clear, clc

load('atti06\_h.mat');

format long

DEL\_TIME = 1000.0;

FNLEN = 256;

now = clock;

tic

%%% Lee

znum = 0;

zsum = 0;

% INPUT : Gyroscope

gyro = fopen("D:\StarTracker-20230706T212348Z-001\StarTracker\02\_Check\_structure\data\g\_rate\_lrs.dat", 'r');

% INPUT : LRS STAR

sttr = fopen("D:\StarTracker-20230706T212348Z-001\StarTracker\02\_Check\_structure\data\FOVs.dat", 'r');

% INPUT : INITIAL ATTITUDE

msg = "this is dumb";

% OUTPUTS

outi = fopen('inertials.dat', 'w');

outr = fopen('rotatings.dat', 'w');

rate = fopen('gyrorate.dat', 'w');

outM = fopen('ckg\_M.dat', 'w');

out1a = fopen('ckg\_1a.dat', 'w');

outpos = fopen('glas\_pos.dat', 'w');

out1b = fopen('ckg\_1b.dat', 'w');

outo = fopen('observed.dat', 'w');

out2 = fopen('ckg\_2.dat', 'w');

outds = fopen('rmsstar.dat', 'w');

outdq = fopen('rmsckg.dat', 'w');

outn = fopen('IDed\_num.dat', 'w');

out3 = fopen('ckg\_3.dat', 'w');

gbias = fopen('ckg\_gbias.dat', 'w');

outgbias = fopen('gbias.average', 'w');

angvel = fopen('ckg\_angvel.dat', 'w');

outmag = fopen('star\_mag.dat', 'w');

outdist = fopen('dm\_dist.dat', 'w');

qc = fopen('ist\_qc.dat', 'w');

merge = fopen('merge.dat', 'w');

ixy = fopen('ided\_xy.dat', 'w');

t\_diff = fopen('t\_diff.dat', 'w');

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

% load all data in datafile in memory area \*/

load('star\_data.mat')

id\_star = zeros(Ns,1);

id\_body = zeros(Ns,1);

bore\_body = T\_B(:,3); %(0,0,1) in the ST frame

ccdstep = T\_STEP\_CCD;

rmssum = zeros(3,1);

rmslsum = zeros(3,1);

b\_add = zeros(3,1);

n\_add = 0 ;

n\_g = 0 ;

nsig = 0 ;

rg = 0 ;

consec = 0;

crf\_count = 0 ;

cnt = 0 ;

cnt0 = 0 ;

ID\_loca = 'F' ;

id\_ = 99 ; % Failed as initial: Used actually?

total\_mstar = 0.0 ;

total\_ided = 0.0 ;

sign1 = 0;

QC1 = 0;

M = 0;

g\_angle = zeros(4,1);

gt\_old = 0.0 ;

% KF (t < KF) : in opt. bench coordinate frame

w0 = [2.0\*pi/T/10.0 + 1e-4, 1.e-4, 1.e-4];

inival = dlmread('q\_initial.txt');

cit = inival(1,1);

q = inival(2,:);

b\_est\_average = (0.01/3600\*pi/180)\*ones(1,3);

q = q/norm(q);

q0 = q;

qq\_old = q;

qq = zeros(4,1);

P\_theta = zeros(3);

P = diag([ 1e-6 1e-6 1e-6 1e-8 1e-8 1e-8 ]);

b\_star\_prev = 0;

crf\_prev = 0;

rtn\_prev = 0;

% READ STAR DATA

ccd\_time = -2000.0;

load('measurement\_data.mat'); % Data file containing FOVs\_meas, gyro\_meas, q0, t0

nstar = 0;

FOVs\_count = 1;

while (nstar < 3)

[ccd\_time, x, y, xmag, m\_star, nstar] = read\_CCD0(FOVs\_meas, Ns, FOVs\_count);

FOVs\_count = FOVs\_count + 1;

end

% READ GYRO DATA %

gyro\_count = 1;

