f\_s=zeros();%functioning sensors

s\_c=zeros();%sensor slit coordinates

pos\_s=zeros();%position or coordinates of each sensor taking center of satellite body frame as the origin(0,0,0)

rot\_axes=zeros();%if the sensor frame is at some angle with the body frame then input the angles.

ns=input('enter no of sensors');%enter no of sensors

%taking position coordinates of each sensor wrt body frame

%for i=1:ns

% for j=1:3

% pos\_s(i,j)=input('enter (x,y,z) coordinate wrt body frame for the sensor');

% end

%end

%other way of taking the inputs of coordinates of the sensor position on

%body knowing each has been positioned on the corner

a=input('enter the length of one side of the cube');

pos\_s1=[-a/2 a/2 a/2; a/2 a/2 a/2; a/2 a/2 -a/2; -a/2 a/2 -a/2; -a/2 -a/2 a/2; a/2 -a/2 a/2; a/2 -a/2 -a/2; -a/2 a/2 -a/2];

disp(pos\_s1);

%taking rotation angles of each axis of each sensor wrt body frame

for i=1:ns

for j=1:3

rot\_axes(i,j)=input('enter yaw,pitch and roll wrt body frame for the sensor');

end

end

disp(pos\_s);

%

dcm\_f=zeros(ns\*3,3);

for i=1:ns

dcm\_1=angle2dcm(rot\_axes(i,1),rot\_axes(i,2),rot\_axes(i,3));

disp('dcm of sensor');

disp(dcm\_1);

%disp(dcm\_1(1,2));

%disp(dcm\_1(1:2,2:3));

dcm\_f(i\*3-2:i\*3,1:3)=dcm\_1(1:3,1:3);

end

disp('dcm matrix of all sensors');

disp(dcm\_f);

%

%

%input 1 if sun is in sensor's fov

for i=1:ns

for j=1:1

f\_s(i,j)=input('enter 1 if the sensor is having sun in fov' );

end

end

disp('sensors functioning in given case');

disp(f\_s);

%input coordintes on slit of each sensor

for i=1:ns

for j=1:2

if f\_s(i,1)==1%do not take input for the sensor if value is not 1

s\_c(i,j)=input('enter x,y coordinates for sunvector falling on sensor"s slit');

elseif f\_s(i,1)==0

s\_c(i,j)=0;

else

continue;%if value is not 1,continue to next sensor

end

end

end

disp('(x,y)coordinates on each sensor slit');

disp(s\_c);

%matrix or array containing x cordinates of each sensor

x\_c=zeros();

for i=1:ns

if f\_s(i,1)==1

x\_c(i)=s\_c(i,1);

elseif f\_s(i,1)==0

x\_c(i)=0;

end

end

disp('array containing x cordinates of each sensor');

disp(x\_c);

%matrix or array containing y cordinates of each sensor

y\_c=zeros();

for i=1:ns

if f\_s(i,1)==1

y\_c(i)=s\_c(i,2);

elseif f\_s(i,1)==0

y\_c(i)=0;

end

end

disp('array containing y cordinates of each sensor');

disp(y\_c);

%x1=x\_c(1);%x cordinate of the sensor 1 slit

%y1=y\_c(1);%y cordinate of the sensor 1 slit

h=input('enter the height difference between the slit and dial');

%elevation matrix of the sensors

el\_s=zeros();

for i=1:ns

if f\_s(i,1)==1

el\_s(i)=atand(y\_c(i)/x\_c(i));

elseif f\_s(i,1)==0

el\_s(i)=0;

end

end

disp('elevation matrix of each sensor');

disp(el\_s);

%azimuth of the sensors

az\_s=zeros();

for i=1:ns

if f\_s(i,1)==1

az\_s(i)=atand(sqrt((x\_c(i)^2)+(y\_c(i)^2))/h);

elseif f\_s(i,1)==0

az\_s(i)=0;

end

end

disp('azimuth matrix of each sensor');

disp(az\_s);

%el1=atand(y1/x1);%elevation of sunvector by sensor 1

%ell1=el\_s(1);%elevation of sunvector by sensor 1 from matrix

%az1=atand(sqrt((x1^2)+(y1^2))/h);%azimuth of sunvector by sensor 1

%disp('x cord of sensor 1');

%disp(x1);

%disp('y cord of sensor 1');

%disp(y1);

%disp('elevation of sensor 1');

%disp(el1);

%disp(ell1);

%disp('azimuth of sensor 1');

%disp(az1);

%disp(az\_s(1));

