Assignment of

Principles of Navigation

《导航原理》作业

（惯性导航部分 2020 春）

My Chinese Name	尉前进
My Class No.	1704104
My Student No.	1170400423

The report is to contain:

1. Description of the tasks – the next page.
2. The code of your programs, and their explanation.
3. The results of your computation or simulation (as listed by the requirement).
4. Your analysis of the result, and your reflection on the programming or simulation
5. Originality statements or reference/assistance acknowledgements.

English writing is encouraged, though Chinese is also accepted.

Assignment 2

In a flight mission, a bomber accelerates and takes off from the runway of an airport located at 24.27° NL and 118.05° EL, which is 120m above the sea level.

Before it accelerates, the bomber stays level on the runway, with only a heading angle of -25°, towards northest.

The bomber is equipped with a strapdown INS whose three gyros, G_X, G_Y, G_Z, and three accelerometers, A_X, A_Y, A_Z, are installed along the axes X_b, Y_b, Z_b of its body frame respectively.

The strapdown INS starts to work at the moment the bomber starts its runway acceleration. The outputs of the gyros and accelerometers are both pulse numbers. Each gyro pulse is an angular increment of 0.01 *arcsec*, and each accelerometer pulse is $1e^{-7}g_0$, with $g_0 = 9.780327 m/s^2$. The gyro output frequency is 500*Hz*, and the accelerometer's is 25*Hz*. The outputs of the gyros and accelerometers during the flight mission are stored in MATLAB data files named **HB2020.mat**, containing 3-column data arrays ***GMM*** from gyros and ***AMM*** from accelerometers. Ten rows of each array are indicated in the tables, in which each row is the outputs of the sensor at a sampling time.

G_X	G_Y	G_Z
2710	5	10
2710	5	10
2710	5	10
2710	5	10
2710	5	10
2516	4	20
2516	4	20
2516	4	20
2516	4	20
2516	4	20

A_X	A_Y	A_Z
-38595	1344694	9771114
-38647	1344601	9771045
-38699	1344508	9770976
-38752	1344415	9770907
-38804	1344322	9770838
-38857	1344229	9770769
-38909	1344137	9770700
-38962	1344044	9770631
-39014	1343951	9770562
-39066	1343859	9770493

The Earth is seen as an ideal sphere, with radius $R = 6371.00km$ and spinning rate $\omega_{ie} = 7.292 \times 10^{-5}$ *rad/s*, The errors of the gyros and accelerometers are ignored, but the effect of height and latitude on the magnitude of gravity has to be considered. The gravity acceleration can be approximately computed using the formula:

$$g_h = g_0[1 + 0.0053024\sin^2\varphi - 0.0000058\sin^2(2\varphi)] - 3.086 \times 10^{-6} h$$

where $g_0 = 9.780327$ m/s², φ is the local latitude (rad), and h is the current height above sea level (m).

Besides, the influence of height on the angular rates of the geographical frame and the changing rates of latitude and longitude should also be considered.

Velocity, position and the geographical frame can be updated every 0.04s, within which the attitude of the bomber is to be updated 20 times (for 1-S algorithm).

You are required to:

(1) Compute the final attitude quaternion, longitude, latitude, height, and east, north, vertical velocities of the bomber.

(2) Draw the latitude-versus-longitude trajectory of the bomber, with horizontal longitude axis.

