_1=298.257223563;
% f=1/f_1;
% b=a-a*f;
% e=sqrt(a^2-b^2)/a;


% transformation from cartesian to ellipsoidal coordinates
% [lambda,phi,~]=cart2ell(rw(1),rw(2),rw(3),a,e);
% lambda=lambda*pi/180;
% phi=phi*pi/180;
lambda=12.8781/180*pi;
phi=49.1449/180*pi;
% rotation
Q1=[-1,0,0;0,1,0;0,0,1]; % from right to left-handed system
rt=Q1*MatRot(pi/2-phi,2)*MatRot(lambda,3)*r_trans;
rt_dot=Q1*MatRot(pi/2-phi,2)*MatRot(lambda,3)*r_dot;
az=atan2(rt(2,:),rt(1,:));
```

```
    elev=atan2(rt(3,:),sqrt(rt(1,:).^2+rt(2,:).^2));
end
```

efix2topo.m