estec

European Space Research
and Technology Centre
Keplerlaan 1
2201 AZ Noordwijk
The Netherlands
T +31 (0)71 565 6565
F +31 (0)71 565 6040
www.esa.int

MEMO

Date	11/07/2017	**Ref**	
From	Andrew Hyslop	**Visa**	
To	David Futyan, Carlos Corral Van Damme		
Copy			

Subject: Proper Handling of APE Time Series and CCD Target Location for CHEOPSim - Simplified

This memo describes an improved but simplified approach for incorporating APE time series data and CCD target location into the synthetic image generation process. The approach is designed to minimize code modifications to CHEOPSim whilst still solving the following issues:

- Avoiding hardcoded and incorrect mapping of APE time series data to perturbations in the CCD frame
- Fixing the center of roll to be at the CCD target location, perturbed by the APE

The simplified approach presented here assumes small angle approximations for interpretation of CCD target location and APE data.

Prototype code was generated in Matlab, and this can be found in accompanying .m files (run *CHEOPSim_update_test.m*) and in the Appendix.

Step 1

Compute orbital position in J2000 using some orbit propagator. This is necessary for computing the roll angle that is input to the cfitsio *fits_world_to_pix* function.
This is already implemented in CHEOPSim.

Step 2

Load APE time series data from the text file and convert APE from arcsec to radians.

Step 3

Compute the guidance attitude matrix M_{i2sat} (direction cosine matrix that maps from inertial frame to satellite frame) and the nominal desired roll angle.
This is already implemented in CHEOPSim. In the prototype code (see Appendix) this is implemented in the *computeNomRoll.m* function.

Note that roll is defined in the context of this document as roll of the satellite about the LoS vector relative to a (zero) orientation with Ysat aligned with J2000 local RA direction and Zsat with local DEC direction.

This procedure can be found in *ACE7.SP.15520.ASTR – AS250 for CHEOPS – Software Requirements Specification, Issue 4.0*, under the name *ComputeCAPSUPGuidance*.

step 2 : **Compute rotation from inertial to orbit frame**

$$Yop = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

$$NEq = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$[0, Yop\_J2000] = Q_{LVLH/J2000} \otimes \begin{bmatrix} 0 & Yop \end{bmatrix} \otimes \overline{Q_{LVLH/J2000}}$$

$$Xop\_J2000 = Yop\_J2000.vector \times NEq$$

$$Xop\_J2000 = Xop\_J2000 \,/\, \|Xop\_J2000\|$$

$$Zop\_J2000 = Xop\_J2000 \times Yop\_J2000$$

step 3 : **Matrix from inertial to orbit frame**

$$Morb\_in = \begin{bmatrix} Xop\_J2000 \\ Yop\_J2000 \\ Zop\_J2000 \end{bmatrix}$$

step 4 : **Compute Los and Earth->S/C vectors in orbit frame**

Xdir_orb= Morb_in * Xdir_J2000

pos_orb= Morb_in * pos_J2000/ $\|pos\_J2000\|$

step 5 : **Ensures a min value of the LoS Y component to ensure the Y axes always has non-zero norm (LoS Y maps into Y axes X/Z components in orb frame)**

```
Xdir_mod=Xdir_orb;
Xdir_mod[y] =sign(Xdir_mod[y]) * max(abs(Xdir_mod[y]), Min_Ycomp);
```

Xdir_mod= Xdir_mod/ $\|Xdir\_mod\|$

step 6 : **Matrix from orbital frame to J2000**

Ydir_mod= $pos\_orb \wedge Xdir\_mod$

Zdir_orb= $Xdir\_orb \wedge Ydir\_mod$ // Z is computed from real LoS and modified Y

Zdir_orb= Zdir_orb/ $\|Zdir\_orb\|$

Ydir_orb= $Zdir\_orb \otimes Xdir\_orb$ // Y is computed from real X and Z

$$Msat\_orb = \begin{bmatrix} Xdir\_orb \\ Ydir\_orb \\ Zdir\_orb \end{bmatrix}$$

step 7 : **Rotation matrix from inertial to satellite frame**

Msat_in = Msat_orb * Morb_in;

$M_{i2sat} = Msat\_in$
Roll = atan2 ($Msat\_in[2,3]$, $Msat\_in[3,3]$)

...with matrix [row,col] indexing beginning at [1,1]

Step 4

Define RA, DEC coordinates for center of roll. This is implemented in the prototype code (see Appendix) in the *computeRefAndRoll.m* function.

