# To obtain RA, Dec & $\phi$ for all the three cameras (Scanning Sky Monitor) aboard ASTROSAT from the Satellite Attitude Information


## *Sushila Mishra*

***under the guidance of Dipankar Bhattacharya***

*Astronomy & Astrophysics Group, Raman Research Institute, Bangalore 560 080*


(Sept 2005)

**Abstract**

RA, Dec & $\phi$ for all the three cameras (SSM) is obtained from the ASTROSAT Attitude Information i.e. the star sensor data, gyro data and resolver data. Star sensors is one of the most accurate means of attitude determination. From the star sensor data using quaternion transformation RA and Dec of the boom axis is obtained. From the given quaternions and resolver data, $\phi_B$ of the boom axis is obtained. The coordinates of the other two axes i.e. the common center coordinates of the slanted cameras is obtained from RA, Dec and $\phi_B$ of the boom axis.

# 1 Introduction

The present day star sensors provide information known as quaternion. This is a four parameter set used to describe the orientation of one reference frame with respect to a second reference frame. Although only three parameters are needed to uniquely specify the relative orientation, all three parameter sets have singularities which make them unsuitable for numerical simulations, or integration in flight software.

The quaternion is represented by a four row-vector

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} s \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \cos\frac{\phi}{2} \\ a_1 \sin\frac{\phi}{2} \\ a_2 \sin\frac{\phi}{2} \\ a_3 \sin\frac{\phi}{2} \end{bmatrix} = \mathbf{q} \tag{1}$$

The first notation is the standard used in software. No distinction is made between the four elements of the quaternion since numerically there is none. Ocassionally authors will make the first element $q_0$ and the remainder $[q_1\ q_2\ q_3]$ to correspond to the scalar and vector notation. The second notation breaks the quaternion into a scalar and 3-vector part. Letting the first term be the "scalar" is arbitrary.

The third form relates to the theorem of Euler that any arbitrary rotation about any number of axes can be reduced to a single rotation about a fixed axis. If the rotation angle is defined as $\phi$ and the axis of rotation is $\vec{a}$, then the sign of the first value is arbitrary since any rotation can be represented by two quaternions of opposite sign. Here we define $\phi$ to be in a right handed corkscrew sense along the axis of rotation. In this text, the first element of the quaternion will always be positive. This would imply that $\phi$ will always be between $\pm 180°$. The transpose of a quaternion is defined as

$$\begin{bmatrix} q_1 \\ -q_2 \\ -q_3 \\ -q_4 \end{bmatrix} = \begin{bmatrix} s \\ -v_1 \\ -v_2 \\ -v_3 \end{bmatrix} = \begin{bmatrix} \cos\frac{\phi}{2} \\ -a_1 \sin\frac{\phi}{2} \\ -a_2 \sin\frac{\phi}{2} \\ -a_3 \sin\frac{\phi}{2} \end{bmatrix} = \mathbf{q}^* \tag{2}$$

with the property that

$$\mathbf{q}\mathbf{q}^{\mathbf{T}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

Introduction to quaternions taken from Princeton Satellite Simulator document, chapter 10 on coordinate transformations, section 10.3.2 (refer [1]).

## 1.1 Quaternion multiplication

Just as one can multiply two rotation matrices to get another rotation matrix, one can multiply quaternions to compute the effect of a series of rotations. Like matrix multiplication, quaternion multiplication is not commutative because the order of rotation matters.

Let's define three quaternions to have elements **q**, **l**, and **r** :

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} \qquad \mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{pmatrix} \qquad \mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} \tag{4}$$

To compute the quaternion product $\mathbf{q} = \mathbf{l} \times \mathbf{r}$, use the following algorithm :

$$\begin{aligned} q_1 &= l_1 \times r_1 - l_2 \times r_2 - l_3 \times r_3 - l_4 \times r_4 \\ q_2 &= l_2 \times r_1 + l_1 \times r_2 - l_4 \times r_3 + l_3 \times r_4 \\ q_3 &= l_3 \times r_1 + l_4 \times r_2 + l_1 \times r_3 - l_2 \times r_4 \\ q_4 &= l_4 \times r_1 - l_3 \times r_2 + l_2 \times r_3 + l_1 \times r_4 \end{aligned} \tag{5}$$

The above algorithm uses 16 multiplications and 12 additions. In comparison, it takes three multiplications and two additions to compute each of the nine elements in a 3-D rotation matrix. Thus, using quaternions saves 11 multiplications, six additions, and five assignment statements. These savings mean that one can multiply quaternions in roughly half the time multiplying rotation matrices takes (refer [2]).