[t\_gyro, w, u, gread] = read\_gyro(b\_est\_average, gyro\_meas, gyro\_count);

gyro\_count = gyro\_count +1;

if gread == 1

fprintf(' No more gyro data\n');

end

while ( (t\_gyro - ccd\_time) > 0.05)

[ccd\_time, x, y, xmag, m\_star, nstar] = read\_CCD(ccd\_time,FOVs\_meas, Ns);

if nstar > Ns

fprintf('Too many stars at t=%8.2f\n', ccd\_time);

elseif nstar == -999

fprintf('Empty IST data file before "while" loop\n') ;

else

total\_mstar = total\_mstar + nstar ;

end

FOVs\_count = FOVs\_count + 1;

end

while ( (ccd\_time - t\_gyro) > 0.05)

[t\_gyro, w, u, gread] = read\_gyro(b\_est\_average, gyro\_meas, gyro\_count);

if gread == 1

fprintf(' Empty gyro data file at ccd\_time = %8.2f\n', ccd\_time) ;

end

gyro\_count = gyro\_count +1;

end

ccd\_time0 = ccd\_time ;

cctime = ccd\_time + Ctime ; % set up the pm range

t\_rms = t\_gyro ;

if ccd\_time < KF

w = w0;

end

c\_dm = 0;

c\_dm\_ided = 0;

c\_pm = 0;

c\_pm\_ided = 0;

while( ccd\_time <= TIMELIMIT )

d\_i = 0.0 ;

fprintf(t\_diff, '%15.6f %15.6f %15.6f\n', ccd\_time, t\_gyro, ccd\_time-t\_gyro) ;

fprintf('%15.6f %15.6f %15.6f\n', ccd\_time, t\_gyro, ccd\_time-t\_gyro) ;

q0 = q0/norm(q0);

cnt = 0;

if ( ccd\_time < cctime && nstar >= 3)

c\_pm = c\_pm + 1 ;

rtn = 0 ;

if (nstar > 6); nstar = 6 ; end % up to 6 stars for PM \*/

% rtn = starID\_pm(ccd\_time, q0, nstar, &cnt, bore\_body, outi, outr, outo, outmag, ixy, x, y, xmag) ;

[ccd\_time, cnt ,m\_star, BLI, NEWBLI, rtn, b\_star, crf, nstar] = starID\_pm(ccd\_time, q0, nstar, cnt, bore\_body, outi, outr, outo, outmag, ixy, x, y, xmag, adjcell, scell, stars, m\_star, b\_star\_prev, crf\_prev,rtn\_prev);

b\_star\_prev = b\_star;

crf\_prev = crf;

rtn\_prev = rtn;

fprintf("NEWBLI\n") ; % !!PRINT STATEMENT!!<-----------------------------------------------------------------\*/

fprintf("mag id num\n") ;

NEWBLI\_size = size(struct2table(NEWBLI),1);

for(i=1:NEWBLI\_size)

fprintf("%f %d %4d\n",NEWBLI(i).mag,NEWBLI(i).id,NEWBLI(i).num) ;

end

fprintf("\n");

fprintf("BLI\n"); % !!PRINT STATEMENT!!<-----------------------------------------------------------------\*/

fprintf("IBL starnum L mag\n") ;

BLI\_size = size(struct2table(BLI),1);

for(i=1:BLI\_size)

fprintf("%0.4f %4d %0.4f %0.4f %0.4f %4f\n",BLI(i).IBL,BLI(i).starnum,BLI(i).L(1),BLI(i).L(2),BLI(i).L(3), BLI(i).mag) ;

end

fprintf("\n");

fprintf("m\_star\n") ; % !!PRINT STATEMENT!!<-----------------------------------------------------------------\*/

fprintf("L mag\n") ;

m\_star\_size = size(struct2table(m\_star),1);

for(i=1:m\_star\_size)

fprintf("%0.4f %0.4f %0.4f %4f\n",m\_star(i).L(1),m\_star(i).L(2),m\_star(i).L(3),m\_star(i).mag) ;

end

fprintf("\n");

for i=1:1:Ns

x(i) = 0.0;

y(i) = 0.0;

xmag(i) = 0.0 ;

end

if rtn ~= 1

fprintf('starID\_pm() finished abnormally at %10.2f\n',ccd\_time);

end

if (cnt > 0); c\_pm\_ided = c\_pm\_ided + 1 ; end

crf\_count = cnt ;

