Owens Valley Solar Array

OVSA

OVSA SOFTWARE NOTE No. 1

Version 1.1 Last updated 18 December 2000

OVSA SOFTWARE AND DATA

Peter T. Gallagher & Dale E. Gary
Big Bear Solar Observatory/Owens Valley Solar Array
New Jersey Institute of Technology
Big Bear City, CA 92314
U.S.A.

ptg@bbso.njit.edu
dgary@njit.edu

1 Introduction

This document is meant as a guide to users of calibrated data from the Owens Valley Solar Array (OVSA). There are currently two levels of data produced, including quicklook data which can be viewed at our web page and calibrated data in Flexible Image and Transport System (FITS) format. Here, we describe the calibrated FITS data and its associated software which allows the production of light curves, spectra and dynamic spectra. Background subtraction and spectral fitting features have also recently been added.

It should also be noted that prior to using this software, you must have installed IDL and the SolarSoftWare (SSW) package together with the OVSA branch of the tree. SSW can be downloaded from `http://www.lmsal.com/ssw/` while further OVSA specific information can be found at `http://www.ovsa.njit.edu/`.

OVSA instrument details can be found in Hurford et al. (1984) and Gary & Hurford (1998).

2 OVSA FITS

The OVSA data is written to FITS files as binary table extensions which are described in detail in Thompson (1994) and references therein. First the primary FITS header is created with the FITS file creation date inserted together with the `EXTEND` keyword set to true – this keyword indicates that a FITS extension will be written. Next, the binary table extension header is created with the following standard keywords included.

```
XTENSION= 'BINTABLE'           /Written by IDL:  Thu Mar 16 17:34:08 2000
BITPIX  =                    8 /
NAXIS   =                    2 /Binary table
NAXIS1  =              1295135 /Number of bytes per row
NAXIS2  =                    1 /Number of rows
PCOUNT  =                    0 /Random parameter count
GCOUNT  =                    1 /Group count
TFIELDS =                    5 /Number of columns
DATE    = '2000-03-16'         /
EXTNAME = 'OVSA Data'          /Binary table extension
FILENAME= 'ovsa_1-18GHz_sp_19991121_1805.fts'         /
ORIGIN  = 'NJIT    '           /Institute where file was written
TELESCOP= 'OVSA    '           /Owens Valley Solar Array
SEQ_NUM = 'G0281439_07263'     /Unique number for this dataset
DATE_OBS= '1999-11-21T18:05:05.000Z' /Start date/time of observation
DATE_END= '1999-11-21T18:29:55.000Z' /End date/time of observation
OBJ_ID  = 'AR8340  '           /Object ID
XCEN    =                   20 /Antenna pointing in arcsec from Sun centre
YCEN    =                   20 /Antenna pointing in arcsec from Sun centre
FREQMIN =              1.20000 /Min freq in observation (GHz)
FREQMAX =              18.0000 /Max freq in observation (GHz)
TFORM1  = '161250E '           /Real*4 (floating point)
TTYPE1  = 'CHANNEL,FREQ,TIME'  /Label for column 1
```

```
TDIM1   = '(25,43,150)'        /Array dimensions for column 1
TFORM2  = '43E     '           /Real*4 (floating point)
TTYPE2  = 'FREQ/GHZ'           /Label for column 2
TFORM3  = '63A     '           /Character string
TTYPE3  = 'FEED NUMBER AND POLARIZATION' /Label for column 3
TDIM3   = '(9,7)   '           /Array dimensions for column 3
TFORM4  = '162325E '           /Real*4 (floating point)
TTYPE4  = 'ERRORS  '           /Label for column 4
TDIM4   = '(25,43,151)'        /Array dimensions for column 4
TFORM5  = '150E    '           /Real*4 (floating point)
TTYPE5  = 'TIME/MSEC'          /Label for column 5
```

The binary table header and data units are written to the FITS file using `write_ovsa_fits`, which is executed using the following syntax:

```
IDL> write_ovsa_fits, avg, 'G0281439_07263',f, feedinfo, sigma, $
IDL> 'AR8340', t, avg, tls, xcen, ycen
```

This routine also writes an IAU standard file name which in this case is `ovsa_1-18GHz_sp_19991121_1805.fts`. The two characters `sp` indicate that the file contains spectral data.

