

更新履歴

- 2017/06/03 (土) 石田
- 2017/06/04 (日) 石田
- 2017/06/12 (月) 石田
- 2017/06/13 (火) 石田

指示書作成

ターミナル上で `ssh -Y cs701kk@ut12x1` を実行し指示書作成用サーバに入る

Device Table

- A1~A4は偏波第1成分、A5~A16は偏波第2成分
- A5, A6は受信機トラブルのため今回は不使用

Receiver	Target line	Other lines (editing)
A1, A2, A15, A16	NH ₃ (1, 1)	-
A3, A7	NH ₃ (2, 2)	-
A4, A8	NH ₃ (3, 3)	-
A9	CCS(2-1)	-
A10	H81 β	-
A11	¹⁵ NH ₃ (1, 1)	-
A12	¹⁵ NH ₃ (2, 2)	-
A13	NH ₃ (4, 4)	-
A14	H ₂ O(6,1,6-5,2,3)	-

Files Source Scan Device Summary

Device Table: h29ut4 [Choose] [Load] [Samples] [Save] [Check]

Please use a sample device table and don't modify "Rx" and "SB" on Frontend panel, and "Frontend" on SAM45 panel

Frontend

	Rx	SB	Tracking Freq	IF
Rx1	H20ch1	USB	23.2	6
Rx2	H20ch2	USB	23.2	6
Rx3		USB		
Rx4		USB		
Rx5		USB		
Rx6		USB		
Rx7		USB		
Rx8		USB		

For TMULT
Initial rotation angle: 0.0 deg

SAM 45

Sub Array : Enable Disable

Resolution (kHz): 15.26 3.81

	Frontend	Att	Center Freq		Frontend	Att	Center Freq
A1	Rx1	5	23.6944955	A2	Rx1	5	23.6944955
A3	Rx1	5	23.7226333	A4	Rx1	5	23.8701292
A5	Rx2	5		A6	Rx2	5	
A7	Rx2	5	23.7226333	A8	Rx2	5	23.8701292
A9	Rx2	5	22.344033	A10	Rx2	5	23.86086
A11	Rx2	5	22.6249295	A12	Rx2	5	22.6498434
A13	Rx2	5	24.1394163	A14	Rx2	5	22.2350798
A15	Rx2	5	23.6944955	A16	Rx2	5	23.6944955

Options: Zeeman

SAM45 option: IPTIM (sec): 0.1

Generate & Send Print Quit

Project Info
ID: cs701kk Project: proj3
Obstable: Source:
Scan: Device: h29ut4

Load (Device): h29ut4.device loaded

観測方法

Terminal (右上)

1. H22_MCP.pyを実行しミラーを変更

Work Station COSMOS User-Interface

1. Group = cs701kk
2. Project = projX (Xはそれぞれに割り当てられた番号)
3. File name = (zobsで作った指示書)
4. (Tuning) 基本的に必要なし
5. Start Observation

SAM45 QUICK-LOOK

- all : 全受信機のスペクトルを表示
- refresh : スペクトルの再描画

解析手順

ターミナル上で `ssh -Y cs701kk@vc06x1` を実行し解析用サーバに入る

JNEWSTAR

NRO 45mで得られたスペクトルを解析するためのソフトウェア

1. 解析用サーバ内で `jnewstar` と入力
2. 自分のプロジェクト番号を入れて実行

Large INTEG

各受信機で観測された異なるスキャンを、観測されたポジションごとに足し合わせる

1. File Listをクリックし観測データを指定
2. spectral file (out)に出力ファイル名を指定。文字数制限があるので注意
3. arrayで解析したい輝線を観測した受信機を指定
4. integration weight = $1 / \text{rms}^2$
5. Exec => Start

Auto SP sum

各受信機のデータを足し合わせる

1. File ListをクリックしLarge INTEGしたファイル(~.INTEG)を選択
2. spectral file (out)に出力ファイル名を指定
3. Mode = XY coord
4. SP.ALLをクリック
5. Exec => Start

Show Spectra

スペクトルの様子を確認する

1. File ListをクリックしAuto SP sumしたファイル(~.ATSUM)を選択
2. SP.ALLをクリック
3. Exec => Start
4. スペクトルの表示画面で `Ctrl+N` でぱらぱら見ていき、ノイズレベルが高いようであれば、次のSMOOTHINGに進む