ID\_loca = 'P' ;

id\_ = 12 ;

if (cnt >= 3 && rg > 0 ) % UPDATE from > 0

cnt0 = 0;

n\_add = 0;

n\_g = 0;

rg = 0 ;

b\_add = zeros(3,1);

end

elseif (ccd\_time >= cctime && nstar > 0 && consec ~= 0)

c\_dm = c\_dm + 1;

[ccd\_time, cnt ,rtn] = starID\_dm(ccd\_time, q0, nstar, cnt, cctime, w, crf\_count, id\_star, id\_body, outi, outr, outo, outmag, outdist, ixy, x, y, xmag, ccd\_time0, ccd\_time, m\_star, crf, stars, T\_B, b\_star, scell2);

for(i=1:BLI\_size)

fprintf("%0.4f %4d %0.4f %0.4f %0.4f %4f\n",BLI(i).IBL,BLI(i).starnum,BLI(i).L(1),BLI(i).L(2),BLI(i).L(3), BLI(i).mag) ;

end

fprintf("\n");

fprintf("m\_star\n") ; % !!PRINT STATEMENT!!<-----------------------------------------------------------------\*/

fprintf("L mag\n") ;

m\_star\_size = size(struct2table(m\_star),1);

for(i=1:m\_star\_size)

fprintf("%0.4f %0.4f %0.4f %4f\n",m\_star(i).L(1),m\_star(i).L(2),m\_star(i).L(3),m\_star(i).mag) ;

end

fprintf("\n");

for i=1:1:Ns

x(i) = 0.0;

y(i) = 0.0;

xmag(i) = 0.0 ;

end

if rtn ~= 1

fprintf('starID\_dm() finished abnormally at %10.2f\n', ccd\_time);

end

if (cnt > 0); c\_dm\_ided = c\_dm\_ided + 1 ; end

crf\_count = cnt ;

ID\_loca = 'D' ;

id\_ = 13 ;

if (cnt > 1) % UPDATE from 0

cnt0 = 0 ;

if (rg > 0)

n\_add = 0;

n\_g = 0;

rg = 0 ;

b\_add = zeros(3,1);

end

end

else

cnt = 0;

end

fprintf('%g %g\n', ccd\_time, TIMELIMIT); %FOR DEBUGGING

total\_ided = total\_ided + crf\_count ; % just for statistics %

if (cnt >= 3)

for i = 1:numel(crf)

for j = 1:numel(crf(i).L)

W(i,j) = crf(i).L(j);

W(i,j+1) = crf(i).mag;

V(i,j) = b\_star(i).L(j);

V(i,j+1) = b\_star(i).mag;

end

end

[qq, P, d\_i, rtn] = q\_method(V, W, nstar, d\_i);

%qq = quest(W(1:3,:),V(1:3,:), ones(nstar,1)/nstar)

rtn = 0;

if (rtn ~= 0)

fprintf(stdout,"q\_method() finished abnormally at %10.2f\n", ccd\_time);

end

% Discontinuity Check

for i = 1:1:3

if(qq(i) > .3 && qq(i)\*qq\_old(i)<0)

opp\_sign=opp\_sign+1;

if(opp\_sign>=2)

sign1=1;

else

sign1=0;

end

end

end

opp\_sign=0;

if (sign1 == 1)

if nsig == 0

nsig = 1;

else

nsig = 0;

end

sign1 = 0;

end

qq\_old = qq;

if nsig == 1

qq = -qq;

end

fprintf(out1a,"%15.6f", ccd\_time);

fprintf(out1a," %15.10f %15.10f %15.10f %15.10f",qq);

fprintf(out1a,"\n");

fprintf(out2,"%15.6f", ccd\_time);

fprintf(out2,"\t12.8f\t%12.8\t%12.8 ", sqrt(4\*P));