3 Analysis Software

This section details the OVSO object class and the methods associated with it. Specifically, we describe how to create an OVSA object together with the methods for plotting, fitting and saving dynamic spectra, light curves and spectra.

3.1 OVSA Class

An instance of the OVSA class is created using the `obj_new` function. This can be achieved using;

```
IDL> obj = obj_new( 'ovsa' )
```

The methods associated with the object together with the various arguments and keywords of each method can be accessed by typing:

```
IDL> obj -> help
  0) PRO ovsa::help
  1) PRO ovsa::read,filename,data
  2) PRO ovsa::plot,ant,gen,tmin,tmax,dynamic=dynamic,light curve=light curve,
     amplitude=amplitude,phase=phase,spectrum=spectrum,interactive=interactive,
     print=print,save=save
```

An OVSA FITS file can then be read using the `read` method associated with the object via

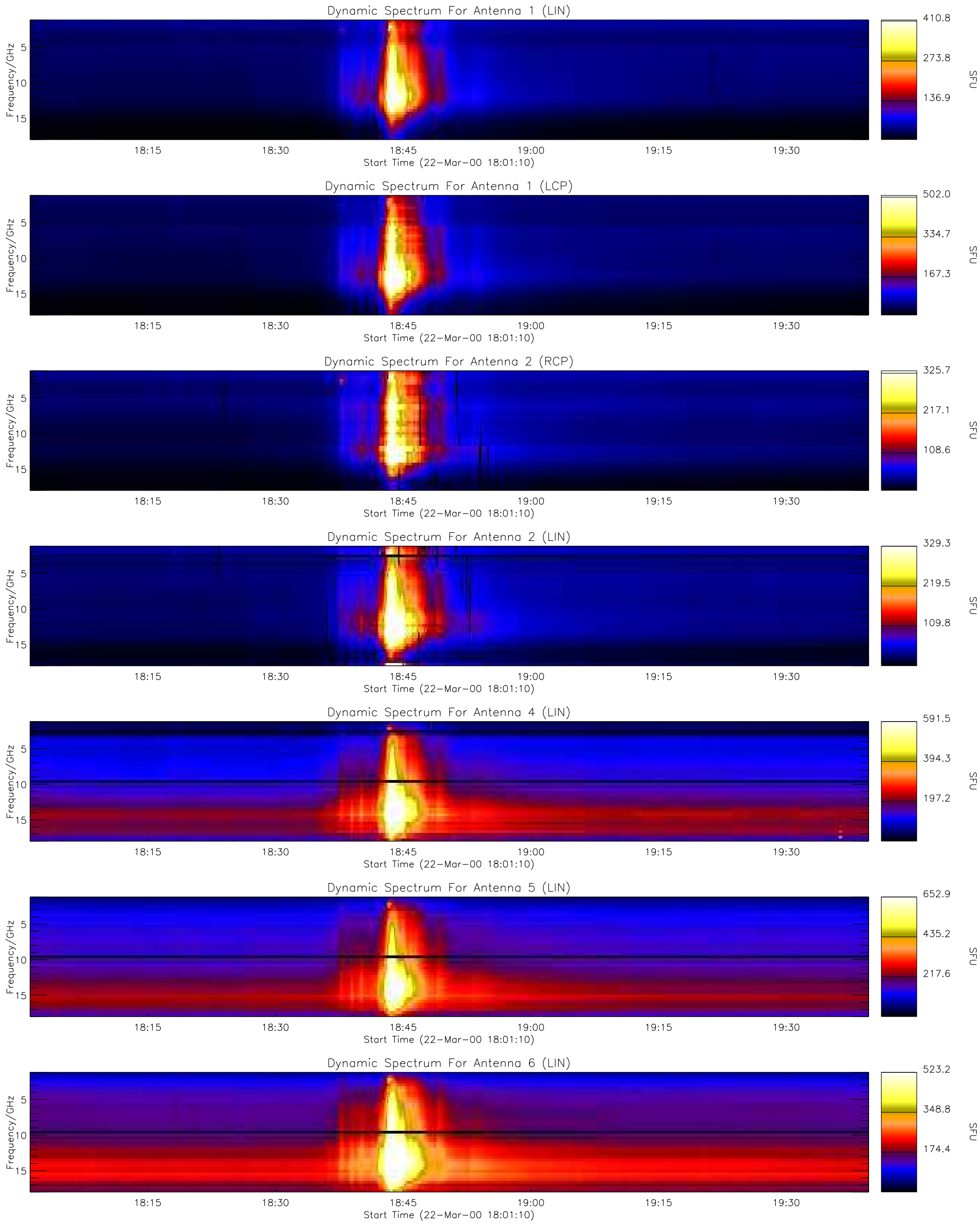

Figure 1: OVSA summary plot showing a large microwave burst as observed with each available antenna and frequency.

