CENTRE NATIONAL DE
LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE
VERSAILLES SAINT QUENTIN

INSTITUT PIERRE-SIMON LAPLACE

**CENTRE D'ETUDE DES ENVIRONNEMENTS
TERRESTRE et PLANETAIRES**

CLUSTER DATA PROCESSING

Transformation of STAFF waveform into a Magnetic Field Aligned coordinate system

Par

Patrick ROBERT & Claude DEVILLEDARY

Version 1.6 - Octobre 2000

Rapport Interne n° RI-CETP/6/2000

Contents

CLUSTER DATA PROCESSING

Transformation of STAFF waveform into a Magnetic Field Aligned coordinate system

P. Robert & C. Devilledary, CETP, Mars 2000
Revised October 2000

1 GENERAL REMARKS

This text is a complement of the document "Requirements Specification WEC/ISDAT Coordinate Transformation", written by the WECdata Coordinate Systems Working Group, ref. [1]. It is a summary of the different process and precaution to take an entire ULF waveform and to transform it into a magnetic field Aligned coordinate system. In particular, main problem are due to the duration of the waveform, and not to simply compute a point to point matrix. A practical method is given, with the detailed matrix and input, and with a temporary method to re-sample the magnetometer data.

2 DEFINITION OF THE MAGNETIC FIELD ALIGNED SYSTEM

The magnetic CLUSTER ULF waveform is delivered in the SR2 coordinate system (Spin Reference #2 system), also called DS system (Despun Satellite), i.e. a system defined as Z along the spin axis, and X in the plane defined by the spin axis and the direction of the Sun.

For scientific use, it is convenient to have these data in a magnetic coordinate system, where an axis is aligned with the DC magnetic field. To use similar convention with the Despun system, we define here a Magnetic Field Aligned system where Z is along the DC magnetic field, and X in the plane containing the DC magnetic field vector and the direction of the Sun. This system will be noted MFA hereafter.

The transformation is not so obvious, because what we call usually "DC magnetic field" (DC-MF hereafter) is not well defined, and its definition can vary according to the scientific use. In particular, for the ULF study, the components of the DC-MF must vary with time-scale long compared to the period of the ULF waves, but this time-scale can be shorter if we compare it to other time-scale, such as interval between magnetic field discontinuities.

A practical approach could be to define a frequency limit f_{MAX} for the maximum frequency of the DC-MF data, this limit having to be less than the lowest frequency of the ULF band, in such a way that there is no common frequency range between ULF and DC-MF data. The f_{MAX} frequency can be set according to the scientific purpose. This last point is important for the physical meaning of the transformation, and in particular, during the re-sampling operation of the law rate DC magnetic waveform, we must take care that this operation do not create high frequency component, which could perturb the initial ULF data.

3 REQUIRED INPUT DATA SET

To execute the transformation, we assume that the following data and parameters are available:

a) ULF calibrated waveform (nT) in SR2 coordinate system, extracted from STAFF/SC Data (fe=25 or 450 Hz).
b) FGM calibrated waveform (nT) in GSE coordinate system, extracted from PPD data set (fe=1/4 Hz), which will always be accessible. The transformation may also be done with FGM data defined in SR2 and sampled at 1 Hz, this will not change the general procedure.
c) Spin axis direction in GSE, extracted from auxiliary SPD data (every mn).
d) Starting time of first point of ULF and FGM period, and time resolution.

Note that the spin frequency, which has been requested to compute the ULF waveform in SR2 system, is available from SATT file with an accuracy of 6 digits. This accuracy of 1.e-6 allow processing of long ULF period, about 20 minutes, but no very much longer because 20 minutes of duration leads to an error of about 0.43 ° on the spin phase.

4 PRINCIPLE OF THE TRANSFORMATION

Before defining the matrix to be used, we summarise here the different steps and intermediate computations requested for doing the transformation.