% completely diffrent from C, 7/10/23

fprintf(out2,"\n");

if(consec < 9)

consec = consec+1;

if(consec == 9)

%10th consecutive data: (re)start filter

P = zeros(3,3);

% for(i=0;i<3;i++)

% for(j=0;j<3;j++) P[i][j] = P\_theta[i][j] ;

P = P\_theta;

t\_old = ccd\_time;

end

A = q\_to\_A(qq);

qp = q;

q = q0;

q0 = qq;

% for(i=0;i<3;i++) for(j=0;j<3;j++) M[i][j] = 0.0 ;

M = zeros(3,3);

% for(i=0;i<3;i++) for(j=0;j<3;j++) for(k=0;k<3;k++)

% M[i][j] += T\_B[k][i] \* A[k][j] ;

M = M + T\_B\*A;

fprintf(out1b,"%15.6f 11 ", ccd\_time);

fprintf(out1b,"\n");

fprintf(out1b,"\t%15.12f",qq);

fprintf(out1b,"\n");

fprintf(qc,"%15.6f\t%3d\t%6.2f\n",ccd\_time, 1, -0.5);

fprintf(outn,"%15.6f 11 %2d %2d %2d %8.3f\n", ccd\_time, nstar, cnt, consec, 0.0);

fprintf(outM,"%12.6f %3d\n%20.15f %20.15f %20.15f \n%20.15f %20.15f %20.15f \n% 20.15f %20.15f %20.15f \n",...

ccd\_time, QC1, inv(M)) ; %updated 7/27 from M

end

end

if (consec == 10 && cnt ~= 0)

t\_elapse = ccd\_time - ccd\_time0 ; % Tuning adjustment

b\_est = b\_est\_average;

% cuvfout = cuvf(ccd\_time, t\_old, t\_elapse, qq, q, P, w, b\_est, cnt, zsum, znum, ccdstep); I dont like this

[zsum, znum, b\_est, q, t\_old, P] = cuvf(ccd\_time, t\_old, t\_elapse, qq, q, P, w, b\_est, cnt, zsum, znum, ccdstep);

cuvfout = 0;

b\_est\_average = b\_est;

if ccd\_time >= 0.0 % not for me

b\_est\_sum = b\_est\_sum + b\_est;

b\_est\_cnt = b\_est\_cnt+1;

end

if cuvfout ~= 0

fprintf(stdout," cuvf() finished abnormally at %10.2f\n", ccd\_time) ;

error(msg) ;

end

ready\_merge = 0;

if ready\_merge == 1

fprintf(merge," %15.6f %15.6f\n", ccd\_time-dt\_merge/10.0, ccd\_time-1.0) ;

ready\_merge = 0 ;

end

if ccd\_time < KF

w0 = w;

end

qp = q0;

q0 = q;

A = q\_to\_A(qp) ;

M = zeros(3,3);

for i=1:1:3

for j=1:1:3

for k=1:1:3

M(i,j) = T\_B(k,i) \* A(k,j) ;

end

end

end

% if (ccd\_time > cctime) % for QC %dont know 5/4/22

% rmssum = rmssum + zsum;

% rmsnum = rmsnum + znum ;

% rmslsum = rmslsum + zsum;

% rmslnum = rmslnum + znum;

% end

% if (cnt >= 3) annoyed me

% qp\_in = -qp;

% qp\_in(3) = qp(3);