## 1.2 Quaternion Transformations

Quaternions transform vectors by means of the following operation

$$\mathbf{x}_b = \mathbf{q}^*_{ba} \mathbf{x}_a \mathbf{q}_{ba} \tag{6}$$

using quaternion multiplication with the vectors defined as quaternions with a zero scalar part :

$$\mathbf{x}_a = \begin{bmatrix} 0 \\ x_a(1) \\ x_a(2) \\ x_a(3) \end{bmatrix} \tag{7}$$

## 2 Star Sensor and Measurements

The functional meaning of the measurement as provided by the sensor is nothing but a transformation between two co-ordinate systems. In star sensor case the information connects star sensor frame to inertial frame. Similarly any transformation matrix can be represented by a 4 parameter $\mathbf{q}$ given by

$\mathbf{q} = \{q_1, q_2, q_3, q_4\}$ and with a constraint $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1.0$

Usually this information is given continuously, for example, every 400 ms or so; time vs $\mathbf{q}$ which can be used to connect vector directions to inertial frame.

Physically this means a transformation matrix connecting the star sensor frame to Earth Centered Inertial Frame as given below. Thus the sensor frame is connected to inertial frame by this transformation matrix and therefore any direction in the spacecraft can be connected to the inertial frame if we know the mounting angles of star sensor and other direction as well.

$$A = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_2 + q_3q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{8}$$

Each row in the A matrix is nothing but direction cosines of that axis of sensor in inertial frame. Incidentally, a co-ordinate frame is attached to a sensor and is called sensor frame. One of the axis will be about the bore axis and this axis is important for astronomy satellite purposes. If the bore axis of the star sensor is x-axis, then the first row provides the bore axis in inertial frame. Similarly, for other axes. For ASTROSAT, y-axis may be the bore axis.

## 3 To convert from quaternions to RA ($\alpha$) and Dec ($\delta$)

Using quaternion transformations given as

$$\mathbf{x}_b = \mathbf{q}_{ba}^{*}\mathbf{x}_a\mathbf{q}_{ba} \tag{9}$$

The quaternion rotates a vector from frame 'a' to frame 'b'. To obtain the RA and Dec of the boom axis, frame 'a' corresponds to the star sensor coordinate and frame 'b' corresponds to the inertial frame. The unit vector $\vec{x}_a$ [$x_a(1)$, $x_a(2)$, $x_a(3)$] corresponds to the SSM boom axis in the sensor frame, and $\vec{x}_b$[$x_b(1)$, $x_b(2)$, $x_b(3)$] gives the same unit vector as seen in the inertial frame. Where,

The vector $\vec{x}_a$ is a function of $\theta_1$ and $\theta_2$ as shown in the figure 1 given as

$$\mathbf{x}_a = \begin{pmatrix} 0 \\ x_a(1) \\ x_a(2) \\ x_a(3) \end{pmatrix} = \begin{pmatrix} 0 \\ \sin\theta_1\cos\theta_2 \\ \sin\theta_1\sin\theta_2 \\ \cos\theta_1 \end{pmatrix} \tag{10}$$