(a) Shift target star unit direction vector m_{LoS} from J2000 into satellite frame, then shift by APE, then shift back into J2000 frame

$$dR = \begin{bmatrix} 1 & -APE_z & APE_y \\ APE_z & 1 & -APE_x \\ -APE_y & APE_x & 1 \end{bmatrix}$$

$$m_{ref} = M_{i2sat}{}^T dR\, M_{i2sat} m_{LoS}$$

(b) Convert adjusted vector to RA, DEC coordinates and perturb roll by APE[x]

$$m_{ref} = \frac{m_{ref}}{\|m_{ref}\|}$$

$$xy = \frac{m_{ref}[1:2]}{\|m_{ref}[1:2]\|}$$

$$RA_{ref} = \cos^{-1}(xy[1])\ \mathrm{sign}(xy[2])$$
$$DEC_{ref} = \sin^{-1}(m_{ref}[3])$$
$$roll_{true} = roll + APE_x$$

Step 5

Loop through all the stars potentially within FoV of instrument and map their RA, DEC coordinates into CCD frame using the cfitsio *fits_world_to_pix* (*ffxypx*) function.

(a) Shift coordinates to *fits_world_to_pix* input frame convention.

$rot = -roll_{true}$ …since *ffxypx* function input is roll about –LoS.

$$\begin{bmatrix} xref \\ yref \end{bmatrix} = M_{radec2ccd0} \begin{bmatrix} RA_{ref} \\ DEC_{ref} \end{bmatrix}$$

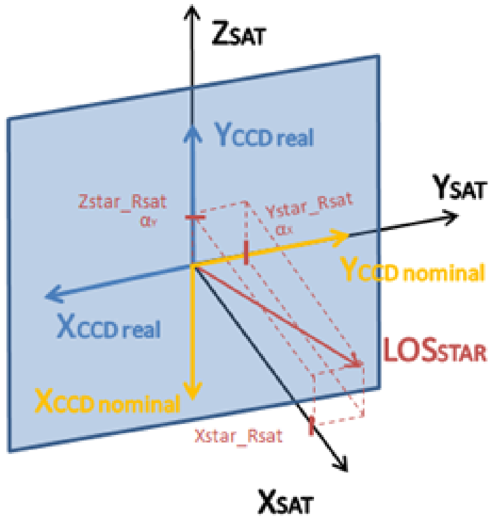
Then for each star…

$$\begin{bmatrix} xpos \\ ypos \end{bmatrix} = M_{radec2ccd0} \begin{bmatrix} RA \\ DEC \end{bmatrix}$$

where $M_{radec2ccd0}$ is a 2x2 matrix mapping delta RA & DEC angles to CCD directions when the satellite is aligned with the J2000 frame.

$$M_{radec2ccd0} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

...with reference to the figure below (from CHEOPS-ECE-SYS-TN-503, Instrument to AOCS interfaces and assumptions, Issue 2, p.14) where the CCD frame is called the 'CCDreal' in the figure.

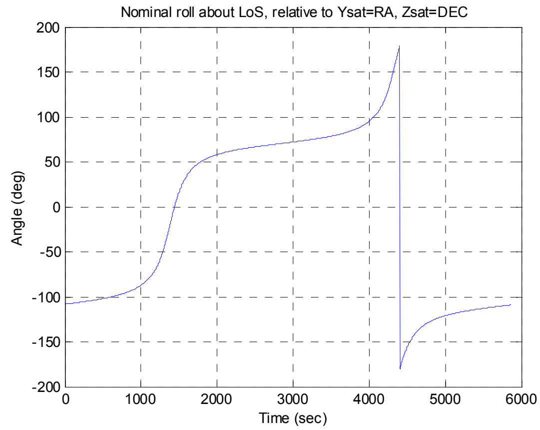


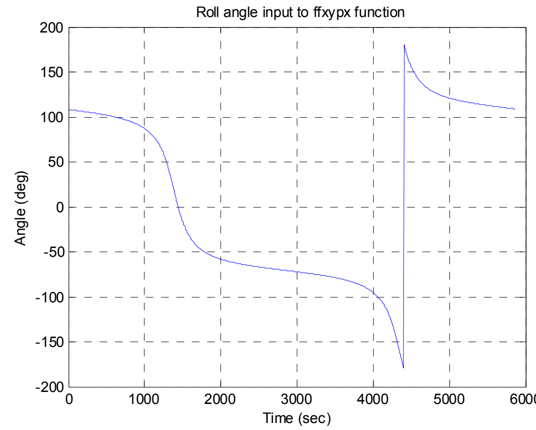
(b) Call *fits_world_to_pix* function.