% z\_bar = qcomp(qq,qp\_in,z\_bar) ;

% temp = sqrt(z\_bar(1)\*z\_bar(1) + z\_bar(2)\*z\_bar(2)+z\_bar(3)\*z\_bar(3) + z\_bar(4)\*z\_bar(4)) ;

% z\_bar = z\_bar/temp ;

% z = 2\*z\_bar; % 3 x 1 vector

% if (t\_gyro > cctime) % for QC

% rmszsum = rmszsum + sqrt(z\*z) ;

% rmszscalar = rmszscalar + sqrt(z\_bar(4)\*z\_bar(4)) ;

% rmsznum = rmsznum + 1;

% rmszlsum = rmszlsum + sqrt(z\*z) ;

% rmszlscalar = rmszlscalar + sqrt(z\_bar(4)\*z\_bar(4)) ;

% rmszlnum = rmszlnum + 1;

% end

% end

if ( cnt == 2 ) % || (ID\_loca == 'D' && cnt >= 3) )

d\_i = b\_star(1).L(1)\*b\_star(2).L(1) + b\_star(1).L(2)\*b\_star(2).L(2) + b\_star(1).L(3)\*b\_star(2).L(3);

end

if ( cnt < 2 )

d\_i = 0.0 ;

end

fprintf(gbias, "%15.6f %15.8e %15.8e %15.8e\n", ccd\_time, b\_est\*3600/3.1415\*180);

fprintf(angvel, "%15.6f %15.5f %15.5f %15.5f\n", ccd\_time, w\*180/3.1415\*36000);

OneSig1 = sqrt(P(1,1))\*180./pi()\*3600 ;

OneSig2 = sqrt(P(2,2))\*180./pi()\*3600 ;

OneSig3 = sqrt(P(3,3))\*180./pi()\*3600 ;

fprintf(out3,"%15.6f",ccd\_time) ;

%fprintf(out3," %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f\n", OneSig1, OneSig2, OneSig3, sqrt(P(3,3))\*180/3.1415\*3600,sqrt(P(4,4))\*180/3.1415\*3600,sqrt(P(5,5))\*180/3.1415\*3600);

%nope

atti\_rms = sqrt(OneSig1\*OneSig1+OneSig2\*OneSig2) ;

if atti\_rms <= 2.0

QC1 = 2;

elseif atti\_rms > 2.0 && atti\_rms < 4.0

QC1 = 1;

else

QC1 = 0 ;

end

if (ID\_loca == 'P')

fprintf(out1b,"%15.6f 12 ",ccd\_time) ;

elseif (ID\_loca == 'D')

fprintf(out1b,"%15.6f 13 ",ccd\_time) ;

end

fprintf(out1b," %15.12f",qp);

fprintf(out1b,"\n") ;

fprintf(qc,"%15.6f %3d %6.2f\n",ccd\_time, QC1, atti\_rms) ;

fprintf(outM,"%12.6f %3d\n%20.15f %20.15f %20.15f \n%20.15f %20.15f %20.15f \n% 20.15f %20.15f %20.15f \n", ccd\_time, QC1, inv(M));

%updated 7/27 from M

if (cnt > 2)

fprintf(outn,"%15.6f %2d %2d %2d %2d %8.3f\n", ccd\_time, id\_, nstar, crf\_count, consec, acos(d\_i/cnt)\*180.0/pi());

elseif (cnt == 2)

fprintf(outn,"%15.6f %2d %2d %2d %2d %8.3f\n", ccd\_time, id\_, nstar, crf\_count, consec, acos(d\_i)\*180.0/pi());

else % if (cnt == 1) \*/

fprintf(outn,"%15.6f %2d %2d %2d %2d %8.3f\n",ccd\_time, id\_, nstar, crf\_count, consec, 0.0);

end

end %end of if ( consec == 10 && cnt != 0) \*/

if (cnt == 0)

cnt0 = cnt0 + 1;

if (cnt0 > 100.0 || (cnt0 > 10.0 && nstar >= 3) ) % Changed

cctime = ccd\_time + Ctime ;

consec = 0 ;