```
IDL> obj -> read, 'filename.fits'
```

This method also prints the `filename.fits` binary extension header which contains information on pointing, start and end times, observed frequencies, campaign IDs and a variety of other information – see Section 2 for further details. If the user is interested in accessing the data directly, a second

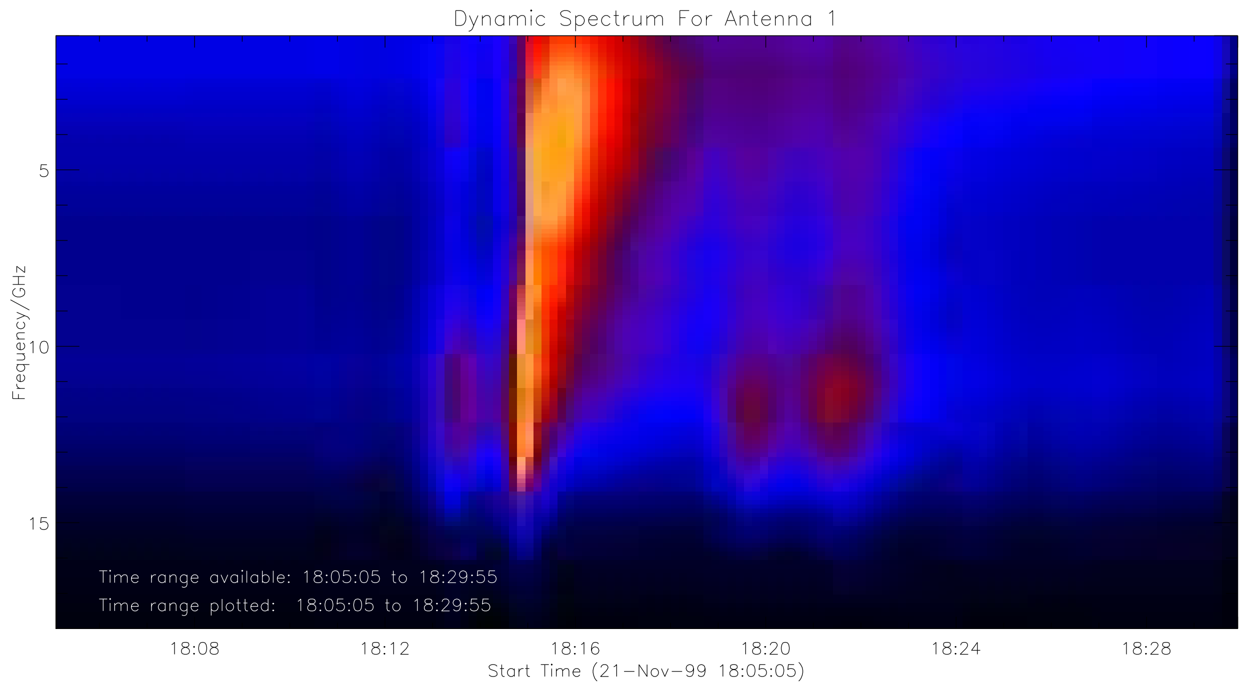

Figure 2: A left-circular polarization dynamic spectrum for antenna 1 at 43 frequencies between 1 and 18 GHz.

argument can be included when calling the `read` method. This second argument then contains the header and data arrays incorporated within a standard IDL structure, e.g.

```
IDL> obj -> read, 'filename.fits', idl_data_structure
```

The data can then be displayed using the `plot` method which allows us to produce dynamic spectra, light curves and spectra. There are a number of keywords associated with the `plot` method, not all of which are described in the following sections.