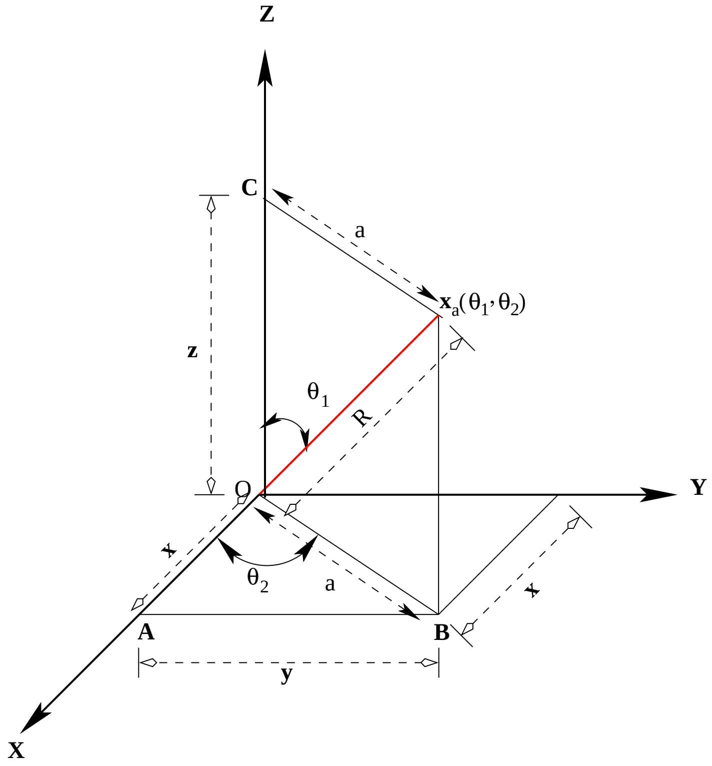


Figure 1: Any vector in space in terms of cartesian star sensor coordinates

Also, $[x_a(1)]^2 + [x_a(2)]^2 + [x_a(3)]^2 = 1$

$$\mathbf{q}^*_{ba} = \begin{pmatrix} \cos\frac{\phi}{2} \\ -a_1 \sin\frac{\phi}{2} \\ -a_2 \sin\frac{\phi}{2} \\ -a_3 \sin\frac{\phi}{2} \end{pmatrix} \tag{11}$$

$$\mathbf{q}_{ba} = \begin{pmatrix} \cos\frac{\phi}{2} \\ a_1 \sin\frac{\phi}{2} \\ a_2 \sin\frac{\phi}{2} \\ a_3 \sin\frac{\phi}{2} \end{pmatrix} \tag{12}$$

$a_1$, $a_2$ and $a_3$ are the components of the unit vector $\vec{a}$ in frame 'a' around which a rotation by angle $\phi$ connects the two frames 'a' and 'b'. By definition, $a_1^2 + a_2^2 + a_3^2 = 1$.
Substituting the values of equation 10, 11 and 12 in equation 9 and doing quaternion multiplication yields the following

$$x_b(1) = 0 \tag{13}$$

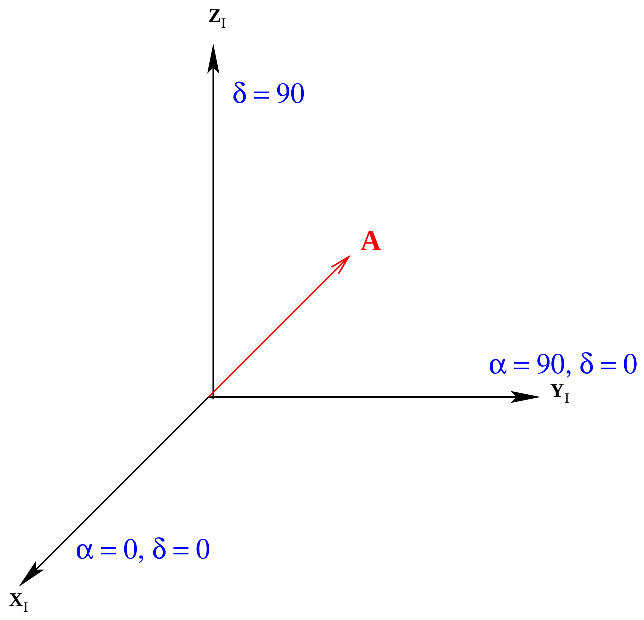


Figure 2: Vector $\vec{A}$ in space

$$\begin{aligned} x_b(2) &= \sin\theta_1\cos\theta_2\sin^2(\phi/2)[a_1^2 - a_2^2 - a_3^2] + \sin\theta_1\cos\theta_2\cos^2(\phi/2) \\ &+ 2a_1a_2\sin\theta_1\sin\theta_2\sin^2(\phi/2) + 2a_1a_3\cos\theta_1\sin^2(\phi/2) \\ &- a_2\cos\theta_1\sin\phi + a_3\sin\theta_1\sin\theta_2\sin\phi \end{aligned} \tag{14}$$

$$\begin{aligned} x_b(3) &= \sin\theta_1\sin\theta_2\sin^2(\phi/2)[-a_1^2 + a_2^2 - a_3^2] + \sin\theta_1\sin\theta_2\cos^2(\phi/2) \\ &+ 2a_1a_2\sin\theta_1\cos\theta_2\sin^2(\phi/2) + 2a_2a_3\cos\theta_1\sin^2(\phi/2) \\ &- a_3\sin\theta_1\cos\theta_2\sin\phi + a_1\cos\theta_1\sin\phi \end{aligned} \tag{15}$$

$$\begin{aligned} x_b(4) &= \cos\theta_1\sin^2(\phi/2)[-a_1^2 - a_2^2 + a_3^2] + \cos\theta_1\cos^2(\phi/2) \\ &+ 2a_1a_3\sin\theta_1\cos\theta_2\sin^2(\phi/2) + 2a_2a_3\sin\theta_1\sin\theta_2\sin^2(\phi/2) \\ &+ a_2\sin\theta_1\cos\theta_2\sin\phi - a_1\sin\theta_1\sin\theta_2\sin\phi \end{aligned} \tag{16}$$

