IUGONET

Metadata DB for Upper Atmosphere

第3回「太陽地球環境データ解析に基づく超高層大気の空間・時間変動の解明」@NIPR

2016/10/18-20

超高層大気長期変動の全地球ネットワーク データ解析セッション

Inter-university Upper atmospher Global Observation NETwork

CUIの使い方（後編）：calcコマンド、get_dataやstore_dataの使い方、時系列データのフィルター処理、スペクトル/相関解析方法

新堀　淳樹(京大生存研)

1. はじめに

1.1 この講習セクションの目標

- 入門編・CUIの使い方（前編）では、データのロード、プロットの基礎、およびプロットの画像出力方法などを行った。
- CUIの使い方（後編）では…
  - –UDAS上での汎用データ形式である ″tplot変数″ の中身について理解し、各自の手持ちのデータから独自の tplot変数 を生成する方法を学ぶ。
  - –非常に便利なtplot変数を使った演算(足し算、引き算、掛け算、時間微分等)について学ぶ。
  - –移動平均、バンドパスフィルター、周波数スペクトル導出など、よく用いられる時系列解析のやり方を覚える。

- GUIよりCUI(コマンドラインでの操作)の方が自由度が高いことから、UDASに慣れてくるとコマンドを使う方が断然便利である！

1. はじめに

1.2 解析に用いるデータと磁気嵐イベント

- 使用観測データ:

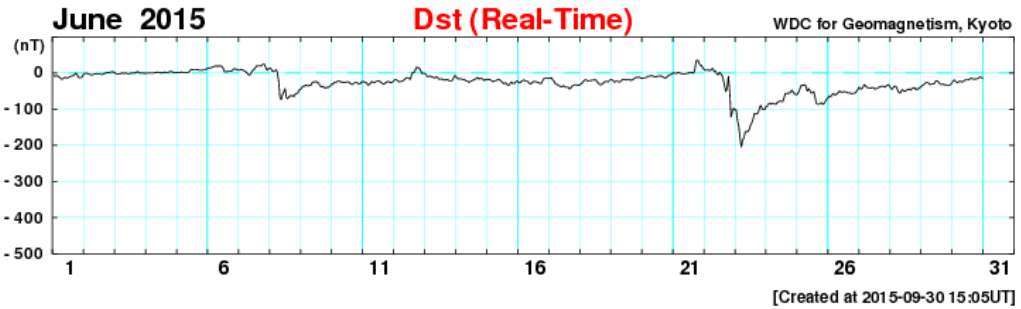
  太陽風・・・ACE衛星

  地磁気データ・地磁気指数・・・WDC Kyoto, 210mm, MAGDAS, AE/SYM

  赤道電離圏データ・・・赤道大気レーダー電離圏観測

- 解析対象イベント

  2015年6月21-25日にかけて発生した磁気嵐(minSYM-H = -109 nT)



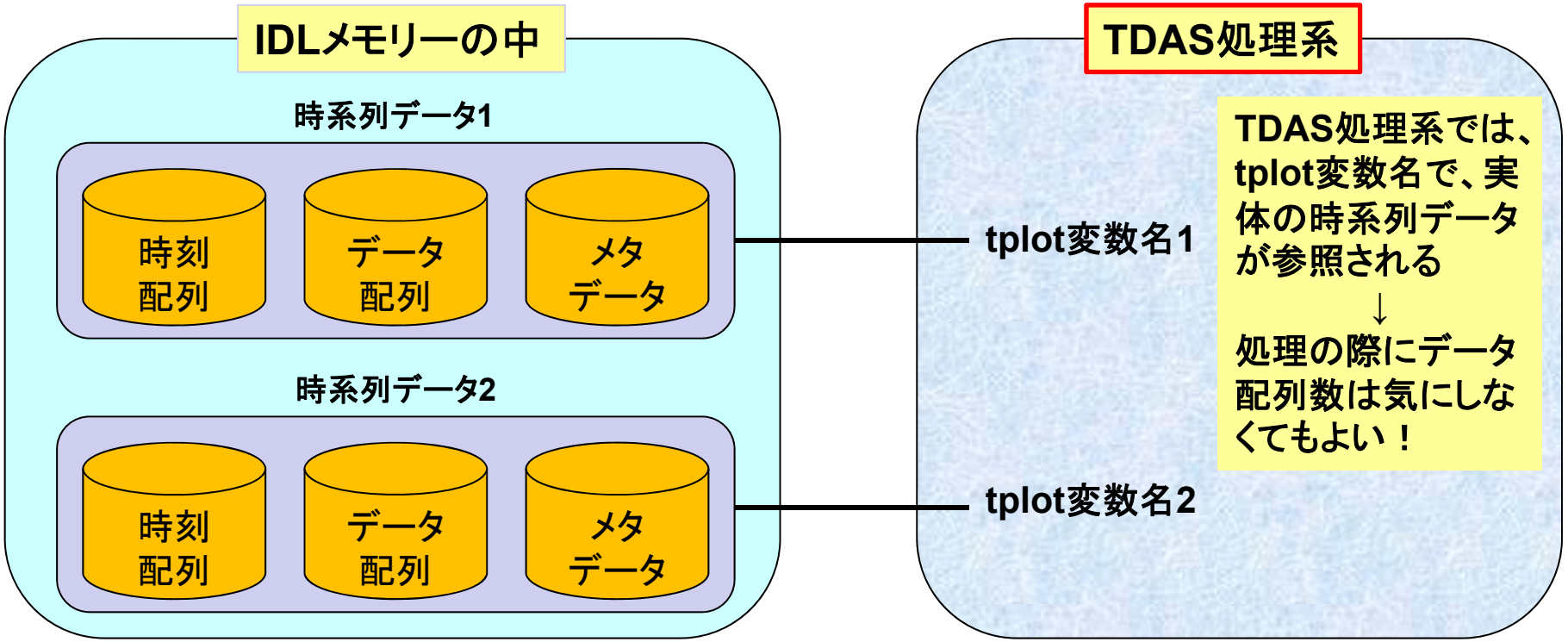
地磁気指数(Dst指数)

赤道大気レーダー

2. tplot変数の取り扱いと演算

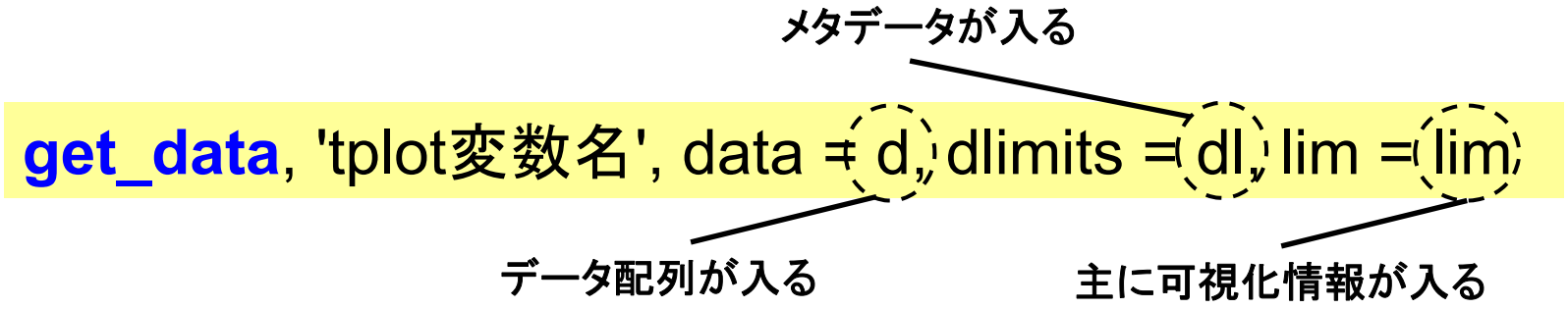
2.1 tplot変数とは

- SPEDAS/UDASのベースになっているTDAS (THEMIS Data Analysis Software)での、汎用時系列データ形式。
- IDL上では単なる文字列であるが、tplot等のいわゆるtコマンドに与えることによって、tplot変数名に紐付けられた時系列データの実体に対して、コマンド処理が実行される。



2. tplot変数の取り扱いと演算

2.2 get_data を用いてtplot変数の中身を見る



※‘tplot変数名’ のところはインデックス番号でも可。その場合はシングルクォーテーションは不要。

```
THEMIS> timespan, '2015-06-21', 5          時間幅として2015年6月21日から5日分を指定
THEMIS> iug_load_ear, datatype = 'e_region', parameter = 'eb3p4g'
                                           赤道大気レーダー電離圏観測データをロード
THEMIS> get_data, 'iug_ear_faieb3p4g_dpl1', data = d, dlimits = dl, lim = lim
THEMIS> help, d, /struct      help コマンドは変数・構造体の情報を表示する。/struct キー
                              ワードを付けると、構造体内の配列情報を表示する。

** Structure <136157a0>, 3 tags, length=1540600, data length=1540600, refs=1 :
   X               DOUBLE    Array[2831]        ⟶  時刻データ(1次元)
   Y               FLOAT     Array[2831, 134]   ⟶  視線方向風データ(2次元)
   V               FLOAT     Array[134]         ⟶  高度データ(1次元)
```

2. tplot変数の取り扱いと演算

2.2 get_data を用いてtplot変数の中身を見る

```
THEMIS> help, d, /struct


** Structure <136157a0>, 3 tags, length=1540600, data length=1540600, refs=1 :
   X               DOUBLE    Array[2831]
   Y               FLOAT     Array[2831, 134]
   V               FLOAT     Array[134]
```

tplot変数の実体のデータ構造体 (今の場合は d) は X, Y, V という3つのメンバーから構成されている。

X: 倍精度浮動小数点で表したUnix time (1970-1-1 00:00:00 UTからの積算秒数)

この例では 2831個の1次元配列。つまりデータのtime frame は2831個ある。このデータは152秒で5日分なので、1日＝86400秒 /152秒　x 5 日分 で 2831。

Y: 温度データが入っている配列

この場合、2831 *134の2次元配列。

V: 高度データが入っている1次元配列

この場合、134の1次元配列。

2. tplot変数の取り扱いと演算

2.2 get_data を用いてtplot変数の中身を見る

```
THEMIS> help, dl, /struct
** Structure <18df09d0>, 1 tags, length=32, data length=32, refs=16:
   DATA_ATT        STRUCT    -> <Anonymous> Array[1]
```

dlimits構造体にはメタデータ(データに関する各種情報)が格納される。

例えば CDF はこれ自体も構造体であり、元データファイルであるCDFファイルの情報(ファイルのセーブ場所など) が格納されている。

```
THEMIS> help, lim, /struct
** Structure <18ee6300>, 3 tags, length=40, data length=34, refs=2:
   YTITLE          STRING    'EAR-iono!CHeight!C[km]'
   ZTITLE          STRING    'snr1!C[dB]'
   SPEC            INT              1
```

lim 構造体の方には主にプロット等に可視化する際に必要な情報が入っている。

例えば tplot コマンドがtplot変数をプロットする場合、ここの情報を参照して、線の色や縦軸のラベル、凡例 等を描画する。

2. tplot変数の取り扱いと演算

2.3 store_dataで新規tplot変数を作成

store_data, 'tplot変数名', data = {x:time, y:data1, v:data2}, dlimits = dl, lim = lim