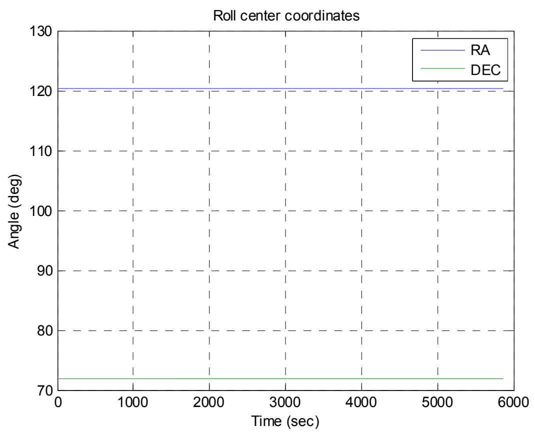
[xpix, ypix, status] = ffxypx*(xpos, ypos, xref, yref, xrefpix, yrefpix, xinc,yinc, rot, type)*

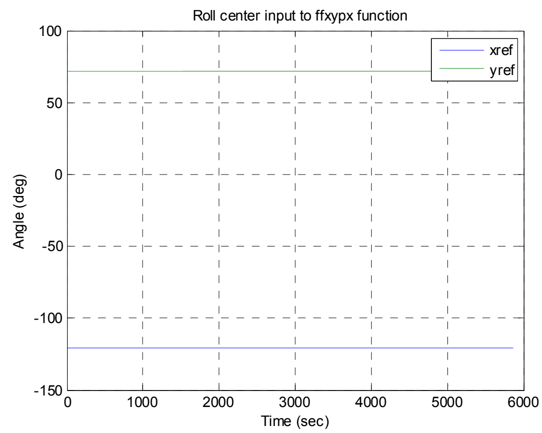
...where:
xpos = inertial (i.e. with rot=0) x angle (deg)
ypos = inertial (i.e. with rot=0) y angle (deg)
xref = inertial (i.e. with rot=0) x angle at center of rotation (deg)
yref = inertial (i.e. with rot=0) y angle at center of rotation (deg)
xrefpix = CCD target location coordinate in desired units, relative to desired origin
yrefpix = CCD target location coordinate in desired units, relative to desired origin
xinc = CCD x-axis degrees per pixel (or degrees per desired units)
yinc = CCD y-axis degrees per pixel (or desired units per degree)
rot = roll about –LoS relative to a (zero) orientation with Ysat aligned with J2000 local RA direction and Zsat with local DEC direction (deg)
type = projection type code, e.g. '-SIN' (note that -SIN, -TAN, -ARC, -NCP, -GLS, -MER or -STG seem to give reasonable results)

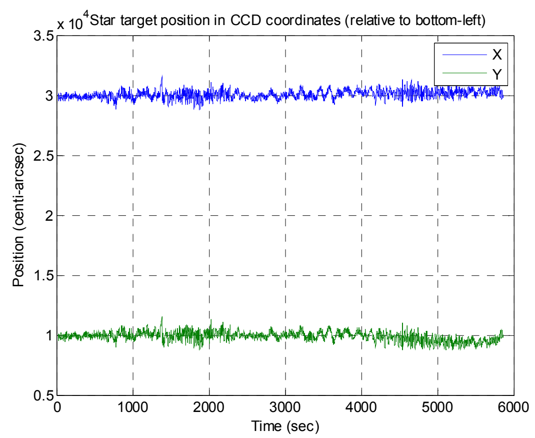
<u>Prototype results</u>

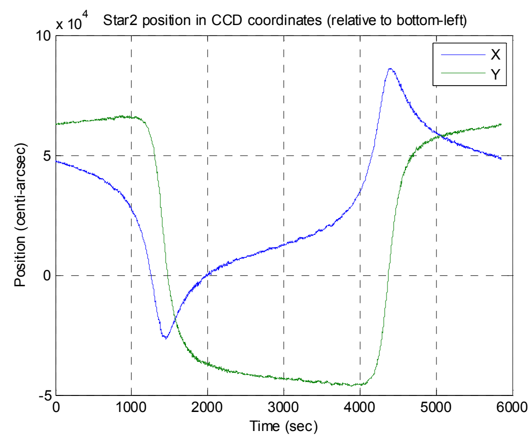

Nominal roll about LoS, relative to Ysat=RA, Zsat=DEC
Angle (deg)
Time (sec)