SMOOTHING

分光計のチャンネルをいくつかにまとめ、ノイズレベルを下げる

1. File ListをクリックしAuto SP sumしたファイル(~.ATSUM)を選択
2. spectral file (out)に出力ファイル名を指定

3. execution mode = batch
4. smoothing mode = binning-up
5. smoothing width = 2 or 4 or 8 (様子を見つつ)
6. Exec => Start

BASELINE

スペクトルのゼロレベルを揃える

1. File ListをクリックしAuto SP sumしたファイル(~.ATSUM)、またはSMOOTHINGしたファイル(~.SMTHG)を選択
2. spectral file (out)に出力ファイル名を指定
3. SP.ALLをクリック
4. Exec => Start
5. スペクトルの表示画面でSetRangesし、ゼロレベルを指定する範囲を選択
6. GOでフィット。うまくいかない場合はOrder & Waveで次数を2にする。もしくはfunctionでpolynomialの代わりに、sin & cosを指定する
7. 良さそうであればSave & Nextし、全スペクトルについて実行

Show Spectra

ベースラインがちゃんと引けているか確認する

PROFILE MAP

観測されたポジションごとのスペクトルマップを作る

1. File ListをクリックしBASELINEしたファイル(~.BASE)を選択
2. Coordinate = (X,Y)
3. POSITION MAPをクリックし、全領域を選択
4. Exec => Start
5. Image&PrintでPSファイルに出力し保存

Gaussian fit

輝線をガウシアンでフィットする

1. File ListをクリックしBASELINEしたファイル(~.BASE)を選択
2. SP.ALLをクリック
3. Exec => Start
4. スペクトルの表示画面でFitting Range => Set range(s)し、輝線の範囲を選択
5. GOでフィット。フィットされた結果は同じ画面に表示される
6. Image & PrintでPSファイルに出力し保存
7. 良さそうであればSave then Nextし、全スペクトルについて実行

MAP (large)

ピーク強度図や積分強度図を作成する

1. File ListをクリックしBASELINEしたファイル(~.BASE)を選択

2. map file (out)に出力ファイル名を指定
3. Coordinate = (RA,DEC)
4. POSITION MAPをクリックし、全領域を選択
5. Map grid Spacing = 10
6. Gridding mode = Gaussian smoothing
7. obs grid / FWHM of Gaussian = 80
8. output mode = integrated intensity or peak temperature
9. (integrated intensityの場合) Sp rangeで積分範囲を指定
10. Exec => Start

その他

FITSファイルにWCS (World Coordinate System)を貼る

以下のpythonスクリプトを作成し、ターミナルで`python filename.py`と打ち込む

```
import pyfits

hdu = pyfits.open('fitsname.fits')
data, header = hdu[0].data, hdu[0].header

header['CRVAL1'] = --- # 基準となるピクセルのRAをdegを単位として入力
header['CRVAL2'] = --- # 基準となるピクセルのDECをdegを単位として入力
header['CRPIX1'] = 3 # 基準となるピクセルがRA方向で何個目のピクセルかを入力
header['CRPIX2'] = 3 # 基準となるピクセルがDEC方向で何個目のピクセルかを入力
header['CDELT1'] = -0.01111 # RA方向のピクセルの間隔をdegを単位として入力(-を忘れずに!)
header['CDELT2'] = 0.01111 # DEC方向のピクセルの間隔をdegを単位として入力
header['CUNIT1'] = 'deg' # RAの単位がdegであることを指定
header['CUNIT2'] = 'deg' # DECの単位がdegであることを指定
header['CTYPE1'] = 'RA---SIN' # RAのprojectionを指定(おまじない)
header['CTYPE2'] = 'DEC--SIN' # DECのprojectionを指定(おまじない)

pyfits.writeto('output.fits', data, header)
hdu.close()
```

カレントディレクトリ内の全ファイル名の末尾に.psを付加する

以下のシェルスクリプトを直接ターミナル上に打ち込む

```
for name in *
do
mv $name $name.ps
done
```