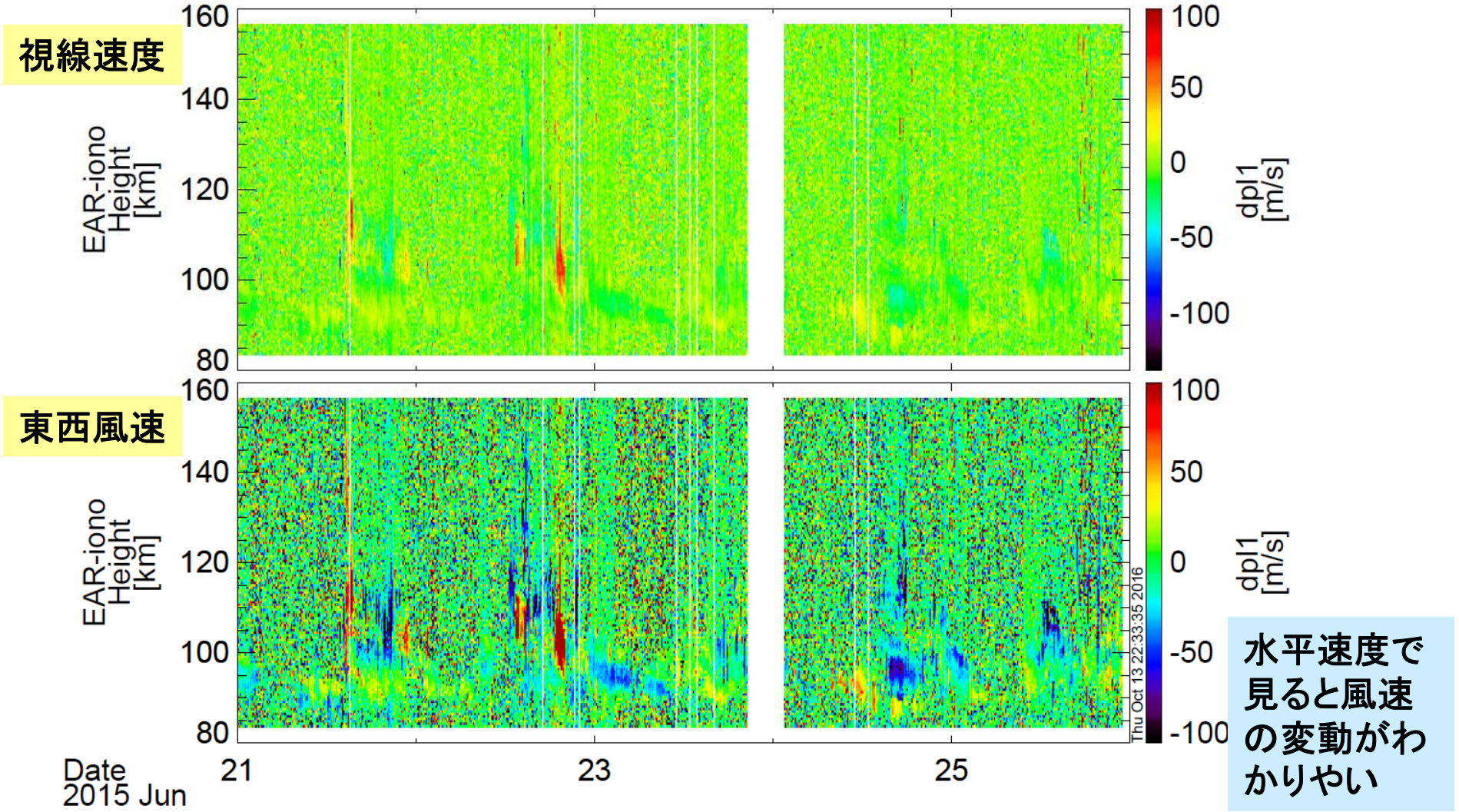
time: データの時刻ラベルを倍精度浮動小数点のUnix time の配列にしたもの。
　　1次元配列 [N]　N: 時刻ラベル数
val: データの配列。
　　スカラーデータの場合は [N] (timeと同じサイズ)、1次元ベクトルデータの場合は [N][J] (J がベクトルの成分数) という配列。
というような time, val を用意すればtplot変数を作成できる。

```
THEMIS> time = d.x
THEMIS> h_wind = -1.0 * d.y * cos(152.9*!pi/180)/sin(23.0*!pi/180)
THEMIS> height = d.v
THEMIS> store_data, 'iug_ear_faieb3p4g_dpl1_h', data = { x:time, y:
h_wind, v: height}, dlimits = dl, lim = lim 実際にtplotでプロットして確認してみる
THEMIS> zlim, 'iug_ear_faieb3p4g_dpl1_h', -100, 100
THEMIS> tplot, ['iug_ear_faieb3p4g_dpl1', 'iug_ear_faieb3p4g_dpl1_h' ]
```

2. tplot変数の取り扱いと演算

2.3 store_dataで新規tplot変数を作成

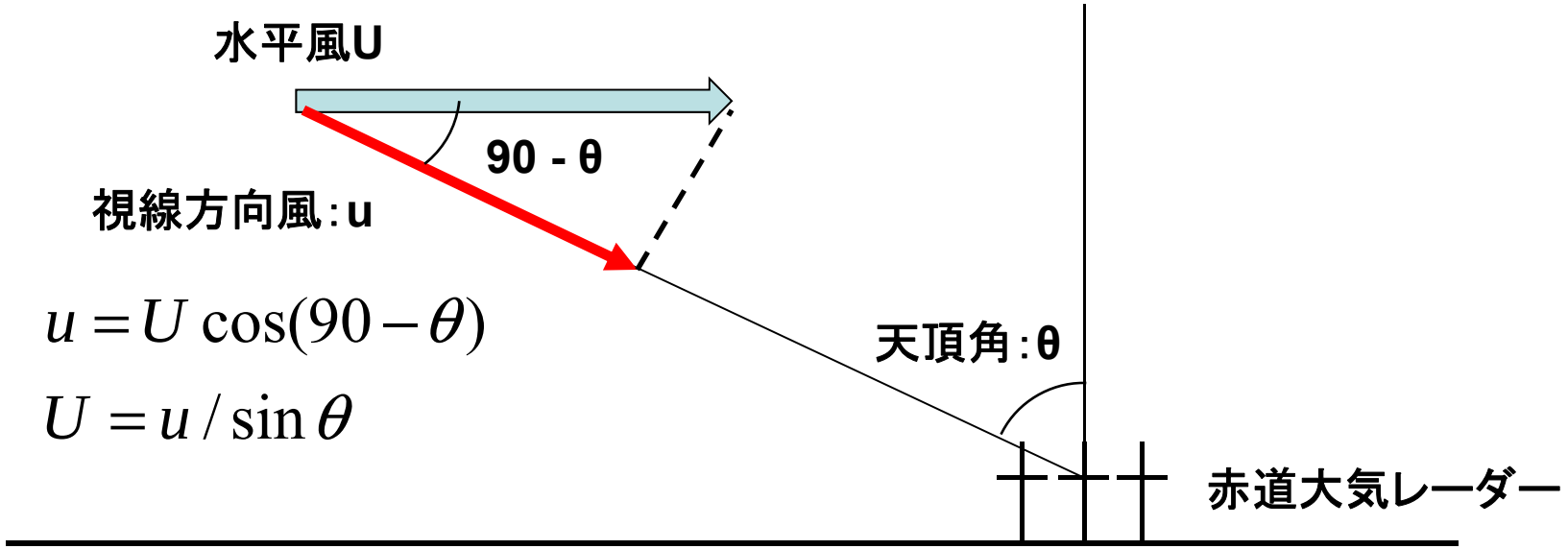
```
THEMIS> tplot, ['iug_ear_faieb3p4g_dpl1', 'iug_ear_faieb3p4g_dpl1_h' ]
```



水平速度で見ると風速の変動がわかりやい

2. tplot変数の取り扱いと演算

2.3 store_dataで新規tplot変数を作成



観測パラメータ：faieb3p4g
ビーム数：3
(天頂角、方位角) = (152.9, 23.0), (180.0, 20.8), (207.1, 23.2)

ビーム方向2と3についても同様の計算が可能

2. tplot変数の取り扱いと演算

2.4 calcコマンドによるtplot変数の演算

calc, ' "新tplot変数名" = … 計算式 … '

(例) calc, ' "newvar" = "iug_ear_faieb3p4g_dpl1" *(-1.0)*
cos(152.9*3.14159/180)/sin(23.0*3.14159/180)'

時系列データであるtplot変数全体を使った演算を、直感的にわかり易い形で書いて実行することができる！

実は、前頁のstore_data を使ってやったことは、

calc, ' "iug_ear_faieb3p4g_dpl1_h" = "iug_ear_faieb3p4g_dpl1" *(-1.0)*
cos(152.9*3.14159/180)/sin(23.0*3.14159/180)'

と、わずか1行で実行できる！

2. tplot変数の取り扱いと演算

2.4 calcコマンドによるtplot変数の演算

calc, ' "新tplot変数名" = … 計算式 … '

(例) calc, ' "newvar" = "iug_ear_faieb3p4g_dpl1" *(-1.0)* cos(152.9*3.14159/180)/sin(23.0*3.14159/180)'

計算式のルール

- フォーマットは普通の計算式と同じ。全体を単引用符(') で囲む。tplot変数は二重引用符(") で囲む。
- 使用可能な演算: 四則(+-*/), べき乗, sin/cos/tan(), exp(), log(), abs(), min(), max(), total(), mean(), median(), …

注意点

- 複数のtplot変数を演算に使う場合、実体の配列のサイズ・次元が同一でないといけない。データの時刻数が異なる、データの次元が異なる(スカラーデータとベクトルデータの混在など)とエラーになる。
- calcを用いて新しいtplot変数を生成した場合、前に含んでいたデータの基本情報を引き継がないので、改めてoptionコマンドで適宜入れる。

2. tplot変数の取り扱いと演算

2.4 calcコマンドによるtplot変数の演算

calcコマンドでサポートされている関数群とオペレータの種類を確認する方法

```
THEMIS>  calc, function_list=f, operator_list=o
THEMIS> print, 'Functions: ',f
Functions:  log(x[,base]) ln(x) exp(x[,base]) sqrt(x) abs(x) min(x,[,dim][,/nan])
max(x,[,dim][,/nan]) mean(x,[,dim][,/nan]) median(x,[,dim][,/even]) total(x,[,dim][,/nan]
[,/cumulative]) count(x,[,dim]) sin(x) arcsin(x) sinh(x) arcsinh(x) cos(x) arccos(x)
cosh(x) arccosh(x) tan(x) arctan(x) tanh(x) arctanh(x) csc(x) arccsc(x) csch(x)
arccsch(x) sec(x) arcsec(x) sech(x) arcsech(x) cot(x) arccot(x) coth(x) arccoth(x)
```

使用可能な関数・演算:

指数・対数関数、平方根、絶対値、最小、最大、平均、中央値、総和、個数、三角関数、双曲線関数、それらの逆関数

常用対数：log、自然対数：ln

2. tplot変数の取り扱いと演算

2.4 calcコマンドによるtplot変数の演算

calcコマンドでサポートされている関数群とオペレータの種類を確認する方法

```
THEMIS>  calc, function_list=f, operator_list=o
THEMIS> print, 'Operators: ',o
Operators:  ~ ++ -- - + * / ^ < > && || # ## mod and eq ge gt le lt or xor +$
```

使用可能な演算子:

~ (論理演算子)・・・否定、++ -- (算術演算子)・・・インクリメント・デクリメント、
+ - * / (算術演算子)・・・加減乗除、^ (算術演算子)・・・冪乗、
< > (最大・最小演算子)・・・最小・最大、&& (論理演算子)・・・論理積、
|| (論理演算子)・・・論理和、# ## (行列演算子)・・・行列の掛け算(BA、AB)、
mod (算術演算子)・・・剰余、and (ビット演算子)・・・ビット単位AND 、
eq、ge、gt、le、lt (関係演算子) ・・・ A=B、A>=B、A>B、A<=B、A<B、
or (ビット演算子)・・・ビット単位OR、xor (ビット演算子)・・・ビット単位XOR、
+$ ・・・改行