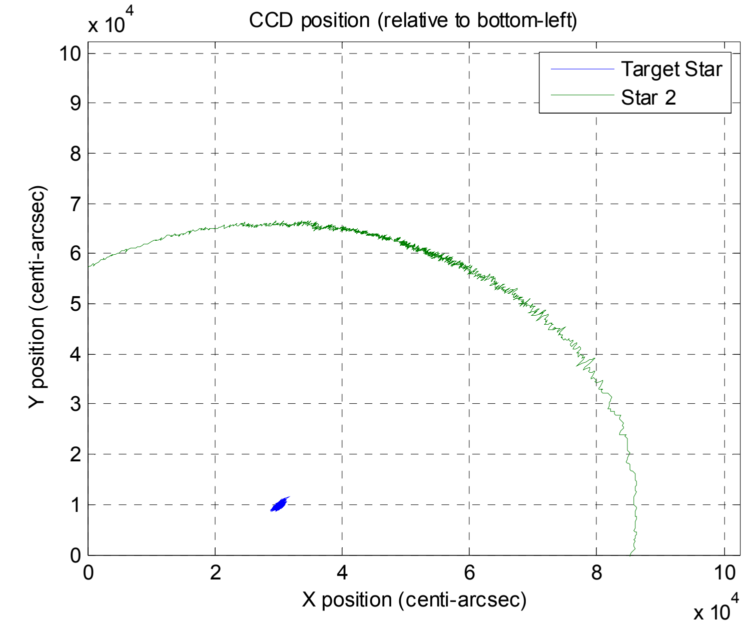

Roll angle input to ffxypx function
Angle (deg)
Time (sec)

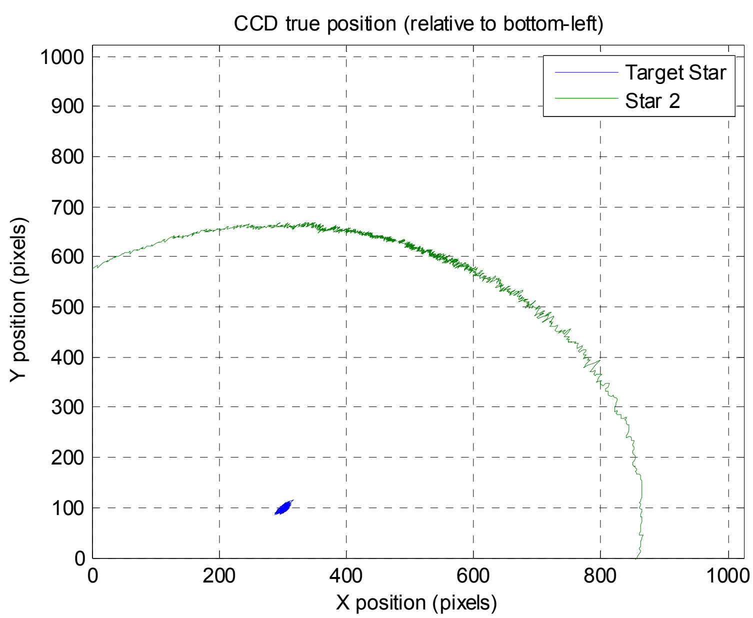

Roll center coordinates
RA
DEC
Angle (deg)
Time (sec)


Roll center input to ffxypx function
xref
yref
Angle (deg)
Time (sec)


x 10^4 Star target position in CCD coordinates (relative to bottom-left)
X
Y
Position (centi-arcsec)
Time (sec)

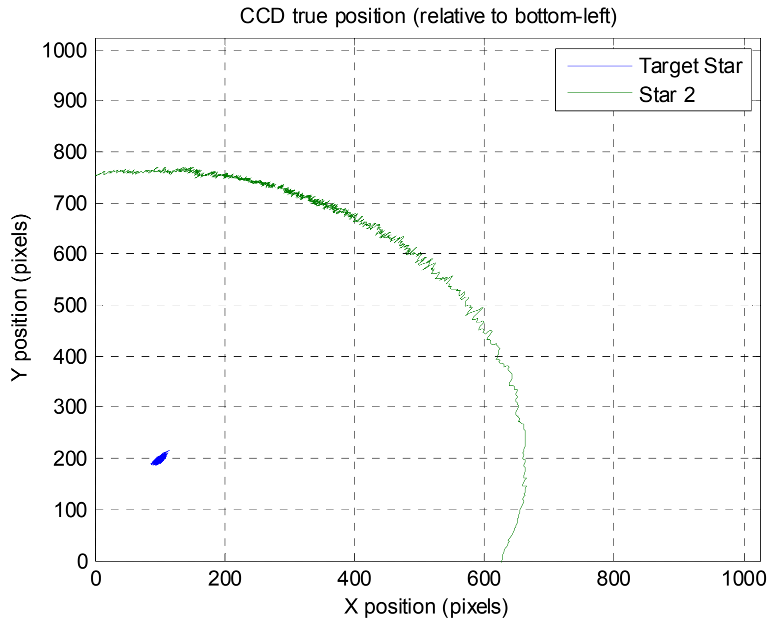

x 10^4
Star2 position in CCD coordinates (relative to bottom-left)
X
Y
Position (centi-arcsec)
Time (sec)





APE polarity validation

With a superimposed additional satellite frame APE of [0, 100, -200] arcsec, for example, we would expect shift of the target star in the CCD frame by -200 pixels in X, and +100 pixels in Y direction. This is confirmed by the result below:


CCD true position (relative to bottom-left)
Target Star
Star 2
Y position (pixels)
X position (pixels)
0
100
200
300
400
500
600
700
800
900
1000
0
200
400
600
800
1000