THEMIS
SCM Calibration Procedure
draft 001

thm_soc_129_scm_calibration
April 12, 2010

1.0 Description of the THM_CAL_SCM procedure:

Note that THM_CAL_SCM is called by THM_LOAD_SCM by default, and that all calibration keywords are available from THM_LOAD_SCM. The following example loads raw data, and then calibrates.

Example:

```
timespan, '2008-08-16'
thm_load_scm, probe='a', level='l1', type='raw', /get_support_data
thm_cal_scm, probe='a'
```

The output of these commands will be tplot variables 'tha_scf', 'tha_scp', and 'tha_scw' fir the given time range, calibrated and in the DSL coordinate system (See ftp://apollo.ssl.berkeley.edu/pub/THEMIS/3%20Ground%20Systems/3.2%20Science%20Operations/Science%20Operations%20Documents/thm_soc_110_COORDINATES_20060929.pdf for descriptions of the different THEMIS coordinate systems. For the general description of the SCM instrument, see http://themis.ssl.berkeley.edu/instrument_scm.shtml .)

The processing is done in steps for individually for each datatype, scf, scp, and scw. (**scf**: fast survey 1/8 second time resolution; **scp**: particle burst 1/128 second time resolution burst mode; **scw**: wave burst 1/8192 second time resolution.)

By default, processing stops after step 5.

step # 0: counts, NaN inserted into each gap for proper tplotting
step # 1: Volts, spinning sensor system, with DC field
step # 2: Volts, spinning sensor system, without DC field [xy DC field in nT]
step # 3: nTesla, spinning sensor system, without DC field
step # 4: nTesla, spinning SSL system, without DC field
step # 5: nTesla, fixed DSL system, without DC field, filtered <fmin
step # 6: nTesla, fixed DSL system, with xy DC field

Data is processed in separate batches, they require differing kernel sizes, based on sample rate and spin rate

Step 0:

a) Find sample rate changes; the convolution kernel size is rate dependent.

b) Find data gaps, working with one sample rate at a time, in the code, the variable iano is the data quality flag, iano=16 is a time reverse (negative step in time), iano=17 is a data gap (discontinuity in time (dt - 1.0/data_rate)>2.0e-5), iano=22 defines a discontinuity in sample rate.
Batches are defined by the different discontinuities, iano > 15. Steps 1 to 6 are done batch by batch, and NaN's are inserted between batches.

For each batch, the process is as follows:

Step 1:

Convert to Volts, The value of the field is multiplied by a calibration factor of: 10.04/2.^16. (A test for saturation is commented out.)

Step 2:

a) Get antenna response at spin frequency, this process calls routines THM_SCM_GAINANT_VEC (returns the gain of the antenna for a given frequency), and THM_SCM_MODPHA (returns the amplitude and phase of the response).

b) Check for abnormally incrementing spin phase; looks for a decrease in spin phase, adds 360.0 to make the negative values positive, the variable bad_sph = where(abs(dph - 360.0*fs/fe) gt 360.0*fs/fe *0.01), where fs is the spin frequency, and fe is the sample frequency for this batch. A warning is issued if there is an odd spin phase, but the program keeps going.

c) Initial centering of the waveforms: The average value of the field ((bx,by,bz)/nbp, where nbp is the number of spins in the batch) is subtracted from the field values.

d) Calculate sine fit for calculation of misalignment angle and classic_despin. Calls the routine THM_SCM_CASINUS_VEC, that computes a pure sine wave from an input signal. The sine wave is then subtracted to get the "useful" signal.

e) Cleanup (if requested) If cleanup = 'spin', then only spin tones are removed, If cleanup = 'full' then spin tones and 8/32 Hz 1s phase locked noise is cleaned up. Calls routine SPIN_TONES_CLEANING_VECTOR_V5, if the parameter clnup_author is 'ole' (default). If clunp_author is 'ccc' then SCM_CLEANUP_CCC is called. The default for cleanup is 'none'.

e.1) SPIN_TONES_CLEANING_VECTOR_V5 process: first the input period is broken up into N_w 'windows' of N_pw points apiece. Each window has the duration of the period of the noise to be cleaned up. (E.g., the spin period for 'spin' cleanup.) The data can be divided into N_w smaller blocks, each with N_pw points. Then the average noise for a window of N_pw points is calculated by successively adding the data in each window into an array of N_pw, and then dividing by the number of windows. In IDL, this process can accomplished by reforming the data into a 2d array and summing over the second index:

```
data = data[0:N_w*N_pw-1]
data_temp = reform(data, N_pw, N_w)
av_data = total(data_temp, 2)/N_w
```

Then the average is repeated N_w times into an array of N_w*N_pw and subtracted from the original data. (Note that the actual IDL code uses a loop rather than the reform call, this may be changed in future versions.)

f) Determine antenna misalignment. First take the spin sine wave (calculated above via THM_SCM_CASINUS_VEC), and divide by the antenna response to get into units of nT. Next the "misalignment angle" is arcsin(B_z*sqrt(2.0)/B), where B_z and B represent the z-component and total value of the sine wave B field. (Note that this sine-wave field eventually forms the basis of the DC field.)

g) If the detrend frequency is not set to zero (which is the default), then a high-pass filter is applied, the field is smoothed over a number of points corresponding to (fe/fdet), where fe is the sampling frequency and fdet is the dtrend frequency, and the smoothed field is subtracted from the overall field.

h) Final centering of the waveform -- the average value of the field is subtracted from the field.

i) The DC field is put into DSL (despun) coordinates: the X,Y amplitude and phase are computed in the scm sensor system. The transformation to SSL is a rotation of about 45 + 12.4 degrees, and must be consistent with the Sensor_to_SSL matrix. The transformation from SSL to DSL is then a rotation of sun pulse phase.

j) Finally, the values of the dc field (only for Bx and By) are saved along with the misalignment angles. NaN values are inserted at the batch edges.