3.2 Getting a General Overview

Before analysing the data, it is useful to view the data obtained from each antenna and in each polarization. This can be achieved using the `/Summary` keword with the `plot` method.

```
IDL> obj -> plot, /Summary
```

Figure 1 gives an example of the typical output from the above.

3.3 Viewing Dynamic Spectra

A dynamic spectrum, showing the total power from a chosen antenna as a function of frequency and time, is create as follows;

Antenna Number	Polarization
1	LIN or LCP
2	LIN or RCP
3	Not in use
4	LIN
5	LIN
6	LIN
7	LIN

Table 1: OVSA antenna numbers and polarization states measured.

```
IDL> AntennaNumber = 1
IDL> obj -> plot, AntennaNumber, /LCP, /Dynamic
```

where the the variable, `AntennaNumber`, can take the values 1, 2, 4, 5, 6, or 7 – antennas number 1 and 2 correspond to the two 27–m dishes, while 4, 5, 6 and 7 refer to the four 2–m dishes. The polarization must also be specified via the `/LIN`, `/LCP` or `/RCP` keywords. These are given in Table 1. An example of a dynamic spectrum created with the `plot` method is given in Figure 1.

A re-gridded dynamic spectrum can be viewed via:

```
IDL> obj -> plot, AntennaNumber, /LCP, /Dynamic, /Mesh
```

The `Mesh` keyword converts the data to a regular grid and fills in missing data values. As with all interpolation procedures, this should be used cautiously.

A background subtracted dynamic spectrum can be produced using the `Background` keyword.

```
IDL> obj -> plot, AntennaNumber, /LCP, /Dynamic, /Background
```

The user can use the cursor to identify a pre-flare background region, from which an average spectrum is created. This average spectrum is then subtracted from the dynamic spectrum. The `Background` keyword can be used with the `Dynamic`, `Spectrum` and `Lightcurve` keywords.

It is also possible to view the dynamic spectrum between any two times by adding a second and third argument to the `plot` method. For example,

```
IDL> obj -> plot, AntennaNumber, '18:12', '18:24', /LCP, /Dynamic
```

will plot the dynamic spectrum for antenna 1 between 18:12 U.T. and 18:24 U.T.

3.4 Creating Light Curves

A light curve at a particular frequency is again created using the `plot` method, but in this case the `Lightcurve` rather than the `Dynamic` keyword is used. The left panel of Figure 2 gives an example of a light curve produced using:

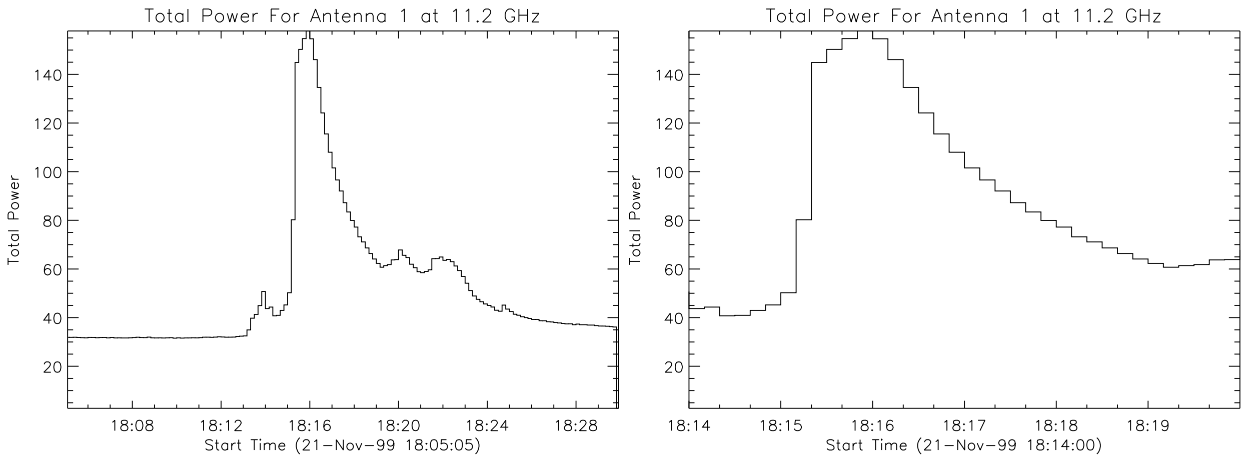

Figure 3: The left panel shows a left-circular polarization OVSA 11.2 GHz total power light curve between 18:05 U.T. and 18:29 U.T for antenna 1, while the right panel shows a section of the same data extracted between 18:14 U.T. and 18:20 U.T.

```
IDL> AntennaNumber = 1
IDL> Frequency = 11.0
IDL> obj -> plot, AntennaNumber, Frequency, /LCP, /Lightcurve
```

where the variable `Frequency` can take any value between 1.2 and 18.0 GHz. As OVSA records 43 frequencies in the 1.2 to 18 GHz range, the `plot` method will display the observed frequency closest to the requested value. In this example the closest observed frequency to the requested frequency of 11.0 GHz was 11.2 GHz which can be seen in Figure 2.

Similarly to the option available when producing a dynamic spectrum plot, we can also change the start and end times used when plotting a light curve. In this case we use,

```
IDL> obj -> plot, AntennaNumber, Frequency, '18:14', '18:20', /LCP, /Lightcurve
```

to plot the 11.2 GHz light curve for antenna 1 between the times 18:14 U.T. and 18:20 U.T.

3.5 Viewing and Fitting Spectra

A radio spectrum can be extracted using the `plot` method together with the `Spectrum` keyword. If for example, we are interested in the spectrum at 18:14 U.T. we can type;

```
IDL> obj -> plot, AntennaNumber, '18:14', /LCP, /Spectrum
```

which is shown in Figure 3.

The spectrum can be fitted using a function of the form

$$S = A\nu^{\alpha}(1 - \exp(-B\nu^{-\beta})) \qquad (1)$$

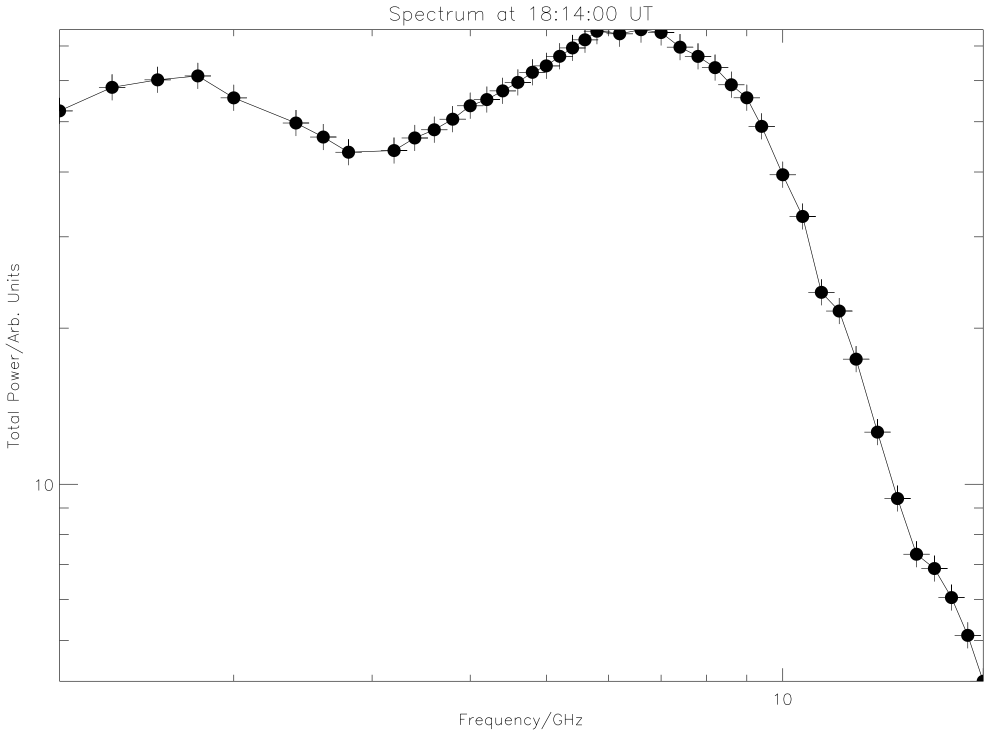

Figure 4: Radio spectrum from 1.2 GHz to 18.0 GHz extracted at 18:14 U.T.