QC1 = 0 ; atti\_rms = -1.0 ;

end

if (cnt0 > 600.0)

% fprintf(stdout, " 1. Warning: No obs for %6.2f min at %15.6f\n",cnt0/10.0/60.0, t\_gyro) ; \*/

dt\_merge = cnt0 ;

ready\_merge = 1 ;

QC1 = 0 ; atti\_rms = -2.0 ;

end

[rg, q0] = no\_stars(q0, w) ; % no idea % make no\_stars work, 7/14/23

qp = q0;

A = q\_to\_A(qp) ;

M = zeros(3,3);

for i=1:1:3

for j=1:1:3

for k=1:1:3

M(i,j) = M(i,j) + T\_B(k,i) \* A(k,j);

end

end

end

fprintf(out1b,"%15.6f 21 ", ccd\_time) ; % no star IDed \*/

fprintf(out1b," %15.12f",qp) ;

fprintf(out1b,"\n") ;

%fprintf(qc,"%15.6f %3d %6.2f\n",ccd\_time, QC1, atti\_rms) ; dont have

ID\_loca = 'G' ; id\_ = 21 ; consec = 0 ;

fprintf(outn,"%15.6f %2d %2d %2d %2d %8.3f\n", ccd\_time, id\_, nstar, cnt, consec, 0.0) ;

fprintf(outM,"%12.6f %3d\n%20.15f %20.15f %20.15f \n%20.15f %20.15f %20.15f \n% 20.15f %20.15f %20.15f \n", ccd\_time, QC1, inv(M)) ;

%updated 7/27 from M

end

if (consec == 9 && t\_old == ccd\_time)

consec = 10 ;

end

[ccd\_time, x, y, xmag, m\_star, nstar] = read\_CCD(ccd\_time,FOVs\_meas, Ns);

while nstar == 0

if ( nstar == -999 )

fprintf(stdout,"No more IST data at or after %10.2f\n",ccd\_time) ;

if ( t\_elapse < DEL\_TIME )

error(msg) ;

else

break ;

end

end

[t\_gyro, w, u, gread] = read\_gyro(t\_gyro, u, b\_est\_average, w, gyro) ;

if (gread == 1)

fprintf(stdout,"No more gyro data. IST data may exist at %10.2f\n", t\_gyro) ;

if (t\_gyro < 86400.0)

error(msg) ;

else

fprintf(outgbias, "%15ld %15.5f %15.5f %15.5f\n", b\_est\_cnt, b\_est\_sum/b\_est\_cnt\*3600/PI\*180.0) ;

error(msg) ;

end

end

if (ccd\_time < KF)

w = w0;

end

[rg,q0] = no\_stars(q0, w) ; % change 7/14/23

%rg = no\_stars(ccd\_time, t\_gyro, q0, w) ;% original code %dont have

qp = q0;

q\_to\_A(qp, A) ;

M = zeros(3,3);

for i=1:1:3

for j=1:1:3

for k=1:1:3

M(i,j) = M(i,j) + T\_B(k,i) \* A(k,j);

end

end

end

cnt0 = cnt0 + 1;

if (cnt0 > 100.0 || (cnt0 > 10.0 && nstar >= 3) ) % Changed \*/

cctime = ccd\_time + Ctime ;

consec = 0 ;

QC1 = 0 ; atti\_rms = -1.0 ;

end

if (cnt0 > 600.0)

dt\_merge = cnt0 ;

ready\_merge = 1 ;

QC1 = 0 ; atti\_rms = -2.0 ;

end

fprintf(out1b,"%15.6f 22 ",ccd\_time) ; % No star observed \*/

fprintf(out1b," %15.12f",qp) ;

fprintf(out1b,"\n") ;