2. tplot変数の取り扱いと演算

2.4 calcコマンドによるtplot変数の演算

課題：ほかの2ビーム方向から求めた東西風を求め、それらを並列プロットする

観測パラメータ：faieb3p4g
ビーム数：3
(天頂角、方位角) = (152.9, 23.0), (180.0, 20.8), (207.1, 23.2)

①各ビームごとの東西風の計算

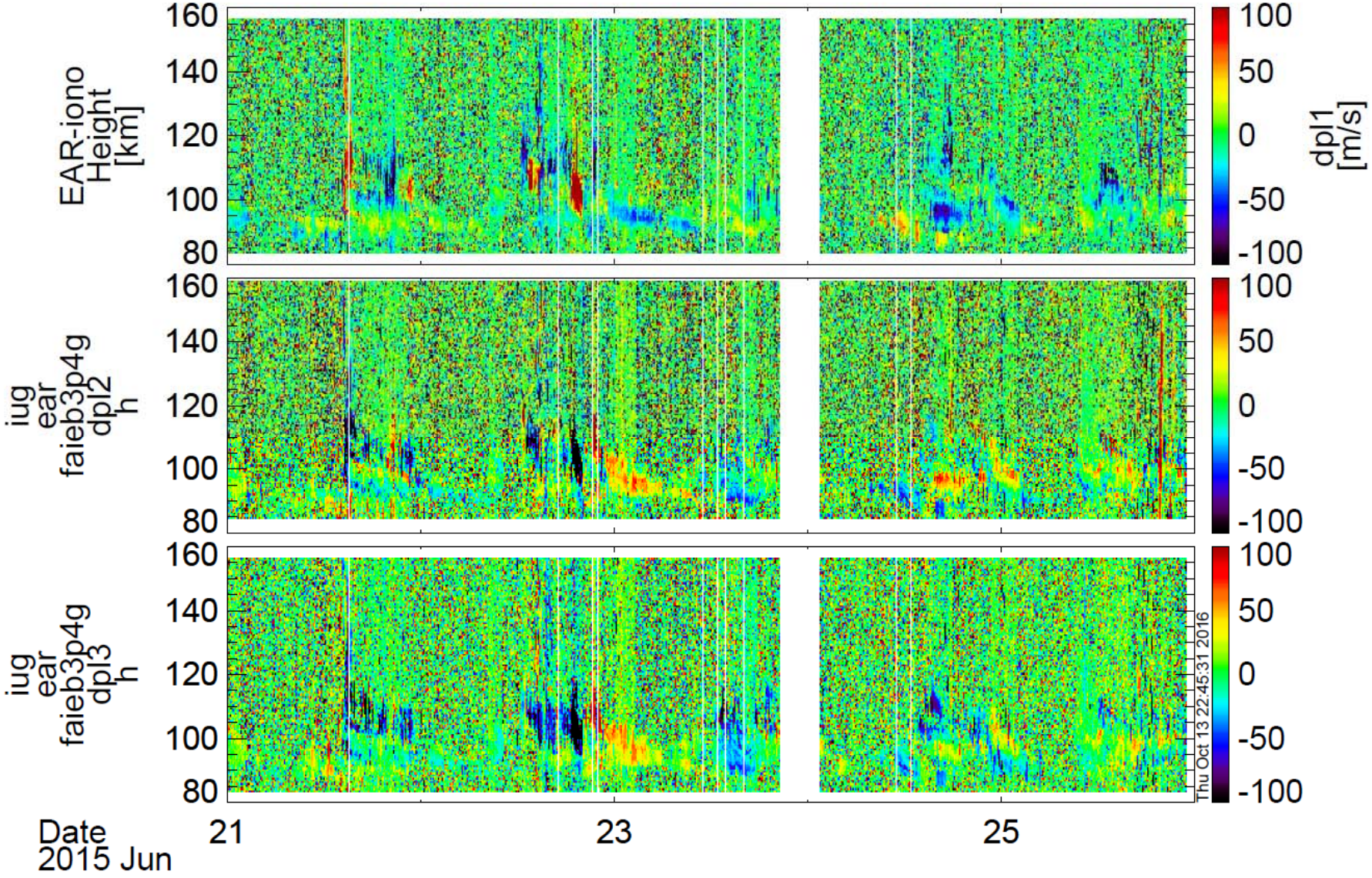
```
THEMIS>  calc, ' "iug_ear_faieb3p4g_dpl2_h" = "iug_ear_faieb3p4g_dpl2" *
cos(180.0*3.14159/180)/sin(20.8*3.14159/180)'
THEMIS> calc, ' "iug_ear_faieb3p4g_dpl3_h" = "iug_ear_faieb3p4g_dpl3" *
cos(207.1*3.14159/180)/sin(23.2*3.14159/180)'
```

②3つの東西風のカラーバーの範囲の調整

```
THEMIS>  zlim, ['iug_ear_faieb3p4g_dpl2_h', 'iug_ear_faieb3p4g_dpl3_h'], -100,
100
 THEMIS> tplot, ['iug_ear_faieb3p4g_dpl1_h', 'iug_ear_faieb3p4g_dpl2_h',
'iug_ear_faieb3p4g_dpl3_h']
```

2. tplot変数の取り扱いと演算

2.4 calcコマンドによるtplot変数の演算



2. tplot変数の取り扱いと演算

2.4 calcコマンドによるtplot変数の演算

calcを使うと、各tplot変数に付帯する情報を引き継げないので、optionsコマンドで適宜、軸などのキャプションや単位は変更する。

●軸のキャプションなどの設定

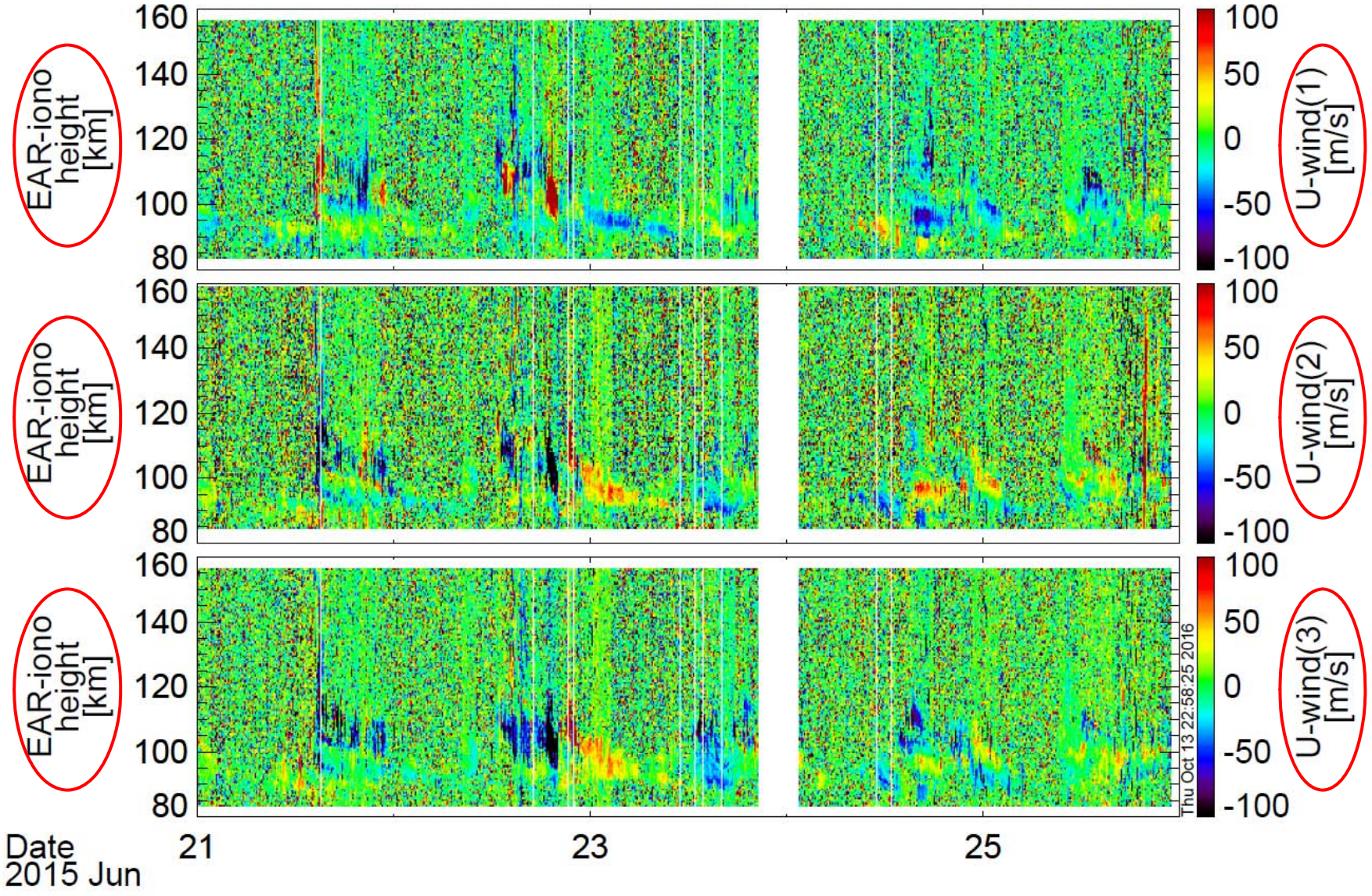
```
THEMIS> options, 'iug_ear_faieb3p4g_dpl1_h', ytitle = 'EAR-iono!Cheight!C[km]', ztitle = 'U-wind(1)!C[m/s]'
THEMIS> options, 'iug_ear_faieb3p4g_dpl2_h', ytitle = 'EAR-iono!Cheight!C[km]', ztitle = 'U-wind(2)!C[m/s]'
THEMIS> options, 'iug_ear_faieb3p4g_dpl2_h', ytitle = 'EAR-iono!Cheight!C[km]', ztitle = 'U-wind(3)!C[m/s]'
```

●プロットの生成

```
THEMIS> tplot, ['iug_ear_faieb3p4g_dpl1_h', 'iug_ear_faieb3p4g_dpl2_h', 'iug_ear_faieb3p4g_dpl3_h']
```

2. tplot変数の取り扱いと演算

2.4 calcコマンドによるtplot変数の演算



2. tplot変数の取り扱いと演算

2.5 calcコマンドの応用

電離圏Pedersen, Hall伝導度からCowling電気伝導度を導出

calc, ' "sigmaC" = "sigmaP" + ("sigmaH" ^2 / "sigmaP")'

注) sigmaP: Pedersen伝導度、sigmaH: Hall伝導度

$$\Sigma_C = \Sigma_P + \frac{\Sigma_H^2}{\Sigma_P}$$

太陽風観測から太陽風動圧を導出

calc, ' "Pdyn" = "ace_Np" * "ace_Vp"^2 * 1.6726 * 1e-6 '

注) ace_Np: 太陽風密度 [/cc]、ace_Vp: 太陽風速度 [km/s]　←　プロトンの質量

$$P_{dyn} = N_p * M * V_p^2$$

2つ目の例のace_Np, ace_Vp というデータは、TDASに収録されている ace_swe_load, datatype='h0' というコマンドでロードできる。