Firstly, since we have the ULF data in SR2 coordinates and the FGM data in GSE coordinates, we have to define the transformation of GSE system to SR2 system. This requires the Spin axis direction in GSE, and the corresponding matrix is given in section 5.1 (see detailed computation in appendix 1)..

Secondly, to transform each point of the ULF waveform, we need to define a value of the DC-MF at each time of each point. This requires to re-sample the FGM waveform, at the same frequency than the ULF waveform (each new point of re-sampled FGM data having to be computed at the same time that the corresponding point of the ULF waveform). The FGM re-sampled data is called MAG data hereafter, and the re-sampling method is defined in section 5.2

Thirdly, we must have a ULF waveform which does not contain frequency in the range of the DC-MF frequency, and we have to apply a filter to the ULF waveform, since at least 2 components of this one (the spinning components) can contain DC-MF components. This process is given in section 5.3.

At last, having ULF and MAG data with the correct corresponding frequency range, the transformation of each point of the ULF waveform from the SR2 system to the MFA system, with the time corresponding MAG data can be done. Note that the Spin axis direction in GSE vary slowly, and can be assumed to be kept constant during the duration of the waveform. In other case, a linear interpolation can be used. The transformation matrix is given in section 6, and detailed calculation is given in appendix 2.

5 INTERMEDIATE COMPUTATIONS

5.1 computation of FGM waveform in SR2 system

Since ULF waveform is given in SR2 system, and FGM data is not always available in this system, we take the FGM data in the PPD data set where they are given in GSE system, with a time resolution of 4.0 seconds. So, the first thing to do is to transform this FGM data into the SR2 coordinate system.

The following matrix can be used to transform GSE system to SR2 system:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2} = \begin{pmatrix} a(Ry^2 + Rz^2) & -a(RxRy) & -aRxRz \\ 0 & aRz & -aRy \\ Rx & Ry & Rz \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{GSE} \quad \text{with} \quad a = \frac{1}{\sqrt{Ry^2 + Rz^2}}$$

where ***R*** vector is the spin axis (or Rotation axis) expressed in GSE coordinate system, and extracted from auxiliary SPD data. This matrix must be applied on each point of the FGM waveform.

The calculation of this matrix is given in details in appendix 1. One can note that we do not need any external library, such as Rocotlib [2], since the spin axis is available directly in GSE system.

5.2 Re-sampling and filtering FGM waveform

Sample rate of ULF data is 25.0 or 450.0 Hz, depending on the bit rate. MAG waveform in SR2 system, issued of previous processing from FGM data, has a sample rate of 4.0 seconds. The re-sampled method must <u>not</u> create artificial high frequency above the limit defined by initial Nyquist frequency, i.e.

$$f_{MAX} = 1/(2*4) = 0.125 \text{ Hz.}$$

Many re-sampling method can be envisaged. The simpler, a linear interpolation, cannot be used alone, because the jagged shape introduce high frequency. An appropriate, and not too complex method, must be defined. Note that the problem of re-sampling is studied in the ISSI book (C.C. Harvey and S.J. Schwartz, page 43).

Note also that for the high frequency band (up to 180 Hz), or for short duration of ULF waveform, we can use a mean constant value of MAG over the whole duration of the waveform.

For a first use, to wait until somebody gives a better algorithm, we propose a linear interpolation followed by a low pass filter, with the cut off frequency f_{MAX} defined above. Frequency f_{MAX} must be chosen less than the minimum frequency of the ULF range, of the order of 0.1 Hz.

5.3 Removing DC components from ULF data

Since the spacecraft is rotating at a spin rate of around 4 seconds, the local DC magnetic field introduces a strong component at fs= 0.25 Hz which is included into the ULF range. This means that the ULF waveform in the spinning coordinate system shows a very large sine curve, supporting the ULF data. After transformation into a “fixed” coordinate system, such as SR2, the two perpendicular

X and Y components, obtained from the spinning components, show a large mean value, which moreover can be compared to the DC magnetic field values given by the FGM waveform.