ID\_loca = '2' ;

id\_ = 22 ;

consec = 0 ;

fprintf(qc,"%15.6f %3d %6.2f\n",ccd\_time, QC1, atti\_rms) ;

fprintf(outn,"%15.6f %2d %2d %2d %2d %8.3f\n",ccd\_time, id\_, 0, 0, consec, 0.0) ;

fprintf(outM,"%12.6f %3d\n%20.15f %20.15f %20.15f \n%20.15f %20.15f %20.15f \n% 20.15f %20.15f %20.15f \n",ccd\_time, QC1, inv(M)) ;

%updated 7/27 from M

end % end of while (nstar = read\_CCD( ) ) \*/

if (nstar > 0)

total\_mstar = total\_mstar + nstar ;

end

if (nstar > Ns )

fprintf(stdout,"Too many stars at t=%8.2f\n", ccd\_time) ;

error(msg) ;

end

if (nstar == -999)

break ; % 2nd break : 1st break occurred with no CCD data \*/

end

% [t\_gyro, w, u, gread] = read\_gyro(b\_est\_average, gyro\_meas, gyro\_count);

% if (gread == 1)

% fprintf(stdout, "No more gyro data at %10.2f while IST data remains\n", ccd\_time) ;

% if (ccd\_time >= TIMELIMIT)

% fprintf(outgbias, "%15ld %15.5f %15.5f %15.5f\n", b\_est\_cnt, b\_est\_sum/b\_est\_cnt\*3600/PI\*180.0);

% error(msg) ;

% end

% else

% error(msg) ;

% end

% while (ccd\_time - t\_gyro) > 0.0001

% pseudo\_time = t\_gyro ;

% if (pseudo\_time < KF)

% w = w0;

% end

% %rg = no\_stars(pseudo\_time, t\_gyro, q0, w) ;

% rg = no\_stars(q0, w) ;

% qp = q0;

% A = q\_to\_A(qp) ;

% M = zeros(3,3);

% for i=1:1:3

% for j=1:1:3

% for k=1:1:3

% M(i,j) = M(i,j) + T\_B(k,i) \* A(k,j);

% end

% end

% end

% cnt0 = cnt0 + 1;

% if (cnt0 > 100.0)

% cctime = pseudo\_time + Ctime ;

% consec = 0 ;

% QC1 = 0 ; atti\_rms = -1.0 ;

% end

% if (cnt0 > 600.0)

% dt\_merge = cnt0 ;

% ready\_merge = 1 ;

% QC1 = 0 ; atti\_rms = -2.0 ;

% end

% fprintf(out1b,"%15.6f 23 ", pseudo\_time) ; % No IST timetags \*/

% fprintf(out1b," %15.12f",qp) ;

% fprintf(out1b,"\n") ;

% ID\_loca = '3' ; id\_ = 23 ;

% %fprintf(qc,"%15.6f %3d %6.2f\n", pseudo\_time, QC1, atti\_rms) ; stupid code

% fprintf(outn,"%15.6f %2d %2d %2d %2d %8.3f\n", pseudo\_time, id\_, 0, 0, consec, 0.0) ;

% fprintf(outM,"%12.6f %3d\n%20.15f %20.15f %20.15f \n%20.15f %20.15f %20.15f \n% 20.15f %20.15f %20.15f \n", pseudo\_time, QC1, inv(M)) ;

% [t\_gyro, w, u, gread] = read\_gyro(b\_est\_average, gyro\_meas, gyro\_count) ;

%

% if (gread == 1)

% fprintf(stdout,"No more gyro data. IST data may exist at %10.2f\n", t\_gyro) ;

% if (ccd\_time >= 86400.)