We can now derive the RA ($\alpha$) and Declination ($\delta$) of a vector from its components in the inertial frame. From figure 2 we can see that suffix 'I' corresponds to inertial frame of reference where, $\delta = 90^\circ$ lies on $Z_I$ axis, $\alpha = 0$, $\delta = 0$ lies on $X_I$ axis and $\alpha = 90^\circ$, $\delta = 0$ lies on $Y_I$ axis. If there is a vector $\vec{A}$ in the space, then $\vec{A}$ in inertial frame is related to $\alpha$ and $\delta$ with the following equations

$$\begin{aligned} A_{xI} &= \cos\delta\cos\alpha \\ A_{yI} &= \cos\delta\sin\alpha \\ A_{zI} &= \sin\delta \end{aligned} \tag{17}$$

Hence, $\mathbf{x_b}$ is also represented as

$$\mathbf{x}_b = \begin{pmatrix} 0 \\ \\ x_b(2) \\ \\ x_b(3) \\ \\ x_b(4) \end{pmatrix} = \begin{pmatrix} 0 \\ \\ A_{xI} \\ \\ A_{yI} \\ \\ A_{zI} \end{pmatrix} = \begin{pmatrix} 0 \\ \\ \cos\delta\cos\alpha \\ \\ \cos\delta\sin\alpha \\ \\ \sin\delta \end{pmatrix} \tag{18}$$

which imply

$$\delta = \sin^{-1}[x_b(4)] \tag{19}$$

and

$$\alpha = \tan^{-1}\left[\frac{x_b(3)}{x_b(2)}\right] \tag{20}$$

Hence $\alpha$ and $\delta$ of the boom axis is obtained. A routine has been written to convert from quaternion to $\alpha$ and $\delta$ [refer code list 6.1, routine name 'qtoradec()'].

# 4 To derive the meridian angle of the Boom rotation

The angle between the meridian passing through the boom and the reference axis of the boom camera is defined as the meridian angle of the boom rotation ($\phi_B$). The $\phi_B$ of the boom axis is obtained from quaternions and resolver data ($\phi_R$, the boom rotation angle with respect to a reference fixed to the satellite body). Let ($\alpha_{ref}$, $\delta_{ref}$) represent the inertial coordinates of the reference point for the resolver with respect to which $\phi_R$ is measured. This reference point would have mounting angles $\theta_{1ref}$ and $\theta_{2ref}$ in the star sensor frame. $\alpha_{ref}$ and $\delta_{ref}$ for this reference point can be obtained using the quaternion transformation to $\alpha$ and $\delta$ as mentioned in the previous section, substituting the value of $\theta_{1ref}$ and $\theta_{2ref}$ in equation (10) in place of $\theta_1$ and $\theta_2$ respectively. Solving equation (9) yields the value of $\alpha_{ref}$ and $\delta_{ref}$. From the resolver angle ($\phi_R$) and $\delta_{ref}$, $\phi_B$ of the boom can then be obtained as follows

$$\phi_B = \phi_R + (90^\circ - \delta_{ref}) \tag{21}$$

# 5 To derive RA and Dec of the slanted cameras

The values of $\alpha_B$, $\delta_B$ and $\phi_B$ is used to obtain ra and dec of the slanted cameras. Let $\alpha_s$ and $\delta_s$ represents the common center coordinates of the slanted cameras and is obtained using spherical transformations as follows

$$\cos\delta_s \sin(\alpha_s - \alpha_B) = -\sin\theta_c \sin\phi_B \tag{22}$$

$$\cos\delta_s \cos(\alpha_s - \alpha_B) = \cos\theta_c \cos\delta_B - \sin\theta_c \sin\delta_B \cos\phi_B \tag{23}$$

$$\sin\delta_s = \cos\theta_c \sin\delta_B + \sin\theta_c \cos\delta_B \cos\phi_B \tag{24}$$

where, $\theta_c$ is the canting angle. Refer figure 6 (Slanted Camera Frame) in the report [3]. Solving equation (22) and (23) yields $\alpha_s$

$$\alpha_s = \alpha_B + \tan^{-1}\left[\frac{-\sin\theta_c \sin\phi_B}{\cos\theta_c \cos\delta_B - \sin\theta_c \sin\delta_B \cos\phi_B}\right] \tag{25}$$

and $\delta_s$ is obtained from equation (24) given as

$$\delta_s = \sin^{-1}\left[\cos\theta_c \sin\delta_B + \sin\theta_c \cos\delta_B \cos\phi_B\right] \tag{26}$$

# 6 Code List

All the codes listed below is in machine **moose.rri.local.net**

## 6.1 /moose2/sushila/2July2005_fits/18Aug05/qtoradec.c

The function 'qtoradec.c' converts from any given quaternion to RA($\alpha$) and dec($\delta$). The four quaternion values, $\theta_1$ and $\theta_2$ is passed as input parameters to this routine and the function returns $\alpha$ and $\delta$.