Before transforming the ULF waveform into the MFA coordinate system, it is essential that this part of DC component from the ULF waveform be removed.

The suggested method, as a first approach, consists to simply subtract the mean value. A more sophisticalled approach consists in filtering the ULF data, to eliminate the frequency below the preceding f_{MAX} value.

6 TRANSFORMATION OF ULF DATA INTO MFA SYSTEM

We now have both the ULF and the MAG data into the same SR2 coordinate system, with the same time resolution, and we took care that the two corresponding frequency bands have no common part. We can now simply apply on each point of the waveform the following transformation matrix, to get the ULF waveform into the Magnetic Field Aligned system defined in section 2:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MFL} = \begin{pmatrix} Ax & Bx & Cx \\ Ay & By & Cy \\ Az & Bz & Cz \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

Each term is a time dependant function of the direction of the DC magnetic field in SR2 system, given by the re-sampled MAG waveform, which is slowly varying with time. The complete calculation of this matrix is given in appendix 2. It is the product of 3 rotations, as :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MFL} = \begin{pmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} cos\varphi & sin\varphi & 0 \\ -sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

Angles θ, φ and ψ are defined as following.

θ : polar angle of **MAG** vector in SR2 system, i.e. :

$$sin\theta = \frac{\sqrt{Bx^2 + By^2}}{Bo}, \quad cos\theta = \frac{Bz}{Bo}$$

φ : azimuthal angle of **MAG** vector in SR2 system, i.e. :

$$sin\varphi = \frac{By}{\sqrt{Bx^2 + By^2}}, \quad cos\varphi = \frac{Bx}{\sqrt{Bx^2 + By^2}}$$

ψ : azimuthal angle of **Sun** vector in MAG system (see appendix 1), i.e. :

$$sin\psi = \frac{Sy_M}{\sqrt{{Sx_M}^2 + {Sy_M}^2}}, \quad cos\psi = \frac{Sx_M}{\sqrt{{Sx_M}^2 + {Sy_M}^2}}$$

where Sx_M, Sy_M, Sz_M are the components of **Sun** vector in MAG system (see appendix 1).

These components are given by:

$$\begin{pmatrix} Sx_M \\ Sy_M \\ Sz_M \end{pmatrix}_{MAG} = \begin{pmatrix} \sqrt{Ry^2+Rz^2}\ \cos\theta\ \cos\varphi - Rx\ \sin\theta \\ -\sqrt{Ry^2+Rz^2}\ \sin\varphi \\ \sqrt{Ry^2+Rz^2}\ \sin\theta\ \cos\varphi - Rx\ \cos\theta \end{pmatrix}$$

R vector being as before the spin axis (or Rotation axis) expressed in GSE coordinate system, and extracted from auxiliary SPD data.

One can note that, as in section 4.1.1, we do not need any external library, since the spin axis is available directly in GSE system.

A FORTRAN code, used to check the above matrix, is provided in appendix 3.

7 CHECK OF THE TRANSFORMATION ON GEOS-2 DATA

7.1 Principle

In order to check the validity of the preceding method and transformations, we have coded the method in a FORTRAN program, to process the filtering of the magnetometer and ULF data, and to do the transformation of coordinate system. This allow us to study the role of the frequency limits f_{MAX} for the maximum frequency of the DC-MF data, and the frequency limits f_{MIN} for the ULF frequency range. It provides moreover a computer code, which can be immediately used, for CLUSTER, or for any other data of the same type.

The data used to check the method are the GEOS-2 data, where both the magnetometer data and the ULF data are available in a frequency band very close to the one on CLUSTER. The GEOS magnetometer data are given with a time period of 5.504 s., to be compared to the 4s period for CLUSTER, and the ULF frequency data are sampled at 23.2556 Hz, to be compared to the 25 Hz of CLUSTER. So, a check of the program applied on the GEOS data, now well known 20 years after, allows us to have a good reliability for the same program which will be applied to the CLUSTER data, and, generally, to any else similar data.