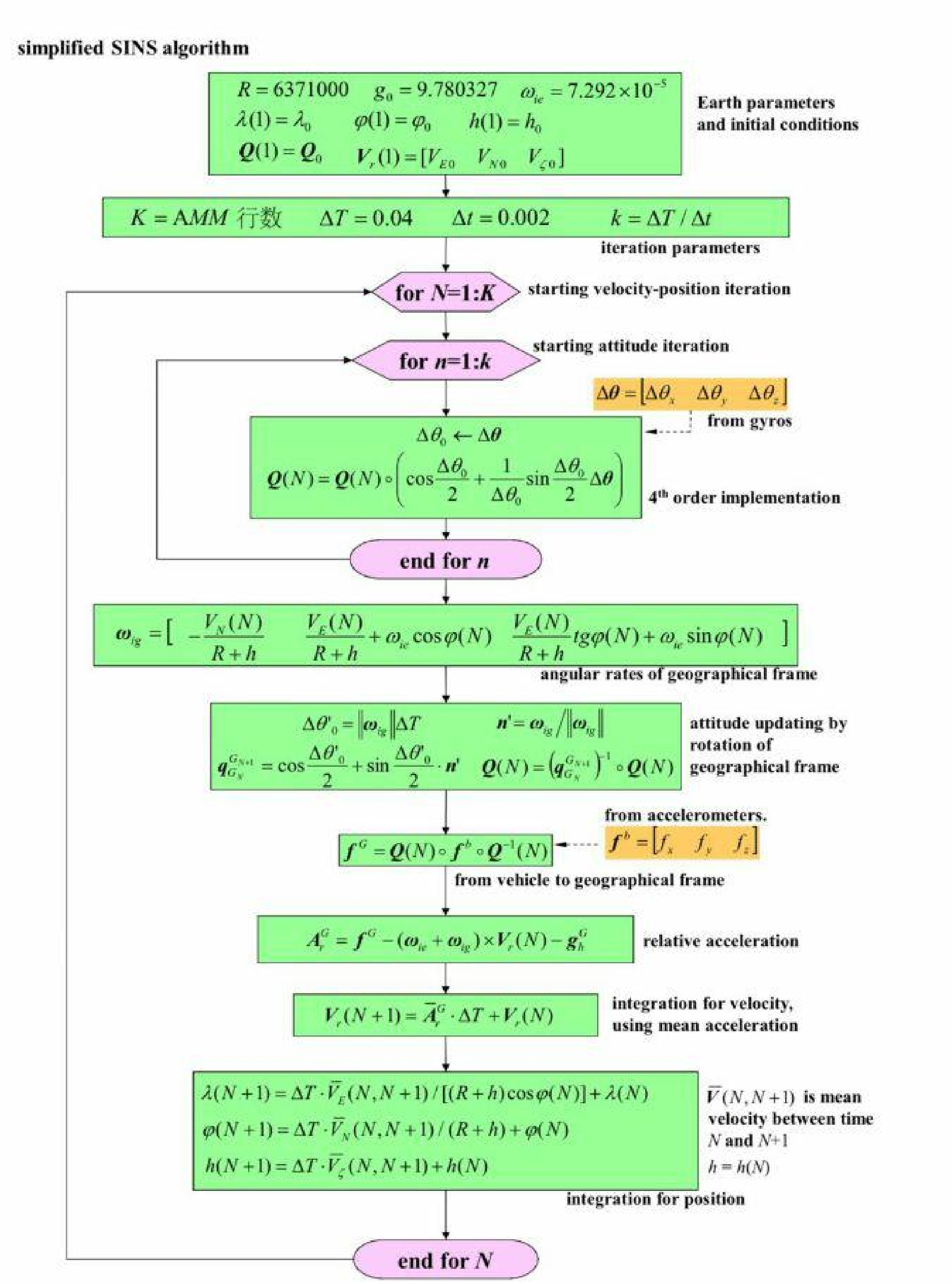
(3) Draw the curve of the height of the bomber with horizontal time axis.

(4) Draw the curves of the attitude angles of the bomber, with horizontal time axis.

More analysis or explorations are encouraged.

2. Analysis

The initial latitude, longitude, altitude and attitude of the aircraft are known. According to the data measured by the gyroscope and accelerometer, and considering the influence of gravity acceleration and earth rotation, the attitude of the aircraft is updated every 0.002s by quaternion, and the geographic coordinate system is updated every 0.04s. The Picca solution of the angular increment algorithm is approximated by the fourth order, and the influence of altitude and latitude on the gravitational acceleration is taken into account.



3.Program Code and Interpretation

%Data initialization

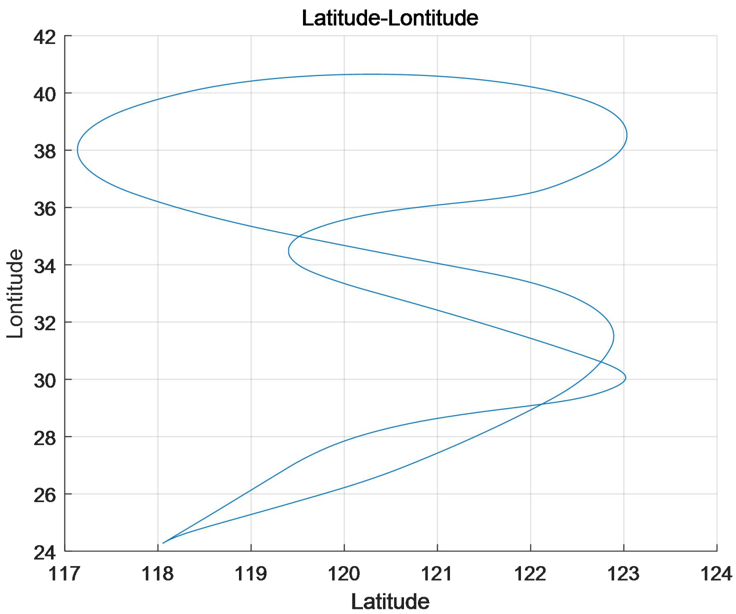
TN: Total time required for aircraft operation; R: arth radius; wie: Earth rotation angular velocity; dT: the cycle of coordinate updating outer cycle; dt: the cycle of coordinate updating inner cycle; K: the number of updates per second; Gyro_pulse: angular increment of gyro pulse; Acc_pulse: pulse of accelerometer; g: acceleration of gravity on the earth's surface.