%conversion of polar coordinates theeta and phi to the cartesian model

v\_x\_s=zeros();%sun vector x cordinate for each sensor from sun sensor frame

for i=1:ns

if f\_s(i,1)==1

v\_x\_s(i)=sin(el\_s(i))\*cos(az\_s(i));

elseif f\_s(i,1)==0

v\_x\_s(i)=0;

end

end

disp('x coordinate of sun vector wrt each sun sensor frame');

disp(v\_x\_s);

v\_y\_s=zeros();%sun vector y cordinate for each sensor from sun sensor frame

for i=1:ns

if f\_s(i,1)==1

v\_y\_s(i)=sin(el\_s(i))\*sin(az\_s(i));

elseif f\_s(i,1)==0

v\_y\_s(i)=0;

end

end

disp('y coordinate of sun vector wrt each sun sensor frame');

disp(v\_y\_s);

v\_z\_s=zeros();%sun vector z cordinate for each sensor from sun sensor frame

for i=1:ns

if f\_s(i,1)==1

v\_z\_s(i)=cos(el\_s(i));

elseif f\_s(i,1)==0

v\_z\_s(i)=0;

end

end

disp('z coordinate of sun vector wrt each sun sensor frame');

disp(v\_z\_s);

for i=1:ns

if f\_s(i,1)==1

disp('sun vector wrt to the');disp(i);disp('sun sensor frame is');

disp(v\_x\_s(i));disp('i^ + ');disp(v\_y\_s(i));disp('j^ + ');disp(v\_z\_s(i));disp('k^');

end

end

%considering frames are having there axes in the same directions so we do

%not need to do rotation of frames.

%performing translation of the vector from sun sensor frame to the body frame

%v\_xt\_b=zeros();

%for i=1:ns

% if f\_s(i,1)==1

% v\_xt\_b(i)=v\_x\_s(i)+pos\_s1(i,1);

% elseif f\_s(i,1)==0

% v\_xt\_b(i)=0;

% end

%end

%disp('sun vector x coordinates wrt body frame after simple translation opoeration from each sun sensor');

%disp(v\_xt\_b);

%v\_yt\_b=zeros();

%for i=1:ns

% if f\_s(i,1)==1

% v\_yt\_b(i)=v\_y\_s(i)+pos\_s1(i,2);

% end

%end

%disp('sun vector y coordinates wrt body frame after simple translation opoeration from each sun sensor');

%disp(v\_yt\_b);

%v\_zt\_b=zeros();

%for i=1:ns

% if f\_s(i,1)==1

% v\_zt\_b(i)=v\_z\_s(i)+pos\_s1(i,3);

% end

%end

%disp('sun vector z coordinates wrt body frame after simple translation opoeration from each sun sensor');

%disp(v\_zt\_b);

%

%

%

%check from here

%

%

%

%rotation matrix for each sensor

%r\_x=zeros(ns,3);

%r\_y=zeros(ns,3);

%r\_z=zeros(ns,3);

%for i=1:ns

% if f\_s(i,1)==1

% r\_x(i)=[1 0 0; 0 cos(rot\_axes(i,1)) -sin(rot\_axes(i,1)); 0 sin(rot\_axes(i,1)) cos(rot\_axes(i,1))];

% r\_y(i)=[cos(rot\_axes(i,2)) 0 sin(rot\_axes(i,2)); 0 1 0; -sin(rot\_axes(i,2)) 0 cos(rot\_axes(i,2))];

% r\_z(i)=[cos(rot\_axes(i,3)) -sin(rot\_axes(i,3)) 0; sin(rot\_axes(i,3)) cos(rot\_axes(i,3)) 0; 0 0 1];

% end

%end

%performing rotation

%v\_x\_bx=zeros();

%v\_x\_by=zeros();

%v\_x\_b=zeros();

%for i=1:ns

% if f\_s(i,1)==1

%performing rotation about x

% v\_x\_bx(i)= mtimes(r\_x(i),[v\_xt\_b(i);v\_yt\_b(i);v\_zt\_b(i)]);

%performing rotation about y

% v\_x\_by(i)= mtimes(r\_y(i),v\_x\_bx(i));

%performing roration about z

% v\_x\_b(i)= mtimes(r\_z(i),v\_x\_by(i));

% end

% disp('position of the sun vector about the body frame as given by sensor is');

% disp(v\_x\_b(i));

%end

%disp('fianl position of the sun vector matrix wrt body frame');

%disp(v\_x\_b);

%furhur now we need to compare the values as given by the each sensor

%output as a cordinate wrt body frame, the values must be same, if they are

%not we need to incorporate standard deviation to have an idea of an error

%occouring in the system.

%

%

%

%

%trying the dcm method

%

%

%

f\_c\_bf=zeros(4\*ns,1);

for i=1:ns

t\_r=zeros(4,4);

t\_r(1:3,1:3)=dcm\_f(i\*3-2:i\*3,1:3);

t\_r(1:3,4)=-(pos\_s1(i,1:3));

t\_r(4,1:4)=[0 0 0 1];

disp('transformation matrix of sensor is');

disp(t\_r);

%f\_c\_b=zeros(4,1);

f\_c\_b=t\_r\*[v\_x\_s(i);v\_y\_s(i);v\_z\_s(i);1];

disp('final sun vector wrt body frame');

disp(f\_c\_b);

f\_c\_bf(i\*4-3:i\*4,1)=f\_c\_b(1:4,1);

end

disp('matrix having all values of sun vector wrt body frame');

disp(f\_c\_bf);