7.2 Method used and computer code

The main steps of the computer code are the following:

1) reading GEOS MAG and ULF data in SR2 system, from pre-processed GEOS data set
2) re-sampling MAG data
3) removing the DC part of the ULF components
4) transform ULF data into MFA system
5) compute ULF spectra in SR2 and MFA to check differences
6) create output files of each step for visualisation

Code is written in FORTRAN 77 available on any computer, and the subroutines doing the transformations does not need any external library. These subroutines can be easily rewritten in C or IDL language. The main subroutines are given in appendix 3. The total program requires the specific routines to read the GEOS data and need a FFT routine to compute the spectra. The total code is available on request.

7.3 Results

Preceding output files are read by a PV-WAVE procedure to visualise the results, which are shown on Figures 1 to 8. Comments above these figures are given below.

Figure 1: example 1 of re-sampled magnetometer data over a short time period

The initial GEOS-2 MAG data are given in GSE system (the top pannel), which are converted into SR2 system (middle pannel) to be used as CLUSTER simulated data. The bottom pannel show the MAG re-sampled and filtered data, with here a f_{MAX} frequency equal to 0.09 Hz. Note that the angular discontinuities disappear, an the new DC-field shows yet slow variations.

Figure 2: example 1 of transformed ULF waveform into the MFA system

The initial GEOS-2 ULF data are shown on the top pannel, where the round symbol visualise the preceding MAG data (not filtered). The middle pannel show the filtered ULF data, with a f_{MIN} frequency of 0.1 Hz. Note that the offset disappear, and only the true ULF components remains visible. The bottom pannel show the result, i.e. the true ULF data in MFA coordinate system. Unfortunately, this case does not correspond to a polarised wave (see the following example), but it has been choosed to show the very good agreement between the DC field measured by the magnetometer experiment, and the 2 spinning components of the ULF sarch coil magnetometer experiment (top pannel). This agreement confirm the absolute necessity to filter both the MAG and ULF data to isolate the respectives components of the DC and alternative magnetic field.

Figure 3: example 2 of re-sampled magnetometer data over a short time period.

Same as figure 1, but for another example.

Figure 4: example 2 of transformed ULF waveform into the MFA system

Same as figure 2, but the choosed example is a transversal polarized wave of period around 1.67 s. (0.6 Hz). One can note that after transformation into MFA system, this component has completely disappeared on the Z component, which is aligned with the DC magnetic field. This result was already found 20 years ago during the studies of the GEOS data with another software, so we confirm the validity of the present software.

Figure 5: corresponding spectra in SR2 system

Note that before transformation the levels of the pic at 0.6 Hz for Bx, By,Bz (top to bottom) are respectively [-1., -0.7, -1.2] in logarithmic scale, which correspond in nT^2/Hz to:

$$[Bx, By, Bz] = [0.1, 0.2, 0.06]$$

Figure 6: corresponding spectra in MFA system

The levels of the pic at 0.6 Hz are now [-0.6, -0.6, -1.8], which correspond in nT^2/Hz to:

$$[Bx, By, Bz] = [0.25, 0.25, 0.016]$$

Thus the Z component has been strongly reduced by the transformation in the MFA system, corresponding to a transversal polarisation of the wave.