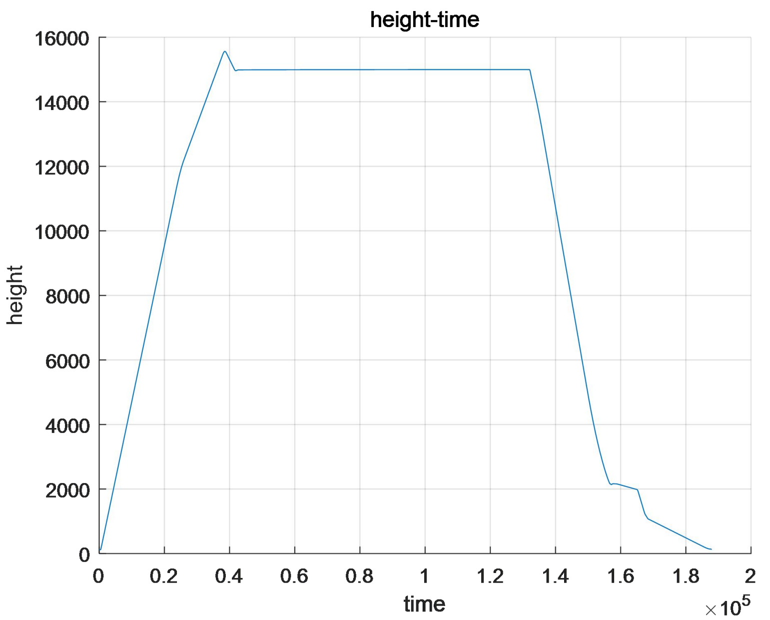
```
%Data initialization
%TN: Total time required for aircraft operation; R: arth radius; wie: Earth
rotation angular velocity; dT: the cycle of coordinate updating outer cycle;
dt: the cycle of coordinate updating inner cycle; K: the number of updates per
second; Gyro_pulse: angular increment of gyro pulse; Acc_pulse: pulse of
accelerometer; g: acceleration of gravity on the earth's surface.
TN=7516;
R=6371000;
wie=7.292e-5;
dT=0.04;
dt=0.002;
K= TN /dT;
k=dT/dt;
g=9.780327;
Gyro_pulse=0.01/3600/180*pi;
Acc_pulse=9.780327/10000000;

%Q: Quaternions representing attitude transformation for each large cycle
Q=zeros(K+1,4);
% Quaternion Initial Value
e0=-25*pi/180;
Q(1,:)=[cos(e0/2),0,0,sin(e0/2)];
% Longitudinal, Latitude and Height Values for Each Large Cycle
LonM=zeros(1,K+1);
LatM=zeros(1,K+1);
HtM=zeros(1,K+1);
% Initial value of location information
LonM(1)=118.05;
LatM(1)=24.27;
HtM(1)=120;
% Define the specific force matrix, initial 0
fx=zeros(1,K+1);
fy=zeros(1,K+1);
fz=zeros(1,K+1);
% Define the velocity matrix in all directions, initially 0.
Ve=zeros(1,K+1);
Vn=zeros(1,K+1);
Vu=zeros(1,K+1);
% Define attitude matrix as zero matrix
heading=zeros(615801,1);
pitch=zeros(615801,1);
```