3. tplot変数を用いた各種データ解析

3.1 tsub_average で平均値を差し引く

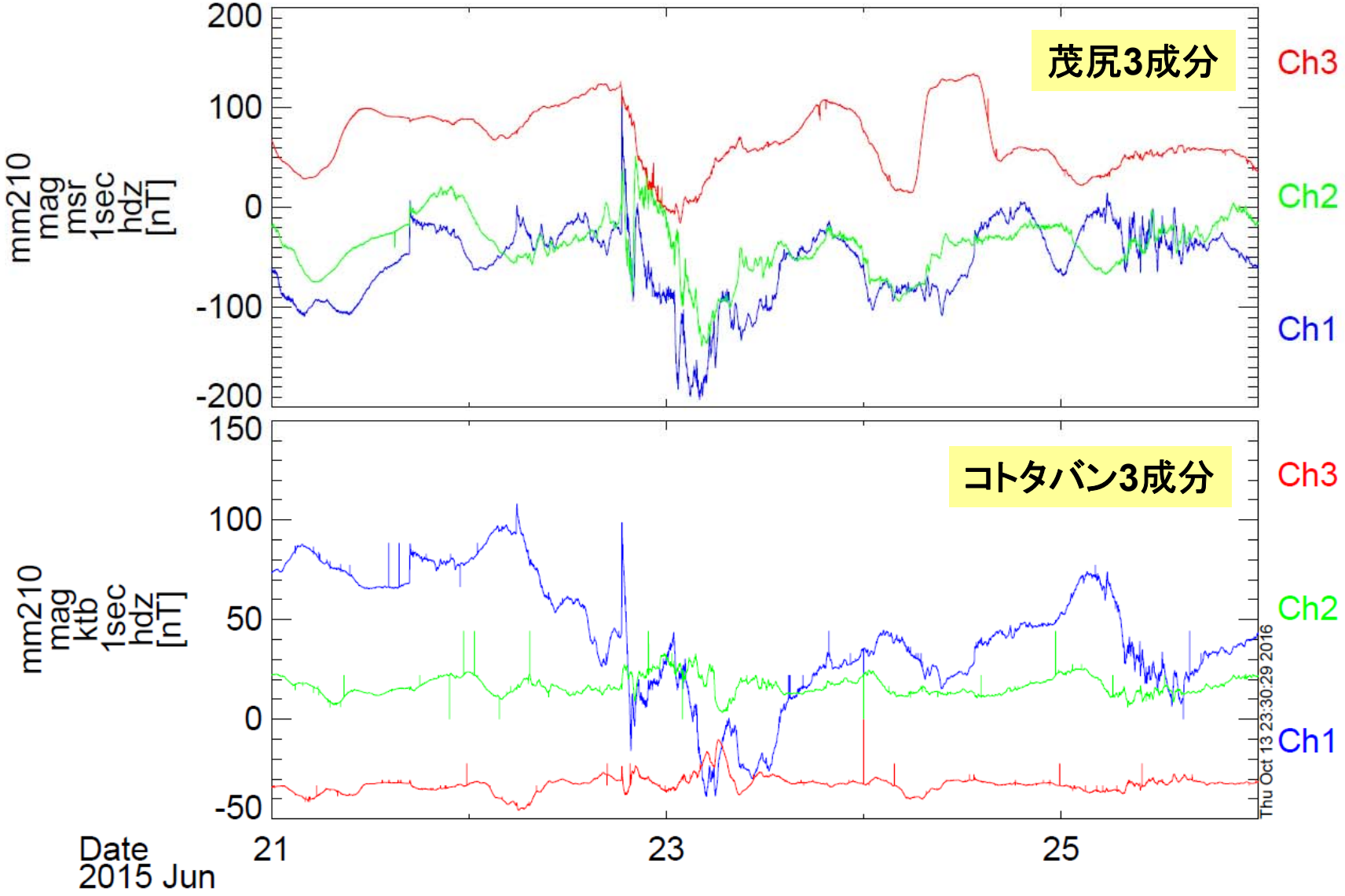
210度地磁気観測点(茂尻とコトタバン)データのロード

```
THEMIS> timespan, '2015-06-21', 5
THEMIS> iug_load_gmag_mm210, site =['msr', 'ktb' ]

THEMIS> tplot_names
        23 mm210_mag_msr_1sec_hdz  ⟶  茂尻の1秒値
        24 mm210_mag_msr_1min_hdz  ⟶  茂尻の1分値
        25 mm210_mag_msr_1h_hdz    ⟶  茂尻の1時間値
        26 mm210_mag_ktb_1sec_hdz  ⟶  コトタバンの1秒値
        27 mm210_mag_ktb_1min_hdz  ⟶  コトタバンの1分値
        28 mm210_mag_ktb_1h_hdz    ⟶  コトタバンの1時間値
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz', 'mm210_mag_ktb_1sec_hdz']
```

3. tplot変数を用いた各種データ解析

3.1 tsub_average で平均値を差し引く



3. tplot変数を用いた各種データ解析

3.1 tsub_average で平均値を差し引く

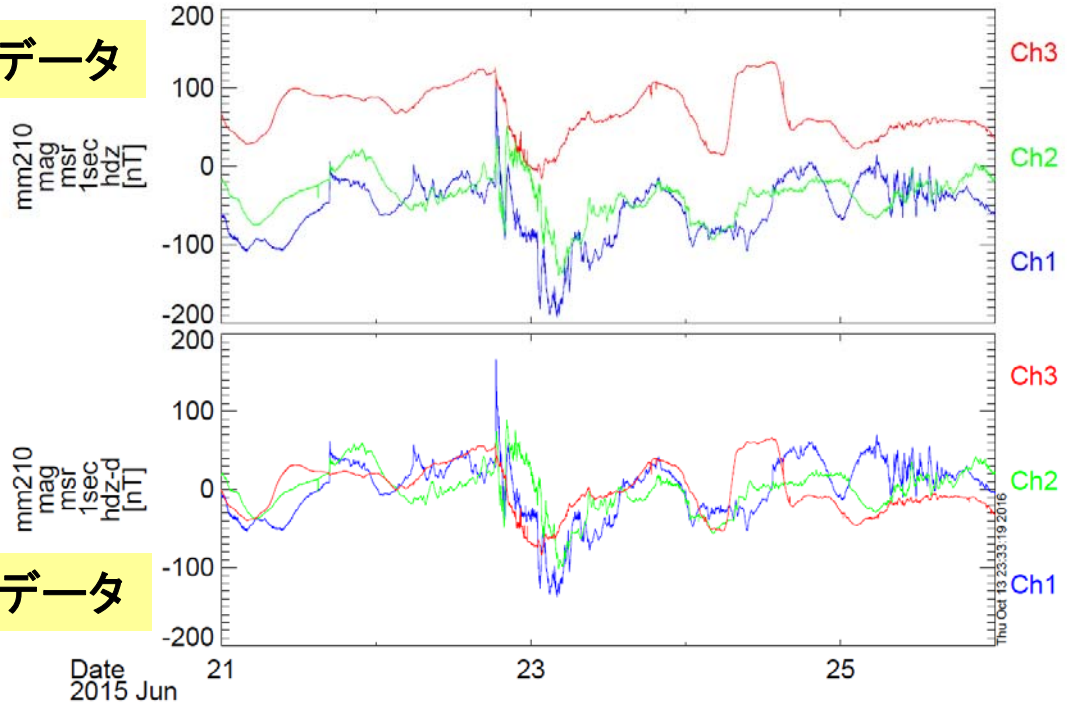
tsub_average, 'tplot変数名'

(例) tsub_average, 'mm210_mag_msr_1sec_hdz'

```
THEMIS> tsub_average, 'mm210_mag_msr_1sec_hdz'
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz',
'mm210_mag_msr_1sec_hdz-d']
```

・元の変数名に -d を付けた新しいtplot変数に結果が格納される。

・プロットする際にゼロ線を揃えたり周波数解析の前処理などで多用される。



3. tplot変数を用いた各種データ解析

3.2 deriv_data で時間微分値を計算

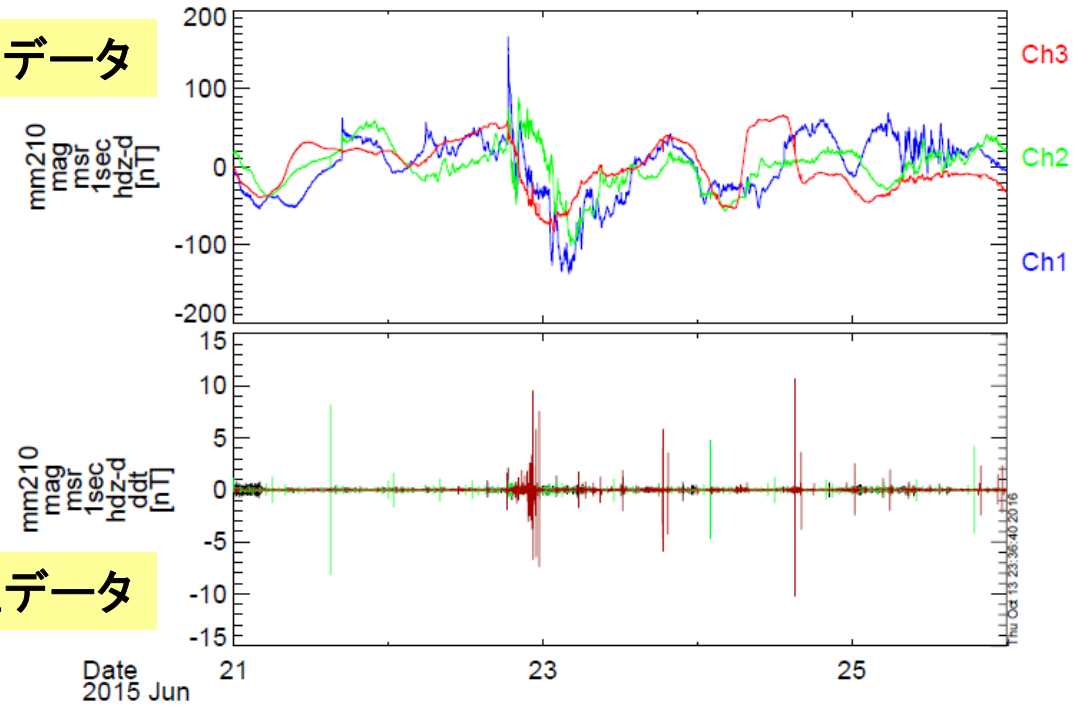
deriv_data, 'tplot変数名'

(例) deriv_data, 'mm210_mag_msr_1sec_hdz-d'

```
THEMIS> deriv_data, 'mm210_mag_msr_1sec_hdz-d'
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz-d',
'mm210_mag_msr_1sec_hdz-d_ddt']
```