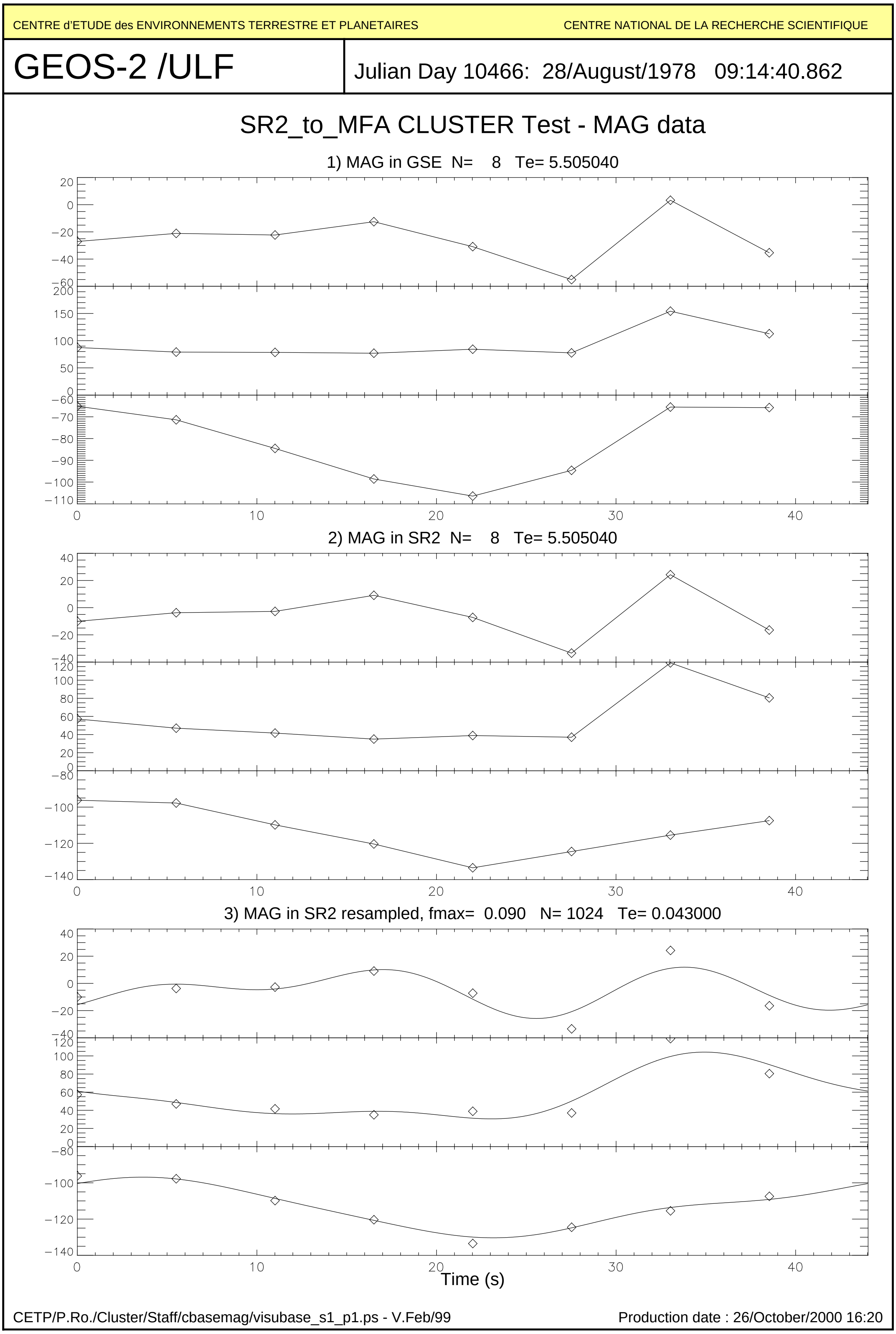
Figure 7: example 3 of re-sampled magnetometer data but over a long time period.

This example show how the re-sampling method carry out over a long time period (here 8192 points, i.e. 5.6 mn.). As previous example, the angular discontinuities disappear, an the new DC-field show slow variations, this is particularly visible at the beginning of the signal.