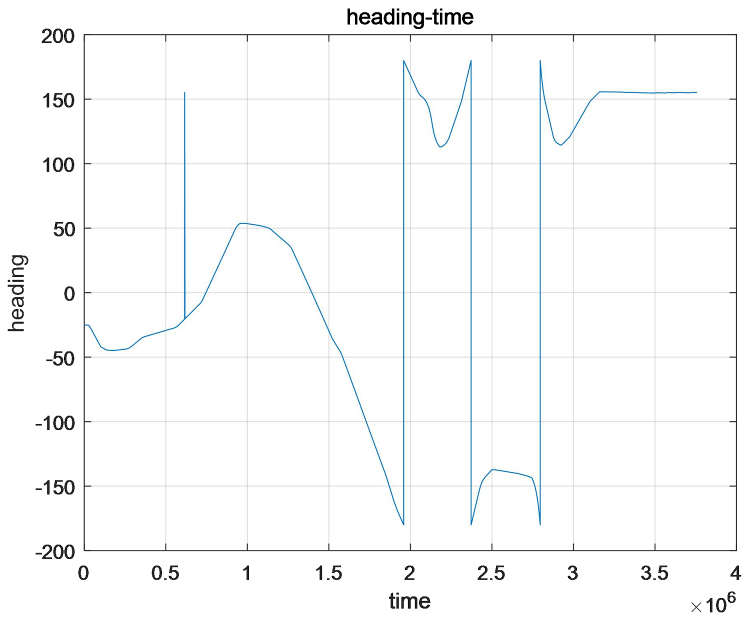
```
roll=zeros(615801,1);
% Data measured by loading gyroscope and accelerometer
load('HB2020.mat')
AR0=[0 0 0];
for N=1:K        % Geographic coordinate system updating
    q=zeros(k+1,4);
    q(1,:)=Q(N,:);
    for n=1:k           % Attitude updating
        w=Gyro_pulse*GMM((N-1)*k+n,:);  % Take the output angle increment of
gyroscope in small cycle
        w_mod=norm(w);
        S=1/2-w_mod^2/48;
        C=1-w_mod^2/8+w_mod^4/384;
        q(n+1,:)=quatmultiply( q(n,:),[C S*w] );  % Fourth-Order Approximate
Picca Solution of Angular Incremental Algorithms
        [h,p,r]=quat2angle(q(n,:));    % Converting Quaternion into Euler Angle
by Quadrangle Function
        heading((N-1)*k+n,1)=h/pi*180;
        pitch((N-1)*k+n,1)=p/pi*180;
        roll((N-1)*k+n,1)=r/pi*180;
    end  % End the Inner Attitude Renewal Cycle
Q(N+1,:)=q(n+1,:); % Quaternion at End of Inner Attitude Updating Cycle
% Perfect the transformation of quaternions into Euler angles
    [hn,pn,rn]=quat2angle(Q(K+1,:));
    heading(615801,1)=hn/pi*180;
    pitch(615801,1)=pn/pi*180;
    roll(615801,1)=rn/pi*180;
% Two-step attitude determination in strapdown inertial navigation system
Wie=[0 wie*cos(LatM(N)/180*pi) wie*sin(LatM(N)/180*pi)];
WE=-Vn(N)/(R+HtM(N));
    WN=Ve(N)/(R+HtM(N))+wie*cos(LatM(N)/180*pi);
WU=Ve(N)/(R+HtM(N))*tan(LatM(N)/180*pi)+wie*sin(LatM(N)/180*pi);
gh=g*(1+0.0053024*(sin(LatM(N)/180*pi))^2-0.0000058*sin(2*LatM(N)/180*pi)^2)
-(3.086e-6)*HtM(N);  % Gravity acceleration affected by altitude and latitude
    Wig=[WE WN WU];
    Wig_mod=norm([WE,WN,WU]);
    n=Wig/Wig_mod;
    Qg=[cos(Wig_mod*dT/2),sin(Wig_mod*dT/2)*n];
    Q(N+1,:)=quatmultiply(quatinv(Qg),Q(N+1,:)); % Quaternions Updated in
Geographic Coordinate System
    % Take the specific force data measured by accelerometer
    fb=Acc_pulse*AMM(N,:);
    f1=quatmultiply(Q(N+1,:),[0,fb]);
    fg=quatmultiply(f1,quatinv(Q(N+1,:)));
    fx(N)=fg(2);
    fy(N)=fg(3);
```