・元の変数名に _ddt を付けた新しいtplot変数に結果が格納される。

・スペクトル解析をする前に必要に応じてスパイク処理を行う。



3. tplot変数を用いた各種データ解析

3.3 clean_spikesでスパイク状のノイズ除去

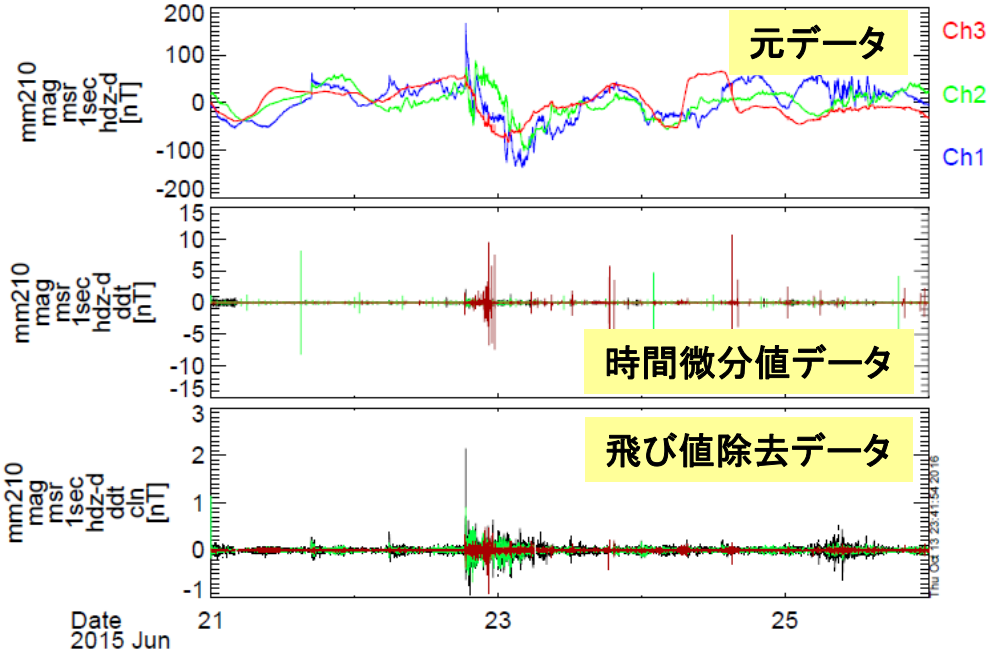
clean_spikes, 'tplot変数名', nsmooth=60, thresh=5

(例) deriv_data, 'mm210_mag_msr_1sec_hdz-d_ddt_cln'

```
THEMIS> clean_spikes, 'mm210_mag_msr_1sec_hdz-d_ddt', nsmooth= 60, thresh = 5
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz-d', 'mm210_mag_msr_1sec_hdz-d_ddt', 'mm210_mag_msr_1sec_hdz-d_ddt_cln']
```

・元の変数名に _cln を付けた新しいtplot変数に結果が格納される。

・clean_spikesの引数(nsmooth, thresh)は状況に応じて変える。規定値はそれぞれ、3と10になっている。



4. 周波数スペクトル解析

4.1 フーリエスペクトル解析 tdpwrspc

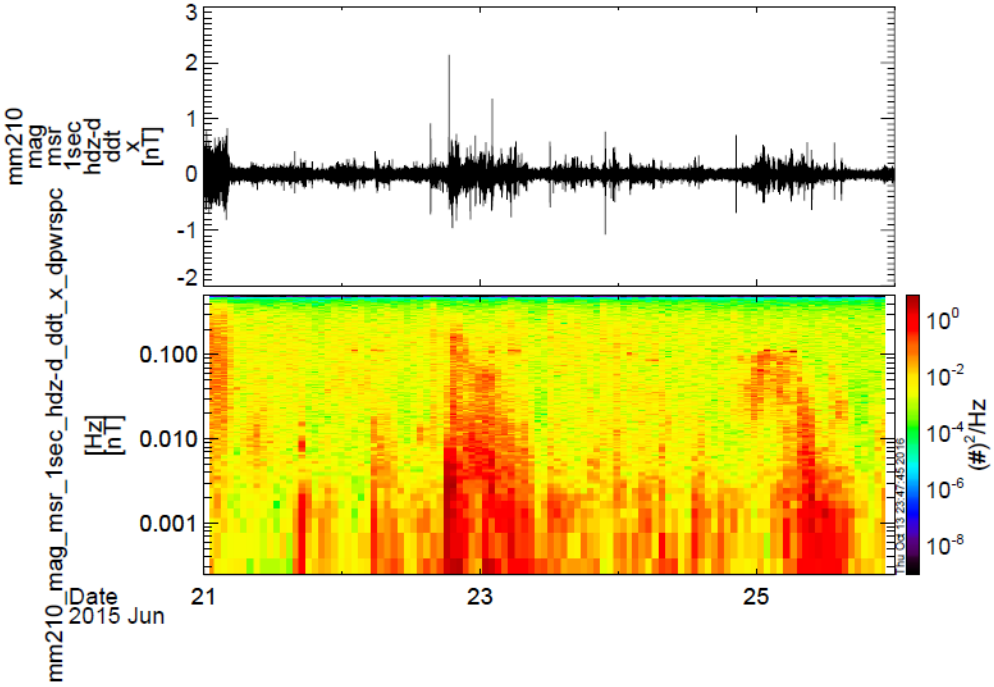
tdpwrspc, 'tplot変数名'

窓幅のデータ点数、ハニング窓を使う/使わない、など色々オプションがある

(例) tdpwrspc, 'mm210_mag_msr_1sec_hdz-d_ddt'

```
THEMIS> tdpwrspc, 'mm210_mag_msr_1sec_hdz-d_ddt'
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz-d_ddt_x',
'mm210_mag_msr_1sec_hdz-d_ddt_x_dpwrspc']
```

- ハニング窓+FFTでダイナミックスペクトル求め, …_dpwrspc という名前のtplot変数に結果を格納する。
- tplotによりカラーコンターでプロットされる。コンターの単位は元の値の単位の2乗/Hz (元: dB ⇒ dB^2/Hz)
- 縦軸のキャプションは、optionsコマンドで適宜修正する。



4. 周波数スペクトル解析

4.1 フーリエスペクトル解析 tdpwrspc

tdpwrspcを使うと、各tplot変数に付帯する情報を正しく引き継げないので、optionsコマンドで適宜、軸などのキャプションや単位は変更する。

●軸のキャプションなどの設定

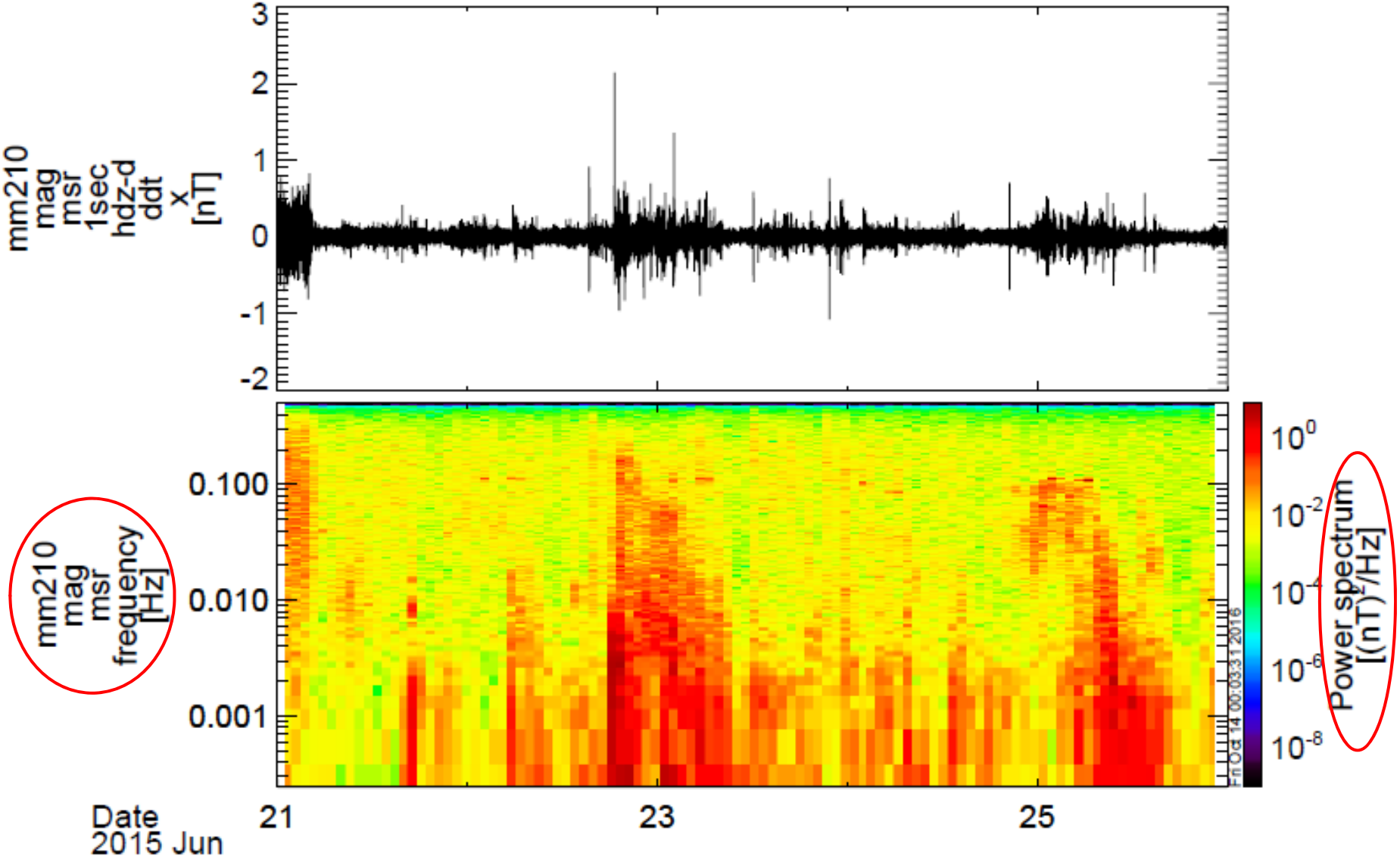
```
THEMIS> options, 'mm210_mag_msr_1sec_hdz-d_ddt_x_dpwrspc', ytitle =
'mm210!Cmag!Cmsr!Cfrequency',ysubtitle = '[Hz]', ztitle = 'Power
spectrum!C[(nT)!U2!N/Hz]'
```

●プロットの生成

```
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz-d_ddt_x',
'mm210_mag_msr_1sec_hdz-d_ddt_x_dpwrspc']
```