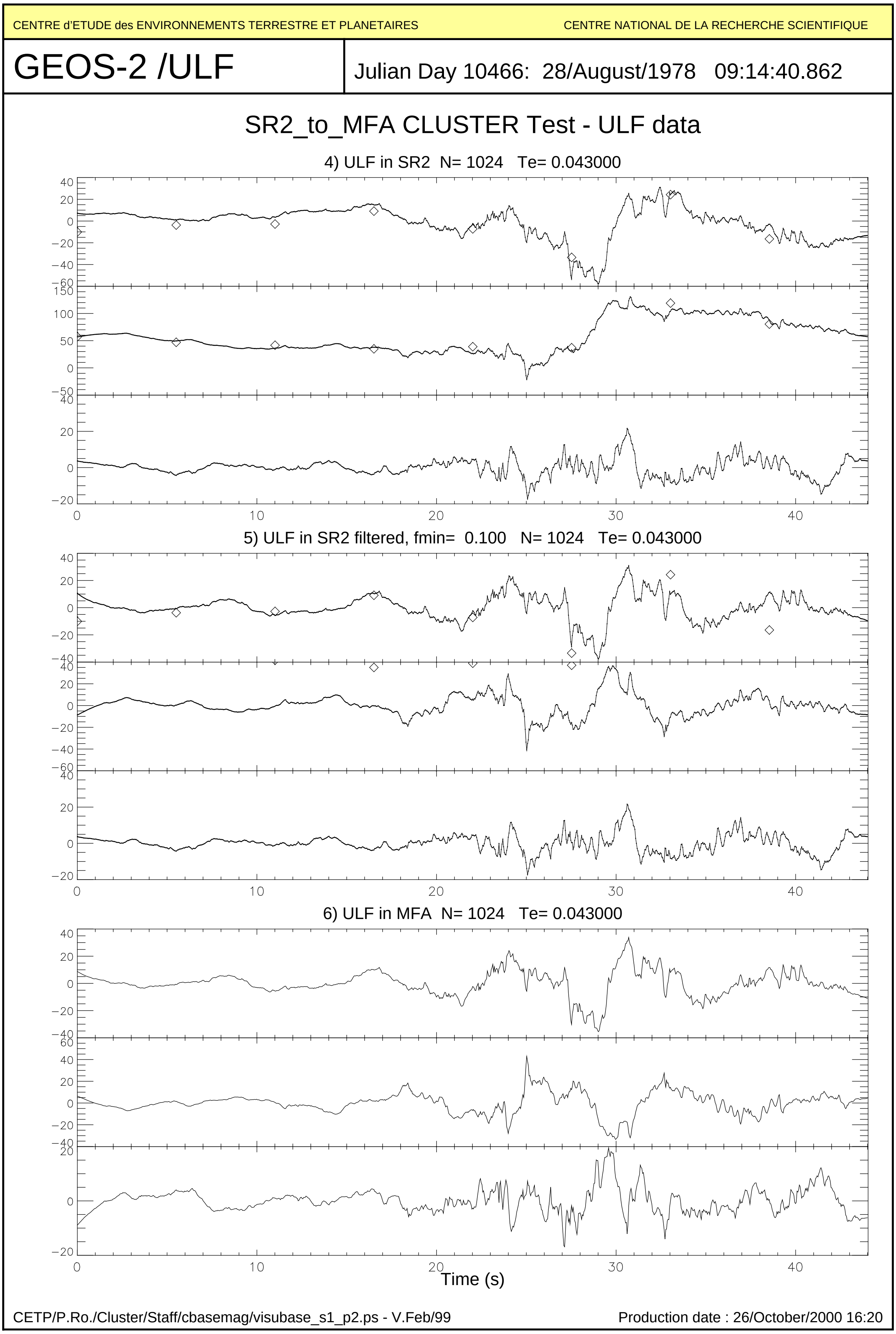
Figure 8: example 3 of transformed ULF waveform into the MFA system

Note as previous the good agreement between the DC magnetic field from magnetometer data (round circles) and the 2 X and Y spinning ULF components before filtering.