## 6.2 /moose2/sushila/2July2005_fits/18Aug05/boomtoslant.c

The function 'boomtoslant.c' evaluates slanted camera coordinate. The values of $\alpha_B$, $\delta_B$ and $\phi_B$ is passed as input parameters to this function and it returns $\alpha_s$ and $\delta_s$ i.e. the common center coordinate of the slanted cameras.

## 6.3 /moose2/sushila/2July2005_fits/18Aug05/sim_fits.c

To convert the simulation file to fits binary file. The code 'sim_fits.c' takes the simulation filename (ascii file) as the command line argument and converts the ascii file to fits binary file. The program also takes care of defining the various primary header into the fits binary file. For example, the start and end quaternions values is defined in the header file (refer 'prim_header.h') which is provided by the star sensor data. These values is then define in the primary header of the fits binary file. Then these values is read from the fits primary header and is passed to the routine 'qtoradec()' to calculate ra and dec. The values of ra and dec is stored into the primary header of the fits binary file. Also, the routine 'boomtoslant()' is called to convert from boom coordiantes to common center coordinate of the slanted cameras.

The fits primary header is listed below

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                   16 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
DATE_OBS=                   0. / Date of observation (calender date)
S_ST_OB =                   0. / Start scan time (on-board clock)
E_ST_OB =                   0. / End scan time (on-board clock)
S_ST_UT =                   0. / Start scan time (UT)
E_ST_UT =                   0. / End scan time (UT)
S_MJD   =                   0. / Start Modified Julian Date (MJD)
E_MJD   =                   0. / End Modified Julian Date (MJD)
P_LEVEL =                    0 / Processing level (0, 1, 2, 3, ...)
OBS_MODE=                    0 / Observation Mode (0=step & stare / 1=rotating)
START_Q1=              0.45677 / Start quaternion value1
START_Q2=              0.08912 / Start quaternion value2
START_Q3=              0.23456 / Start quaternion value3
START_Q4=              0.77345 / Start quaternion value4
S_RA_B  =     100.356292609109 / Start RA Boom
S_DEC_B =     48.3303102812087 / Start DEC Boom
S_RESOLV=                  20. / Start Resolver Angle
S_MERI_A=     91.7408368259634 / Start Meridian Angle
S_RA_SL =     65.7474727345672 / Start RA Slant Center Coordinate
S_DEC_SL=      41.948880985664 / Start DEC Slant Center Coordinate
END_Q1  =              0.03024 / End quaternion value1
END_Q2  =              0.40617 / End quaternion value2
END_Q3  =              0.05607 / End quaternion value3
END_Q4  =              0.45007 / End quaternion value4
E_RA_B  =     11.7301062801922 / End RA Boom
E_DEC_B =     2.21112065922902 / End DEC Boom
E_RESOLV=                  50. / End Resolver Angle
E_MERI_A=      138.07702031136 / End Meridian Angle
E_RA_SL =     354.628793079223 / End RA Slant Center Coordinate
E_DEC_SL=    -16.2151705876558 / End DEC Slant Center Coordinate
GTI_FILE= 'sim_rt0.000_rp0.067_f1.0000_sx0.0_sy0.0_t0.00_p0.00.gti'
COMMENT   GTI_FILE : Good Time Interval Filename
DATE    = '2005-09-09T06:10:09' / file creation date (YYYY-MM-DDThh:mm:ss UT)
END
```

This primary header is at a very basic level. More fits keywords will be added into it later.

# References


[1] "Satellite Simulator User's Guide V1.1", 2002, Princeton Satellite Systems, URL : http://www.psatellite.com/products/manuals/SatSim.pdf

[2] Do-While Jones, "Quaternions quickly transform coordinates without error buildup", 1995, EDN design feature. URL : http://www.edn.com/archives/1995/030295/05df3.htm

[3] Ravi Shankar, "Dynamic Sky Simulation with Scanning Sky Monitor on ASTROSAT using spherical transformations", 2004, RRI Internal Report (RRI_SSM_0010).