4. 周波数スペクトル解析

4.1 フーリエスペクトル解析 tdpwrspc



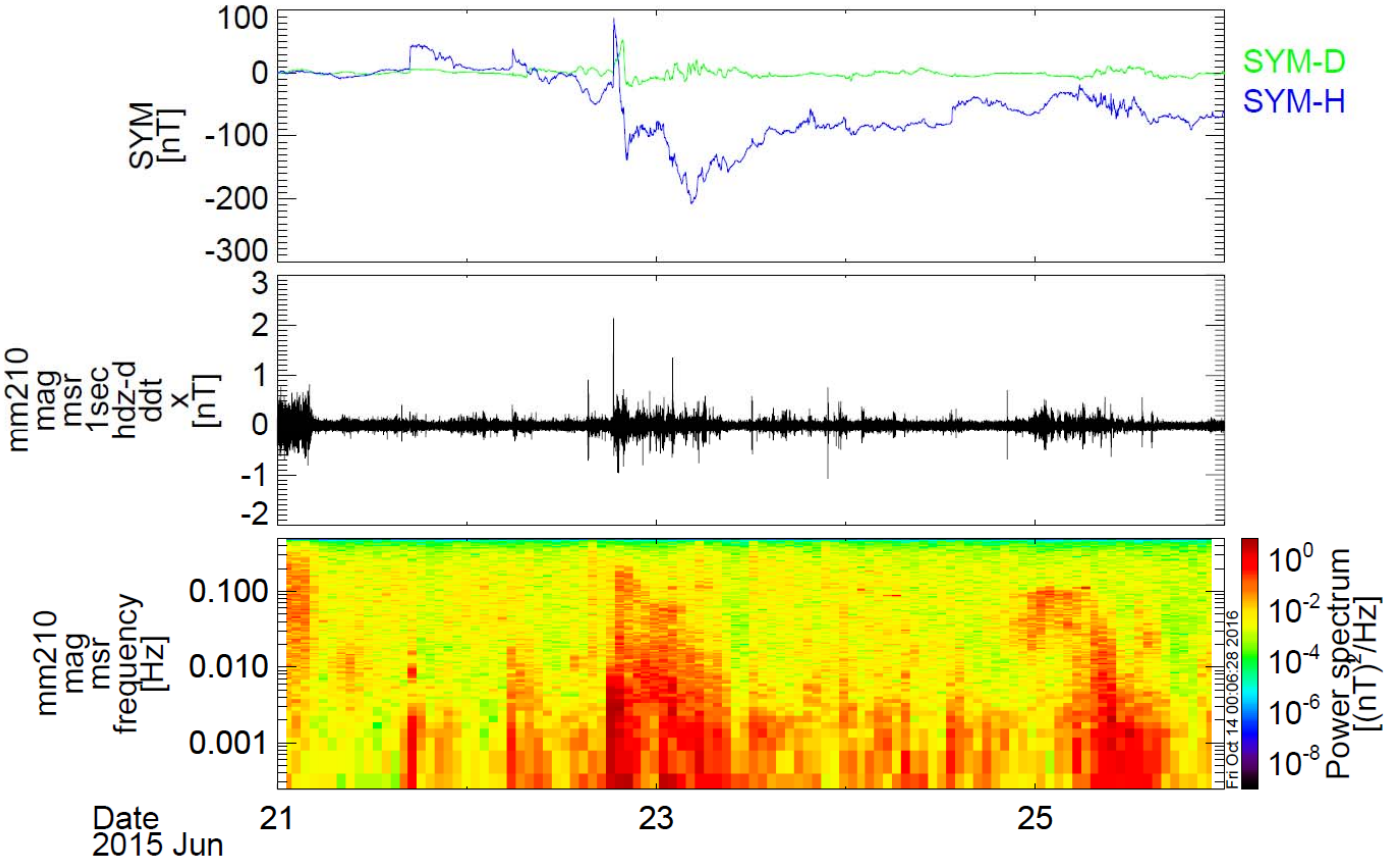
4. 周波数スペクトル解析

4.1 フーリエスペクトル解析 tdpwrspc

京大地磁気指数(SYM)のデータを追加してみる

THEMIS> tplot, ['wdc_mag_sym','mm210_mag_msr_1sec_hdz-d_ddt_x', 'mm210_mag_msr_1sec_hdz-d_ddt_x_dpwrspc']

- **磁気嵐主相時において低周波から高周波のスペクトルが卓越していることがわかる。**
- **回復相(6/25あたり)には高周波(0.05-0.1Hz)のスペクトルが見える**



4. 周波数スペクトル解析

4.2 ウェーブレット変換 wav_data

wav_data, 'tplot変数名'

Wavelet変換で周波数スペクトルを求める

(例) wav_data, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip'

```
THEMIS> time_clip, 'mm210_mag_msr_1sec_hdz-d_ddt_x', '2015-06-22/15:00',
'2015-06-23/00:00'
THEMIS> wav_data, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip'
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz-d_ddt_x_tclip',
'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_wv_pow']
```

時間の切り出し

- ウェーブレット変換を用いるので、tdpwrspcよりは速い時間変動にも追随できる。
- その代わり処理に時間がかかるので、1度に変換するのは3万点程度にした方がよい。

特定の時間だけデータを切り出す方法

time_clip, 'tplot変数名', '開始時刻', '終了時刻'

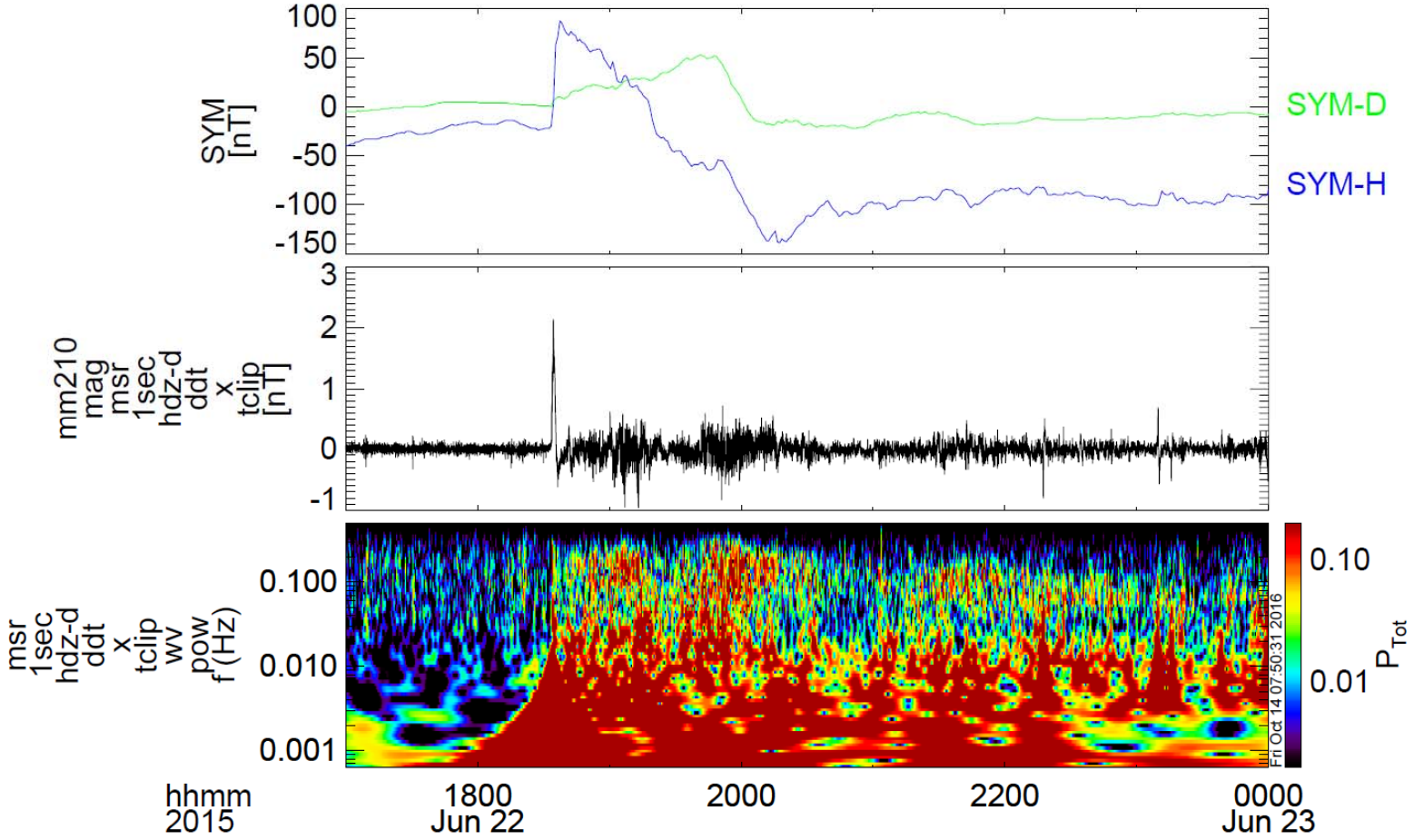
(例) time_clip, 'mm210_mag_msr_1sec_hdz-d_ddt_x', '2015-06-22/15:00', '2015-06-23/00:00'

4. 周波数スペクトル解析

4.2 ウェーブレット変換 wav_data

THEMIS> tlimit, '2015-06-22 17:00', '2015-06-23 00:00'

THEMIS> tplot, ['wdc_mag_sym','mm210_mag_msr_1sec_hdz-d_ddt_x_tclip', 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_wv_pow']



4. 周波数スペクトル解析

4.3 S(Stockwell)変換 ustrans_pwrspc

ustrans_pwrspc, 'tplot変数名', /sampling, /abs

S変換で周波数スペクトルを求める

(例) ustrans_pwrspc, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip', /sampling, /abs

```
THEMIS> time_clip, 'mm210_mag_msr_1sec_hdz-d_ddt_x', '2015-06-22/18:00', '2015-06-22/20:00'
THEMIS> ustrans_pwrspc, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip',/sampling, /abs
THEMIS> options, ' mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_stpwrspc',ytitle = 'mm210-msr!CPeriod', ysubtitle = '[sec]'
THEMIS> ylim, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_stpwrspc', 0, 300
THEMIS> zlim, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_stpwrspc', 0, 0.2
THEMIS> tplot, [‘wdc_mag_sym’, ‘mm210_mag_msr_1sec_hdz-d_ddt_x’, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_stpwrspc']
```

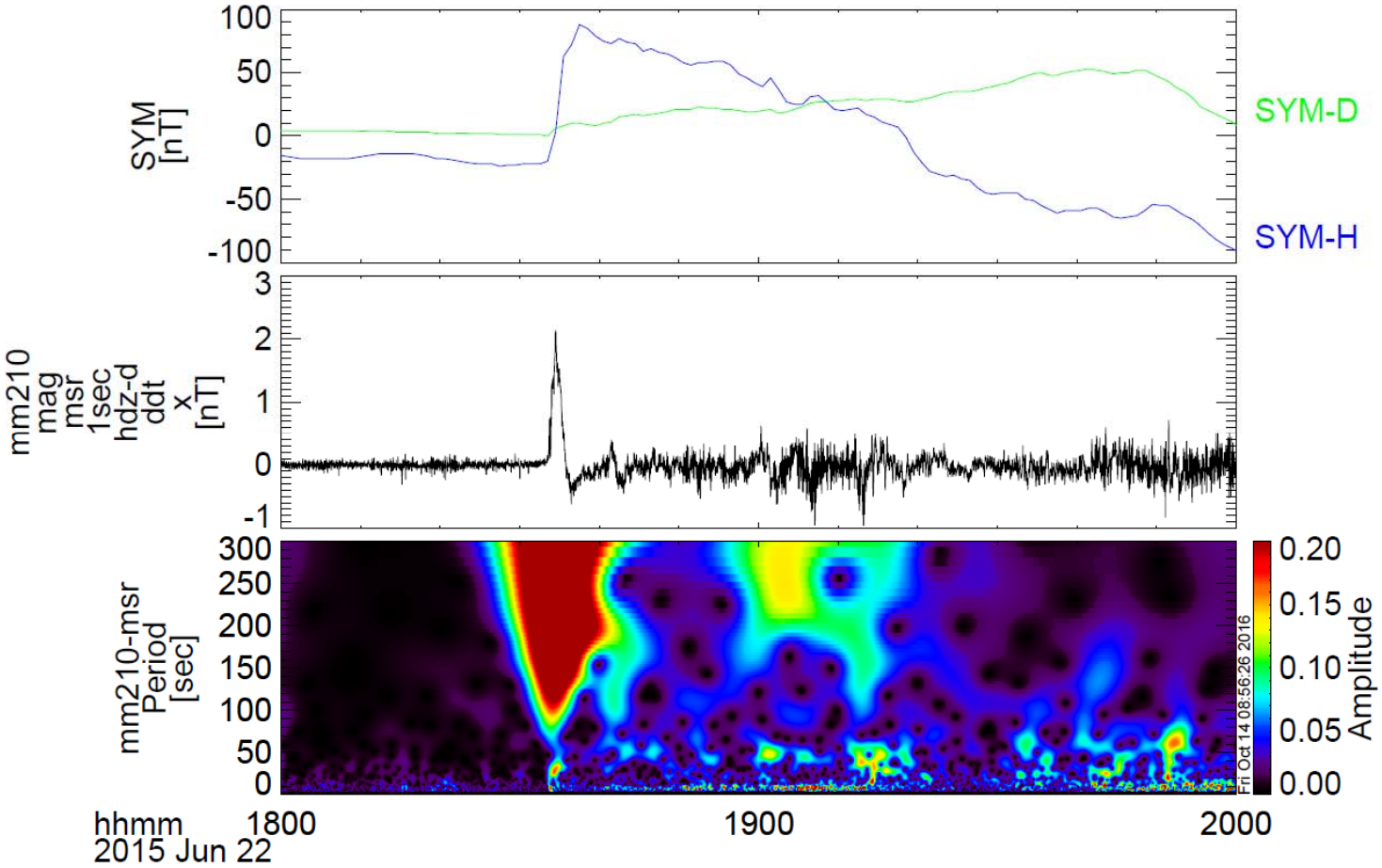
- **引数/absの代わりに/powerとすると、振幅ではなくパワー値を算出する。**
- **処理に時間がかかるので、1度に変換するのは1万点程度にした方がよい。**