Step 3:

Calibrated data in sensor spinning system [nT] without DC. This first checks to be sure that there are more data points than the number of points in a convolution kernel, then calls the routine THM_SCM_DECONVO_VEC for each component, using the sine-wave subtracted, average subtracted field. THM_SCM_DECONVO_VEC performs a continuous calibration accomplished by convolving a signal with a kernel. The kernel is derived by taking the fft of the inverse of the frequency response. The kernel is also constructed to account for any sample delay.

THM_SCM_DECONVO procedure: i) center the waveform. ii) generate the kernel, THM_SCM_CORGAIN gives the inverse of the antenna gain. THM_SCM_CORETERcorrects for sample delay, THM_SCM_FILTSPE applies a bandpass filter between the cutoff frequency (default value of 1.0 Hz) and the nyquist frequency. Then an inverse FFT to get the kernel, which is the real part of the inverse FFT. A warning is issued of the imaginary part is > 1.0e-5 times the real part. iii) The kernel is shifted by nk/2 (nk = n_elements(kernel)). iv) the kernel is then multiplied by a hanning function. v) the kernel is normalized. vi) THM_SCM_FASTCONVOL is called to do the convolution -- this calls IDL's BLK_CON preferentially over convol, but reverts to CONVOL if there are too few data points.

Step 4:

Calibrated data in SSL spinning system [nT] without DC. Rotates the data to SSL coordinates to SSL coordinates, using the function THM_SCM_SENSOR_TO_SSL, which applies an approximate 57.4 degree rotation:

Step 5:

Calibrated data in DSL system [nT] without DC. Applies rotation of spin phase angle, as in the SSL2DSL routine. This is the end of the default SCM calibration.

Step 6:

Calibrated data in DSL system [nT] with DC. Adds the DC component of B (calculated at the end of step 2) back into the X,Y components.

2.0 Calibration keywords and default values:

keywords:

probe = Probe name. The default is 'all', i.e., calibrate data for all available probes. This can be an array of strings, e.g., ['a', 'b'] or a single string delimited by spaces, e.g., 'a b'

datatype = The type of data to be calibrated/created. The default is 'scf', 'scp', and 'scw', that is, SCM fast mode, particle burst mode and waves burst mode data, with no diagnostic outputs. Each mode of output has its own set of possible diagnostic outputs: 'sc?_dc', 'sc?_misalign' and 'sc?_iano', where ? can be f, p or w. To calibrate all modes loaded and create all diagnostic outputs you can specify 'all'. Diagnostic outputs: raw data in Volts, dc field, axis misalignment (angle between x and y axes in degrees), and iano (data quality flag). Created with '_dc', '_misalign', and '_iano' suffixes, respectively.

in_suffix = optional suffix to add to name of input data quantity, which is generated from probe and datatype keywords.

out_suffix = optional suffix to add to name for output tplot quantity, which is generated from probe and datatype keywords.

trange= array[2] string or double. Limit calibration to specified time range

nk = N points of the cal-convolution kernel, if set, this value will be used regardless of sample rate.

mk = If nk is not set, set nk to (sample frequency)*mk, where sample frequency is determined by the data. default 8 for scf, 4 for scp, and 1 for scw.

despin =classic despin algorithm. 0:off 1:on Default is on.

n_spinfit=n spins to fit for misalignment, dc field calculation and optionally despin.

cleanup= type of cleanup (default is 'none'):
'spin' for only spin tones (power ripples) cleanup,
'full' for spin tones and 8/32 Hz tones
'none' for no cleanup processes.

wind_dur_1s = window duration for 8/32 Hz cleanup (has to be a multiple of 1s); default is 1.

wind_dur_spin = spintone cleanup window duration as a multiple of the spin period; default is 1.

clnup_author = choice of cleanup routine (default is 'ole'):
'ccc' to use scm_cleanup_ccc
'ole' to use spin_tones_cleaning_vector_v5

fdet = detrend freq. in Hz. Detrend is implemented by subtracting a boxcar avg., whose length is determined by fdet. Use fdet=0 for no detrending. Default is fdet=0.
fcut = Low Frequency cut-off for calibration. Default is 0.1 Hz.
fmin = Min frequency for filtering in DSL system. Default is 0 Hz.
fmax = Max frequency for filtering in DSL system. Default is Nyquist.
step = Highest Processing step to complete. Default is step 5.
dfb_butter = correct for DFB Butterworth filter response (default is true)
dfb_dig = correct for DFB Digital fiter response (default is true)
gainant = correct for antenna gain (default is true)
blk_con = use fast convolution for calibration and filtering. Block size for fast convolution will be equal to (value of blk_con) * (size of kernel). blk_con = 8 by default. set blk_con=0 to use brute-force convolution.
edge_truncate, edge_wrap, edge_zero= Method for handling edges in step 3, cal-deconvolution. For usage and exact specification of these keywords, see docs for IDL convol function. Default is to zero data within nk/2 samples of data gap or edge.
coord = coordinate system of output. Step 6 output can only be in DSL.
no_download=don't access internet to check for updated calibration files.
dircal= If set to a string, specifies directory of calibration files. Use /dircal or dircal=" to use calibration files in IDL source distribution.
verbose=set to zero to eliminate output. currently only works with valid_names