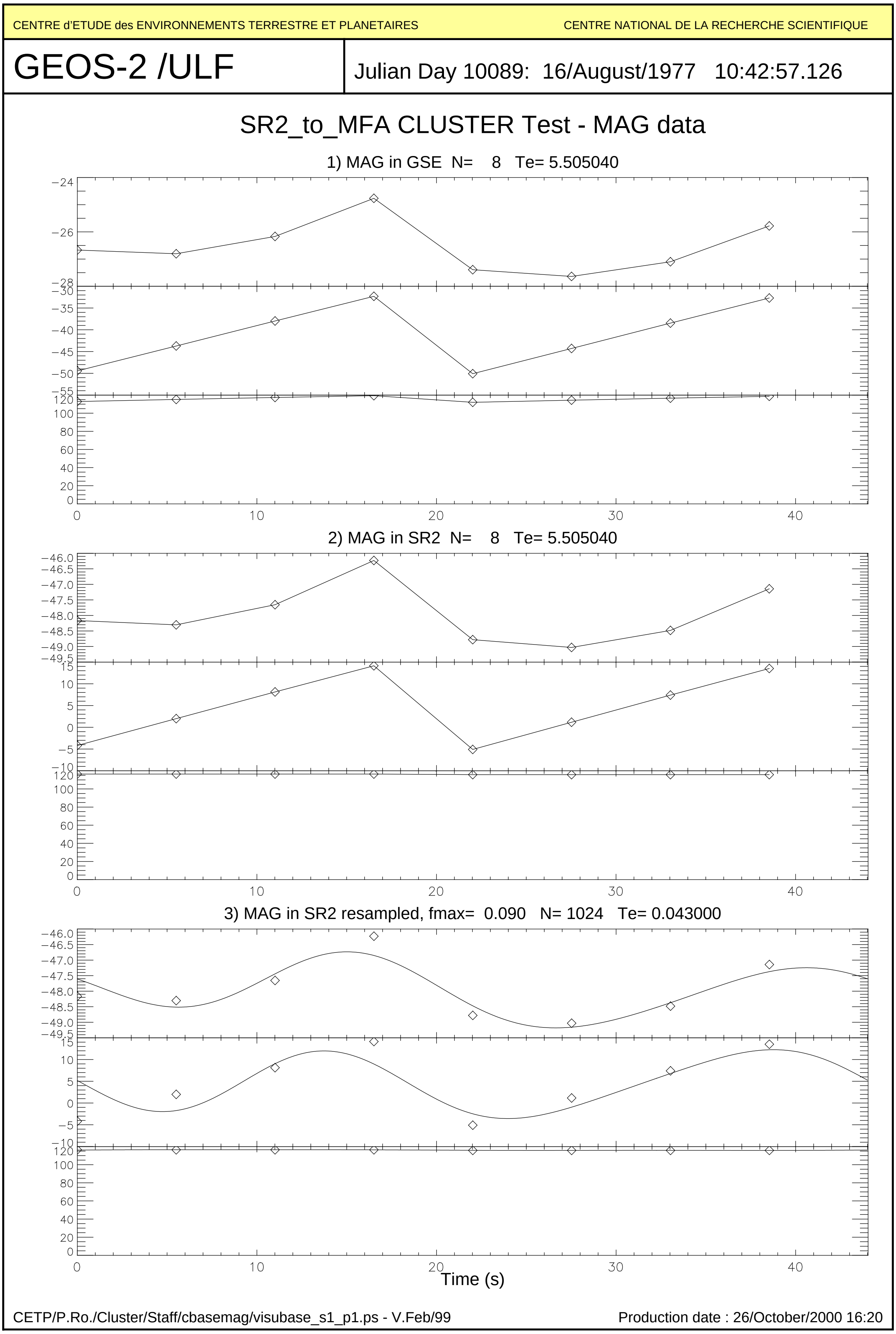
7.3.1 Figure 1: example 1 of re-sampled magnetometer data over a short time period.