4. 周波数スペクトル解析

4.3 S(Stockwell)変換 ustrans_pwrspc

THEMIS> tlimit, '2015-06-22 18:00', '2015-06-22 20:00'

THEMIS> tplot, ['wdc_mag_sym','mm210_mag_msr_1sec_hdz-d_ddt_x', '
mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_stpwrspc']

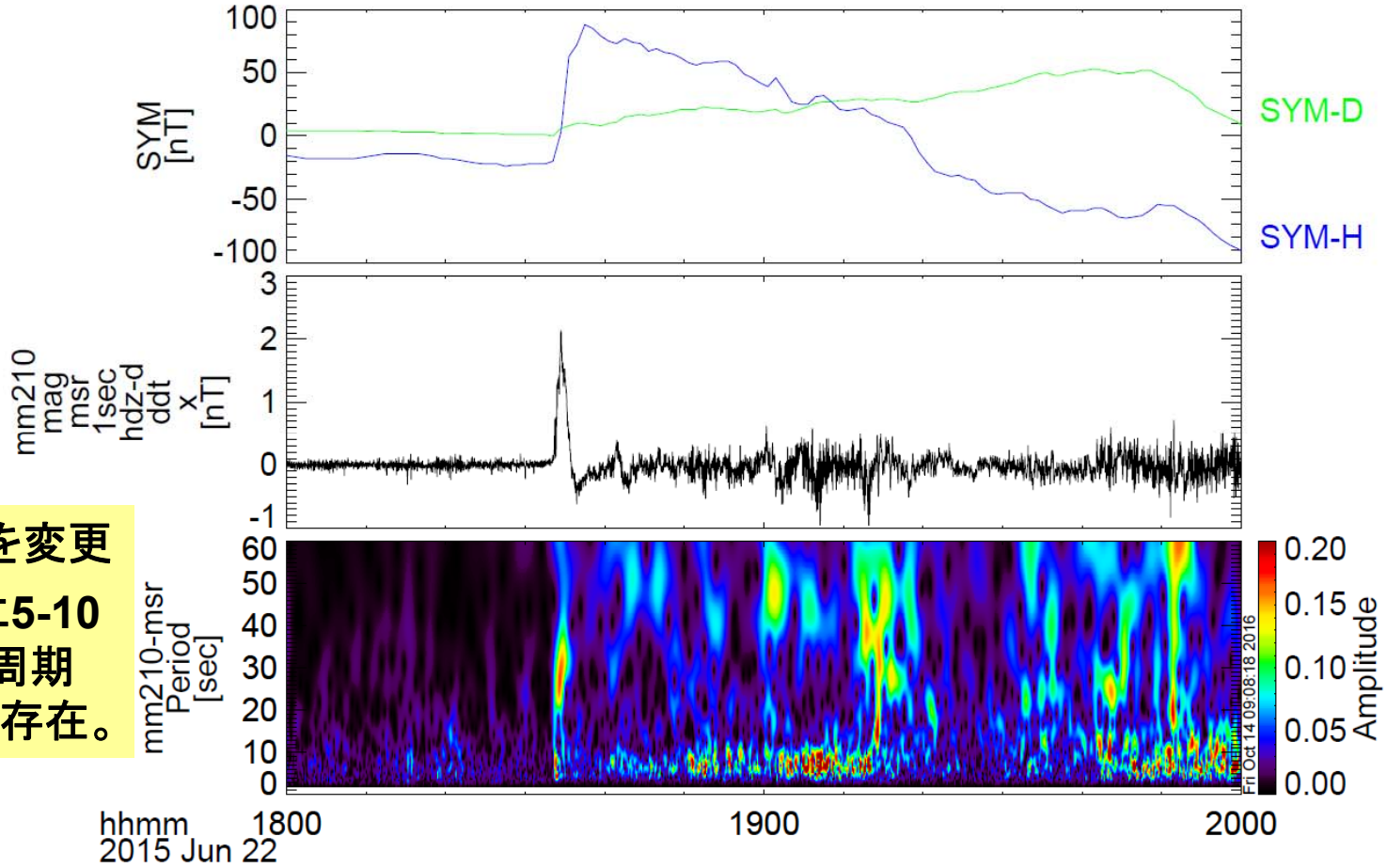


4. 周波数スペクトル解析

4.3 S(Stockwell)変換 ustrans_pwrspc

THEMIS> ylim, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_stpwrspc', 0, 60

THEMIS> tplot, ['wdc_mag_sym','mm210_mag_msr_1sec_hdz-d_ddt_x', ' mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_stpwrspc']



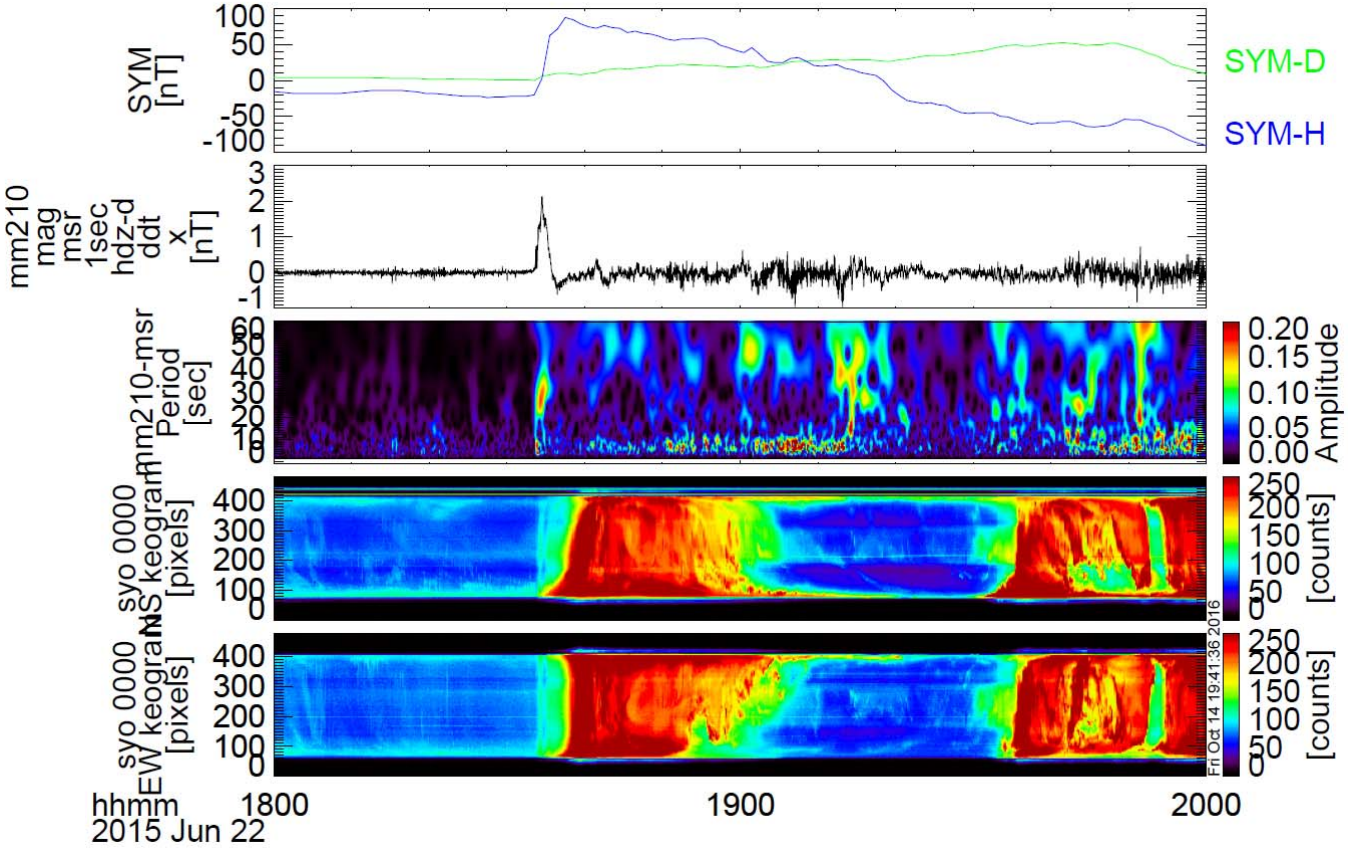
- **y軸のスケールを変更**
- **磁気嵐開始後に5-10秒と30-60秒の周期に卓越した波が存在。**

4. 周波数スペクトル解析

4.3 S(Stockwell)変換 ustrans_pwrspc

```
THEMIS> iug_load_ask_nipr, site='syo'
THEMIS>  tplot, ['wdc_mag_sym','mm210_mag_msr_1sec_hdz-d_ddt_x',
'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip_stpwrspc', 'nipr_ask_syo_0000_ns',
'nipr_ask_syo_0000_ew']
```

- 昭和基地のオーロラデータとの比較をしてみてもよい。
- S変換した茂尻のスペクトルと対応関係があるか？



6. まとめ

- tplot変数とはTDAS上の時系列データ参照の概念であり、IDLのメモリー上にその実体となるメタデータ付きデータ構造体がある。

- get_dataおよびstore_data によりIDLの通常の配列とのやり取りが可能。

- calc コマンドによりtplot変数の演算ができる。

- 各種フィルター処理やスペクトル解析を行うことができる。

- UDAS3.00.1以降のバージョンでは、IUGONETで独自に開発した描画や解析ツール(相互相関・無相関検定、コヒーレンス解析、トレンド検定)などが付け加わっている。