which expresses the frequency dependence of the flux density (Stähli et al., 1989). The parameters α and $\alpha - \beta$ represent the low- and high-frequency slopes respectively. This equation is derived by combining the radiative transfer equation for an isolated source with constant effective temperature

$$T_b = T_{eff}(1 - \exp(-\tau_n u)) \tag{2}$$

with $S = \Delta\Omega k T_b n u^2/c^2$ (Dulk & Marsh, 1982).

In general the microwave spectrum shows multiple peaks so the user must select a section of the spectrum using the cursor. When this selection has been made, the `plot` method in conjunction with `Fit` keyword calls the `mpfitexpr` procedure which performs a Levenberg-Marquardt least-squares fit i.e.,

```
IDL> obj -> plot, AntennaNumber, '18:14', /LCP, /Spectrum, /Fit
```

3.6 Saving And Restoring Your Plots

In order to save the data plotted in a particular dynamic spectrum, light curve or spectrum, the `Save` keyword is used with the `plot` method. This feature is particularly useful when comparing OVSA data to data from other instruments. For example, we can save the time and total power arrays together with the start and end times of the following

```
IDL> obj -> plot, AntennaNumber, Frequency, '18:14', '18:20', /LCP, /Lightcurve, /Save
```

by using the `Save` keyword as shown. The variables and arrays used in the plotting are written to `ovsa.sav` within an IDL structure named `ovsa_data`.

We can then restore, view and plot the these data outside the OVRO software using standard IDL and SSW routines. For example;

```
IDL> restore, 'ovsa.sav'
IDL> help, ovsa_data, /Structure
** Structure <8297fe4>, 5 tags, length=1224, refs=1:
   T_SEC           FLOAT     Array[150]
   POWER           FLOAT     Array[150]
   T_START         STRING    '1999-11-21T18:14:00'
   T_END           STRING    '1999-11-21T18:20:00'
   FREQ            STRING    '16.4'
IDL> utplot, ovsa_data.t_sec, ovsa_data.power, ovsa_data.t_base,$
IDL> timerange = [ ovsa_data.t_start, ovsa_data.t_end ]
IDL> oplot, other_instrument_time, other_instrument_flux
```

where the `other_instrument` arrays are time and flux arrays recorded by another instrument such as SXT or HESSI.

3.7 Interactive Mode

The interactive mode allows the user to point and click on the dynamic spectrum and view spectra or light curves at a variety of times or frequencies and can be run using the `Interactive` keyword combined with either the `Lightcurve` or `Spectrum` keywords.

For example, light curves for antenna 1 can be selected using the cursor by entering;

```
IDL> obj -> plot, 1, /LCP, /Lightcurve, /Interactive
```

while spectra can be extracted using the following:

```
IDL> obj -> plot, 1, /LCP, /Spectrum, /Interactive
```

3.8 Movie Mode

The movie mode can provide an interesting insight into the temporal evolution of a flare spectrum using the `movie` method. A movie from antenna 1 in linear polarization can be viewed by entering the following;

```
IDL> obj -> plot, AntennaNumber, /LIN, /Dynamic, /Movie
```

4 References

Dulk, G. A. & Marsh, K. A., 1982, *ApJ*, **259**, 350.
Stähli, M., Gary, D. E. & Hurford, G. J., 1989, *Sol. Phys.*, **120**, 351.
Gary, D. E. & Hurford, G. J., 1998, in *Proceedings of the Nobeyama Symposium*, Eds. T. S. Bastian, N. Gopalswamy and K. Shibasaki, NRO Report No. **479**, 429.
Hurford, G. J., Read, R. B., Zirin, H., 1984, *Sol. Phys.*, 94, 413.
Thompson, W., 1994, `http://orpheus.nascom.nasa.gov/cds/software_notes.html`, CDS Software Note No. 4.