

```
%% Attitude Determination using TRIAD
%% Mohamed Temam Nasri
%% Matlab Implementation
%% University of Manitoba
%% Advisor: Prof. Witold Kinsner
%% October 23, 2012
%% Version 2.6
```

```
clc
clear all
```

```
load sun_inertial_00001.csv
load igrf_inertial_00001.csv
load sun_005_body_00001.csv
load igrf_005_body_00001.csv
load quaternion_005_00001.csv
```

```
tfile = 'dtusat1.TLE';
tle = readTLE(tfile);
[epoch_yr, epoch_m, epoch_d, epoch_h, epoch_min, epoch_s] = days2ymdhms(tle.epoch_year,
tle.epoch_day);
Ttdb = cal_Ttdb(epoch_yr, epoch_m, epoch_d, epoch_h, epoch_min, epoch_s);
```

```
jd1= juliandate(epoch_yr, epoch_m, epoch_d, epoch_h, epoch_min, epoch_s);
```

```
R=[];B=[];A=[];S=[];P=[];
co_var=[];co_var1=[];co_var2=[];
q1=[];
q2=[];
q3=[];
q4=[];
t=[0:.01:20];
theta_a=[];
si_a=[];
phi_a=[];
t_sec=[];
theta=[];
angle_sun=[];
angle_mag=[];
```

```
sigma1 = 0.1; % Sun Sensor Noise(Unit Vector)
sigma2 = 0.1; % Magnetometer Noise (Unit Vector)
```

```
q_4=[];
```

```

S1=[-0.655571      0.752337      0.064924];
S2=[-0.389829  -0.430934      0.813836];

R1=[ -0.969339   -0.225445   -0.097755];
R2=[-0.225105    0.955641    0.189944];

sun_true=[];
sun_meas=[];
mag_true=[];
mag_meas=[];

%*****magnetic field*****
%IGRF model initializations 21001.5
[G,H] = IGRF2005;      % IGRF coefficients for 2005
nmax = 13;             % max degree of geopotential
mmax = 13;             % max order of geopotential
Kschmidt = schmidt(nmax,mmax);

for i=0:1:5000

    t_sec=[t_sec; i*0.0144];
    delta_sim=i*(0.0144);
    jd=jd1+(i*0.00001);

    R1=[sun_inertial_00001(i+1,2) sun_inertial_00001(i+1,3) sun_inertial_00001(i+1,4)];

    R1=R1/(sqrt(sun_inertial_00001(i+1,2)^2+sun_inertial_00001(i+1,3)^2+sun_inertial_00001(i+1,4)^2));

    R2=[igrf_inertial_00001(i+1,1) igrf_inertial_00001(i+1,2) igrf_inertial_00001(i+1,3)];

    %R1=rsun1;
    %R2=b_E_T;

    S1=[sun_005_body_00001(i+1,2) sun_005_body_00001(i+1,3)
sun_005_body_00001(i+1,4)];
    S1=S1/sun_005_body_00001(i+1,5);
    S2=[igrf_005_body_00001(i+1,6) igrf_005_body_00001(i+1,7)
igrf_005_body_00001(i+1,8)];

    %***** Adding the Noise Values to body Frame *****
    NOISE1 = sigma1*(eye(3,3)-S1'*S1)*randn([3,1]);
    NOISE2 = sigma2*(eye(3,3)-S2'*S2)*randn([3,1]);
    % NOISE1 = sigma1*randn([3,1]);

```

```
S1N=S1+NOISE1';
S2N=S2+NOISE2';
```

```
sun_true=[sun_true; S1];
sun_meas=[sun_meas; S1N];
mag_true=[mag_true; S2];
mag_meas=[mag_meas; S2N];
```

```
R=[R1; cross(R1,R2)/norm(cross(R1,R2)); cross(R1,cross(R1,R2)/norm(cross(R1,R2))) ];
S=[S1N; cross(S1N,S2N)/norm(cross(S1N,S2N))
cross(S1N,cross(S1N,S2N)/norm(cross(S1N,S2N))) ]';
```

```
A=S*R;
[qs]=Rotation_to_Quaternion(A);
if(qs(4)<0)
    qs=-qs;
end
```

```
[e1,e2,e3,a]=Quaternion_to_Axis(qs(1),qs(2),qs(3),qs(4));
```

```
qt=[quaternion_005_00001(i+1,2) quaternion_005_00001(i+1,3)
quaternion_005_00001(i+1,4) quaternion_005_00001(i+1,5)];
```

```
qe=[ qt(4) qt(3) -qt(2) qt(1);
    -qt(3) qt(4) qt(1) qt(2);
    qt(2) -qt(1) qt(4) qt(3);
    -qt(1) -qt(2) -qt(3) qt(4)]*[-qs(1);
    -qs(2);
    -qs(3);
    qs(4)];
```

```
qe(4)=sqrt(1-qe(1)^2-qe(2)^2-qe(3)^2);
```

```
if(qe(4)<0)
    qe=-qe;
```

```
end
```

```
[e1,e2,e3,a]=Quaternion_to_Axis(qe(1),qe(2),qe(3),qe(4));
```

```
q_4=[q_4; qe(4)];
a=a*180/pi;
```

```
theta=[theta;a];
```

```
ang=acos(dot(S2,S1));  
angle_sun=[angle_sun; ang*180/pi];
```

```
ang=acos(dot(R2,R1));  
angle_mag=[angle_mag; ang*180/pi];
```

```
end
```

```
er_TRIAD=0;
```

```
for j=1:1:length(t_sec)  
    er_TRIAD=er_TRIAD+theta(j);  
end
```

```
er_TRIAD=er_TRIAD/length(t_sec)
```

```
figure(3)  
subplot(2,1,1)  
plot(t_sec,theta)  
xlabel('Minutes')  
ylabel('Error Angle')
```

```
figure(4)  
subplot(3,1,2)  
plot(t_sec,mag_meas(:,2),'.');  
hold on  
plot(t_sec,mag_true(:,2),'r');  
xlabel('Minutes')  
ylabel('Unit Y')
```

```
subplot(3,1,3)  
plot(t_sec,mag_meas(:,3),'.');  
plot(t_sec,mag_true(:,3),'r');  
hold on  
xlabel('Minutes')  
ylabel('Unit Z')
```

```
figure(4)
subplot(3,1,1)
plot(t_sec,sun_meas(:,1),'');
hold on
plot(t_sec,sun_true(:,1),'r');
xlabel('Minutes')
ylabel('Unit X')
```

```
subplot(3,1,2)
plot(t_sec,sun_meas(:,2),'');
hold on
plot(t_sec,sun_true(:,2),'r');
xlabel('Minutes')
ylabel('Unit Y')
```

```
subplot(3,1,3)
plot(t_sec,sun_meas(:,3),'');
hold on
plot(t_sec,sun_true(:,3),'r');
xlabel('Minutes')
ylabel('Unit Z')
```

```
figure(5)
plot(t_sec,theta)
hold on
plot(t_sec,er_TRIAD,'--')
xlabel('Time [Min]')
ylabel('Error Angle [Deg]')
axis tight
```

```
legend('Error','Average', 'Location','NorthEast')
set(gcf, 'paperunits', 'centimeters', 'paperposition', [0 0 15 5])
print -dtiff -r300 ErrAng_TRIAD_deg.png
```