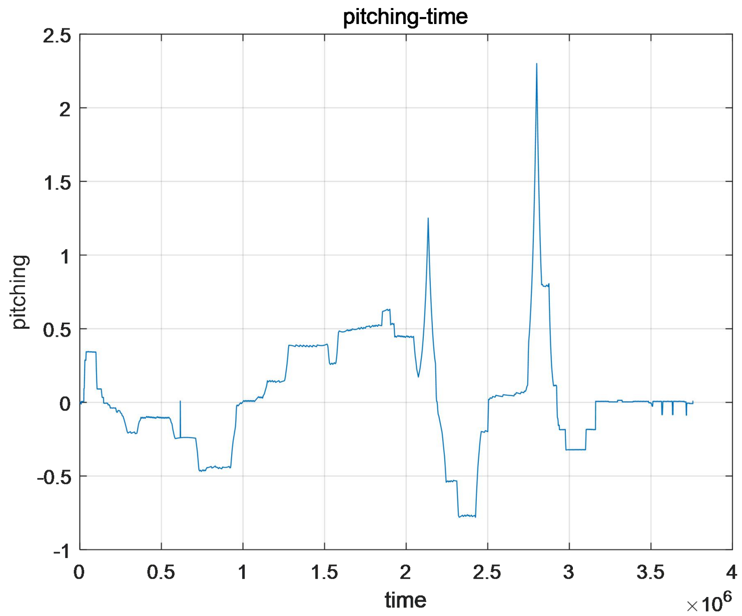
```
    fz(N)=fg(4);
% Compensation for harmful acceleration
AR=[fx(N) fy(N) fz(N)] - cross( Wie+Wig, [Ve(N) Vn(N) Vu(N)]) - [ 0 0 gh];
% Velocity solution
Ve(N+1)=(AR(1)+AR0(1))/2*dT+Ve(N);
Vn(N+1)=(AR(2)+AR0(2))/2*dT+Vn(N);
Vu(N+1)=(AR(3)+AR0(3))/2*dT+Vu(N);
AR0=AR;
% Update location information
LatM(N+1)=(Vn(N)+Vn(N+1))/2*dT/(R+HtM(N))/pi*180+LatM(N);
LonM(N+1)=(Ve(N)+Ve(N+1))/2*dT/((R+HtM(N))*cos(LatM(N)/180*pi))/pi*180+LonM
(N);
HtM(N+1)=(Vu(N)+Vu(N+1))/2*dT+HtM(N);
end  % End the Outer Geographic Coordinate System Renewal Cycle

HPRatt=[heading,pitch,roll];

% Drawing Longitudinal and Latitude Images
figure(1);
title('Latitude-Lontitude');
xlabel('Latitude');
ylabel('Lontitude');
grid on;
hold on;
plot(LonM,LatM);
% Drawing Height Images
figure(2);
title('height-time');
xlabel('time');
ylabel('height');
grid on;
hold on;
plot((0:K),HtM);
% Drawing Attitude Angle Change Map
figure(3);
plot(heading);
xlabel('time');
ylabel('heading');
title('heading-time');
grid on;
figure(4);
plot(pitch);
xlabel('time');
ylabel('pitching');
title('pitching-time');
grid on;
```

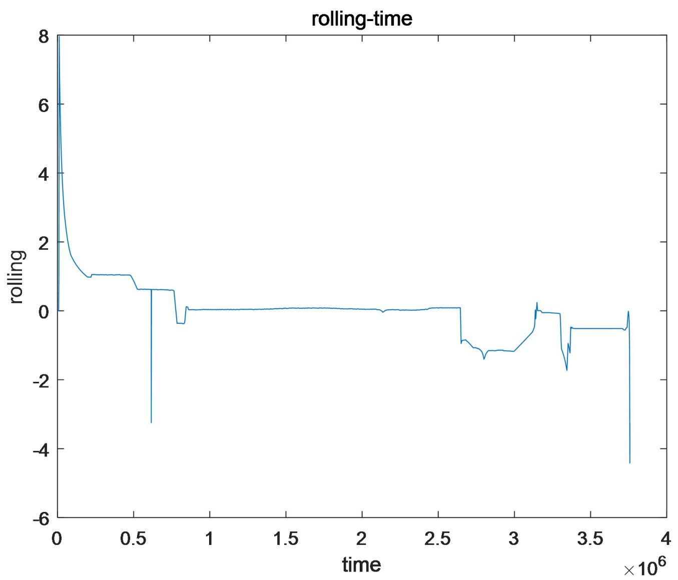
```
figure(5);
plot(roll);
xlabel('time');
ylabel('rolling');
title('rolling-time');

% Output final quaternion, velocity in all directions and aircraft position
Q(K+1,:);
LatM(K+1);
LonM(k+1);
HtM(K+1);
Ve(K+1);
Vn(K+1);
Vu(K+1);
save FlightData HPRatt LonM LatM HtM
```






heading-time
heading
time
×10^6


pitching-time
pitching
time
×10^6


rolling-time
rolling
time
×10^6

4. Key benefits

This large operation let me know and master the algorithm of SINS, and preliminarily grasp the simulation calculation of SINS with MATLAB, which has further improved the use of MATLAB and benefited a lot.

5. Job statement

This assignment refers to YuHan Liu's procedure and got a lot of help from YuHan Liu.