A. tplot変数への各種フィルター処理

A.1 tsmooth_in_time でスムージング

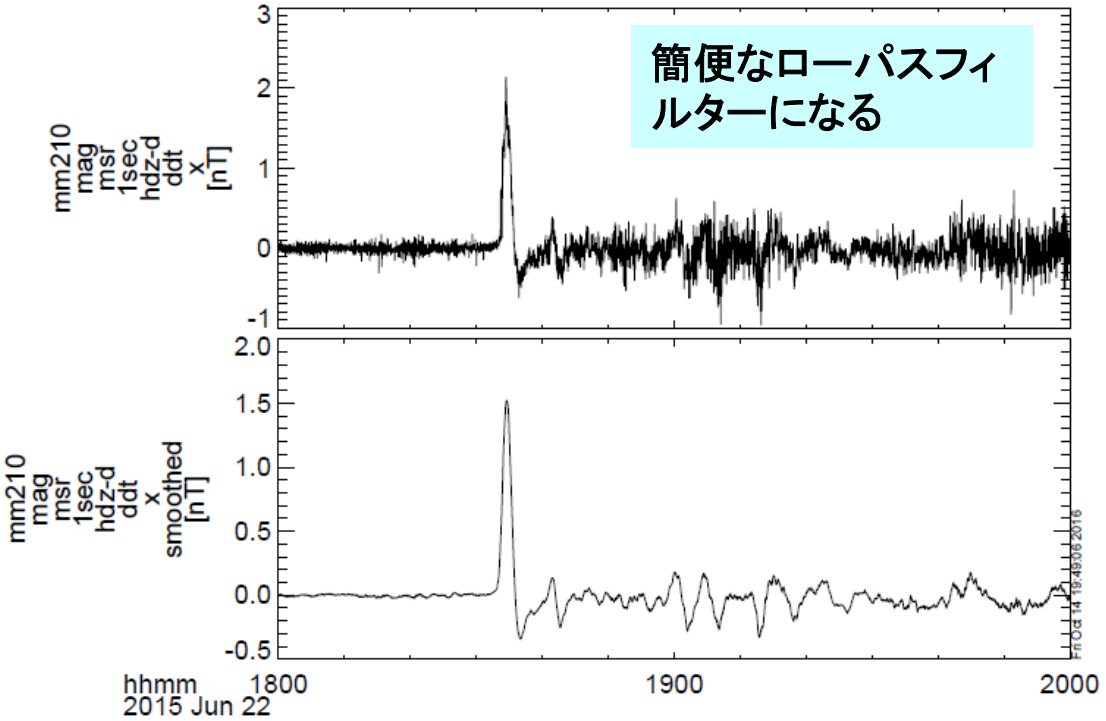
tsmooth_in_time, 'tplot変数名', 平均幅[秒]

(例) tsmooth_in_time, 'mm210_mag_msr_1sec_hdz-d_ddt_x', 60

```
THEMIS> tsmooth_in_time, 'mm210_mag_msr_1sec_hdz-d_ddt_x' , 60
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz-
d_ddt_x','mm210_mag_msr_1sec_hdz-d_ddt_x_smoothed']
```

- 指定された**時間幅で移動平均**することでスムージングされた結果が …**_smoothed** という名前の新しいtplot変数に格納される。

- **平均幅を秒数で与える**点に注意。上の例は60秒=1分幅で移動平均している。



A. tplot変数への各種フィルター処理

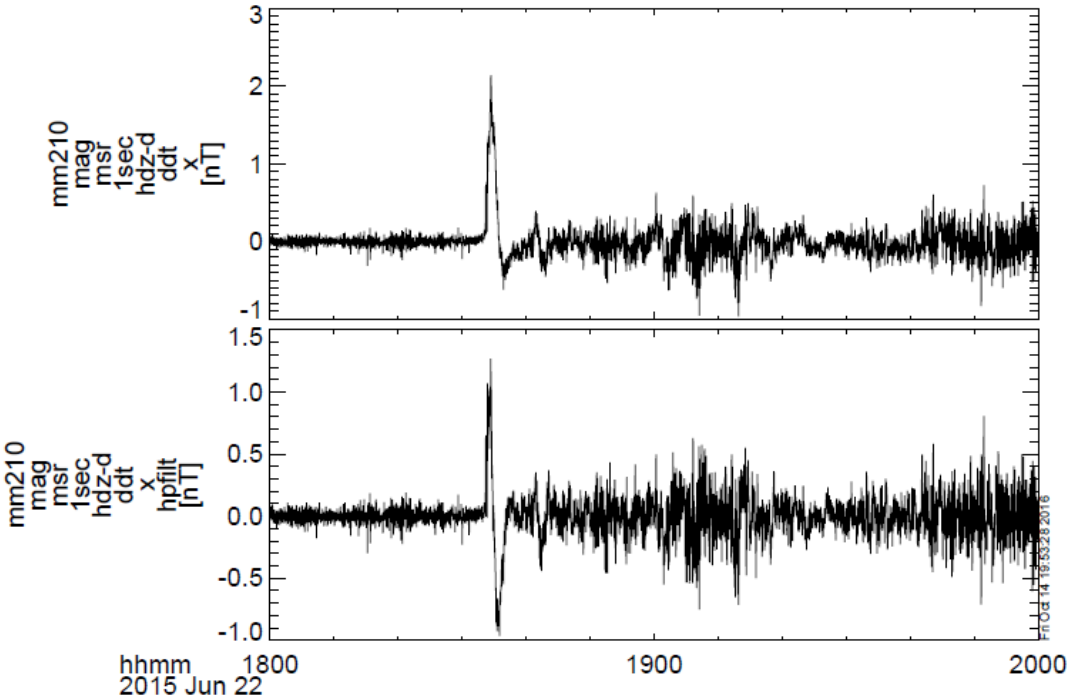
A.2 thigh_pass_filter でハイパス・フィルター

thigh_pass_filter, 'tplot変数名', 下限周期[秒]

(例) thigh_pass_filter, 'mm210_mag_msr_1sec_hdz-d_ddt_x', 60

```
THEMIS> thigh_pass_filter, 'mm210_mag_msr_1sec_hdz-d_ddt_x', 60
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz-d_ddt_x',
'mm210_mag_msr_1sec_hdz-d_ddt_x_hpfilt']
```

- **結果が …_hpfilt という名前の新しいtplot変数に格納される。**
- **ただしデジタルフィルターではなく、簡易的なもの。**
- **実際は前頁の tsmooth_in_time でローパスフィルターされたデータを元データから差し引いている。**



A. tplot変数への各種フィルター処理

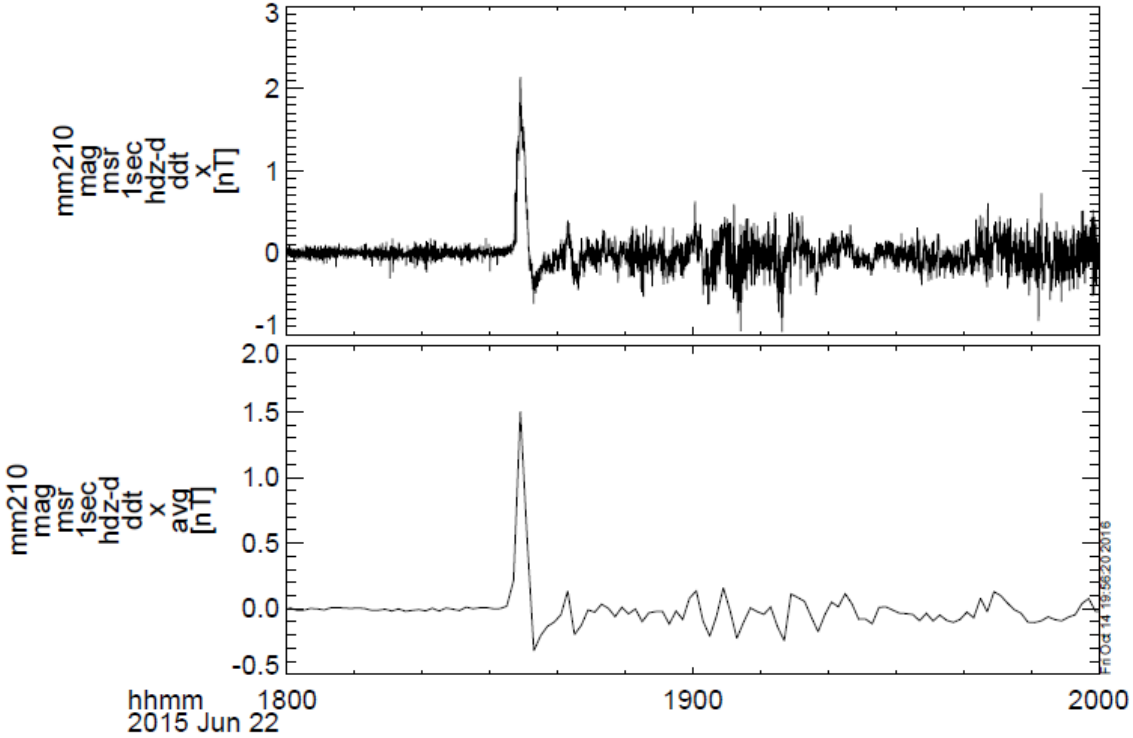
A.3 avg_dataで~分値、~時間値に平均

avg_data, 'tplot変数名', 平均時間幅[秒]

(例) avg_data, 'mm210_mag_msr_1sec_hdz-d_ddt_x', 60

```
THEMIS> avg_data, 'mm210_mag_msr_1sec_hdz-d_ddt_x' , 60
THEMIS> tplot, ['mm210_mag_msr_1sec_hdz-d_ddt_x',
'mm210_mag_msr_1sec_hdz-d_ddt_x_avg']
```

- 結果が …**_avg** という名前の新しいtplot変数に格納される。
- 第2引数に平均の時間幅を与える。3600[秒]にすれば1時間平均、60にすれば1分平均。
- 元データの時間分解能より小さい時間幅を与えると、結果が歯抜けデータになってしまうので注意。



B. tplot変数の相関解析など

B.1 tinterpolでデータ補間

tinterpol, 'tplot変数名1', 'tplot変数名2', out_var = 'tplot変数名3' ,/LINEAR

tplot変数2のデータに合わせてtplot変数1を補間する

(例) tinterpol, 'mm210_mag_msr_1sec_hdz-d_ddt_x', 'mm210_mag_msr_1sec_hdz-d_ddt_y', out_var = 'mm210_mag_msr_1sec_hdz-d_ddt_x_interp_linear' ,/linear

```
THEMIS> tinterpol, 'mm210_mag_msr_1sec_hdz-d_ddt_x', 'mm210_mag_msr_1sec_hdz-d_ddt_y', out_var = 'mm210_mag_msr_1sec_hdz-d_ddt_x_interp_linear' ,/linear
```

- out_varを何も新しいtplot変数名を指定しない場合、結果が …**_interp** という名前の新しいtplot変数に格納される。
- 補間する方法は3種類。

  線形補間(linear)、2次補間(quadratic)、スプライン補間(spline)
- 元データの時間分解能より小さい時間幅を与えると、結果が歯抜けデータになってしまうので注意。

B. tplot変数の相関解析

B.2 uspec_cohで2種のデータ間のコヒーレンス、位相を計算

uspec_coh, 'tplot変数名1', 'tplot変数名2'
(例) uspec_coh, 'mm210_mag_msr_1sec_hdz-d_ddt_x', 'mm210_mag_msr_1sec_hdz-d_ddt_z'

●指定時刻によるデータの切り出し

THEMIS> time_clip, 'mm210_mag_msr_1sec_hdz-d_ddt_x', '2015-06-22/19:00', '2015-06-22/20:00'

THEMIS> time_clip, 'mm210_mag_msr_1sec_hdz-d_ddt_z', '2015-06-22/19:00', '2015-06-22/20:00'

●コヒーレンスの計算

THEMIS> uspec_coh, 'mm210_mag_msr_1sec_hdz-d_ddt_x_tclip', 'mm210_mag_msr_1sec_hdz-d_ddt_z_tclip'

```
----------------Coherence analysis result--------------------------
coherence confidence interval = 0.153318
max coherence = 0.85513685
main_period = 41.8721
-------------------------------------------------------------------
```

uspec_cohを実行すると、コンソール上にコヒーレンスの最大値とその時の周期、コヒーレンスの統計的優位基準が出力される

B. tplot変数の相関解析

B.2 uspec_cohで2種のデータ間のコヒーレンス、位相を計算