% fprintf(outgbias, "%15ld %15.5f %15.5f %15.5f\n", b\_est\_cnt, b\_est\_sum/b\_est\_cnt\*3600/3.1415\*180.0) ;

% error(msg) ;

% else

% error(msg) ;

% end

% end

% end % end of while (ccd\_time == t\_gyro) \*/

% if (t\_gyro < KF) for(i=0;i<3;i++) w[i] = w0[i];

numlines = 0; % stupid code

if ( (t\_gyro - t\_rms) >= DT\_RMS) % rms calc. with 10 minute intervals \*/

if (rmslnum > 0)

fprintf(outds," %12.5f %9d %12.8f %12.8f %12.8f\n", ccd\_time, rmslnum,rmslsum/rmslnum\*180.0/3.1415\*3600) ;

else

fprintf(outds," %12.5f %9d %12.8f %12.8f %12.8f\n",ccd\_time, rmslnum, 0.0, 0.0, 0.0) ;

rmslsum = zeros(3,1) ;

rmslnum = 0 ;

end

if (rmszlnum > 0)

fprintf(outdq," %12.5f %9d %12.8f %12.8f %12.8f %12.8f\n", ccd\_time, rmszlnum, rmszlsum/rmszlnum\*180.0/3.1415\*3600.0, rmszscalar/rmsznum ) ;

else

fprintf(outdq," %12.5f %9d %12.8f %12.8f %12.8f %12.8f\n", ccd\_time, rmszlnum, 0.0, 0.0, 0.0, 0.0) ;

rmszlsum = zeros(3,1) ;

rmszlscalar = 0.0 ;

rmszlnum = 0 ;

t\_rms = t\_rms + DT\_RMS ;

end

end

if(ccd\_time < KF)

w0 = w;

end

numlines = numlines + 1;

end

%end of while \*/

if (rmslnum > 0)

fprintf(outds," %12.5f %9d %12.8f %12.8f %12.8f\n",ccd\_time, rmslnum, rmslsum/rmslnum\*180.0/3.1415\*3600.0) ;

else

fprintf(outds," %12.5f %9d %12.8f %12.8f %12.8f\n",ccd\_time, rmslnum, 0.0, 0.0, 0.0) ;

end

fprintf(outds," %12.5f %9d %12.8f %12.8f %12.8f\n",ccd\_time, rmsnum, rmssum/rmsnum\*180.0/3.1415\*3600.0);

if (rmszlnum > 0)

fprintf(outdq," %12.5f %9d %12.8f %12.8f %12.8f %12.8f\n",ccd\_time, rmszlnum, rmszlsum/rmszlnum\*180.0/3.1415\*3600.0,rmszscalar/rmsznum ) ;

else

fprintf(outdq," %12.5f %9d %12.8f %12.8f %12.8f %12.8f\n",ccd\_time, rmszlnum, 0.0, 0.0, 0.0, 0.0) ;

end

fprintf(outdq," %12.5f %9d %12.8f %12.8f %12.8f %12.8f\n",ccd\_time, rmsznum,rmszsum/rmsznum\*180.0/3.1415\*3600.0,rmszscalar/rmsznum ) ;

fprintf(outgbias, "%15ld %15.5f %15.5f %15.5f\n",b\_est\_cnt, b\_est\_sum/b\_est\_cnt\*3600/3.1415\*180.0) ;

fprintf(stdout, "End of PROGRAM\n") ;

fprintf(stdout, "total\_mstar = %15.0f total\_ided = %15.0f\n",total\_mstar, total\_ided) ;

fprintf(stdout, "c\_pm = %8d c\_pm\_ided = %8d c\_dm = %8d c\_dm\_ided = %8d\n", c\_pm, c\_pm\_ided, c\_dm, c\_dm\_ided) ;

t1 = clock() ;

time(now2);

wallclock = difftime (now2, now) ;

fprintf(stderr, "It's now %s \n", ctime(now2)) ;

fprintf(stderr, "it took %.5f CPU secs.\n",(t1-t0)/CLOCKS\_PER\_SEC) ;

fprintf(stderr, "it took %.5f wall clock secs.\n", wallclock) ;

error(msg);