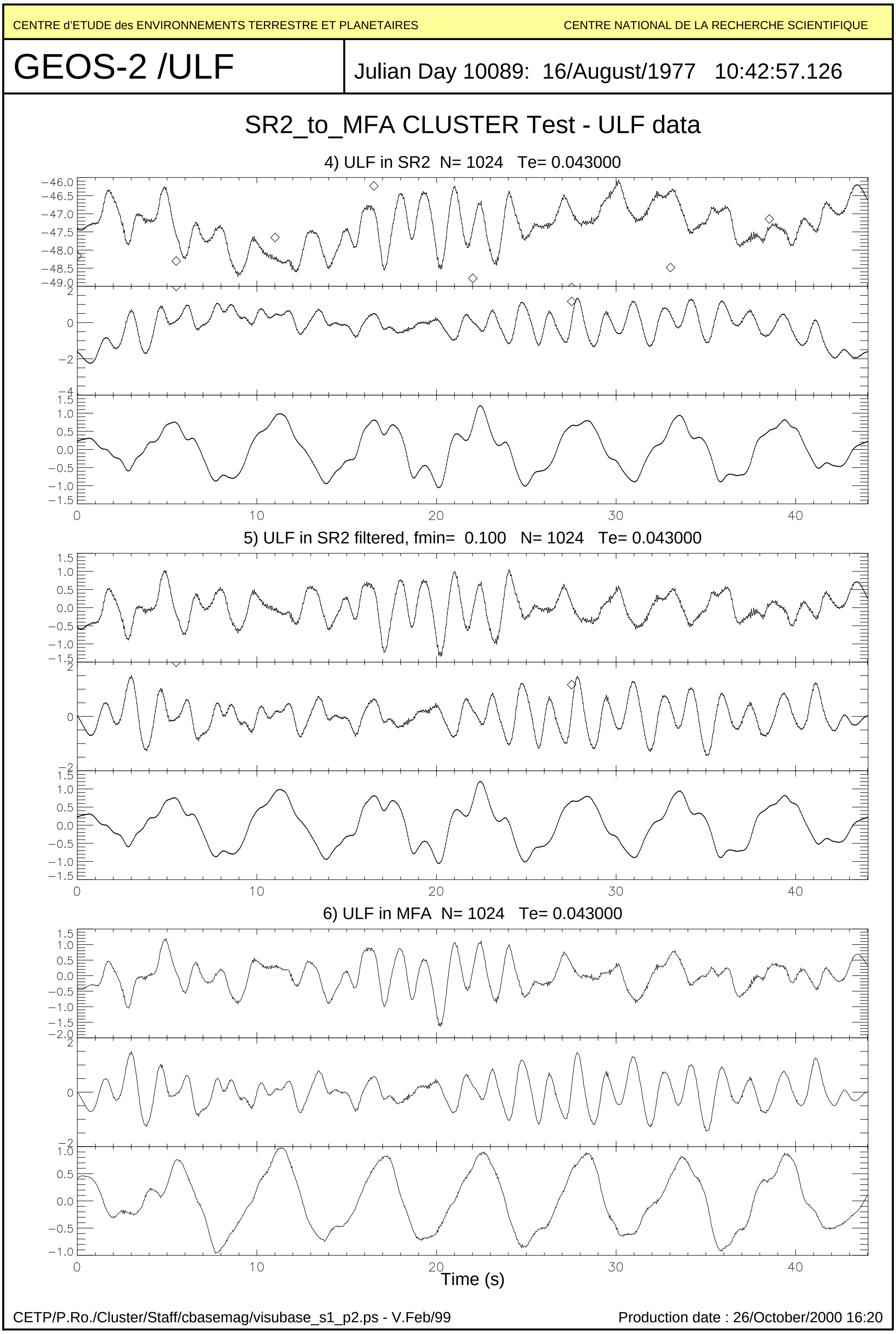
7.3.2 Figure 2: example 1 of transformed ULF waveform into the MFA system



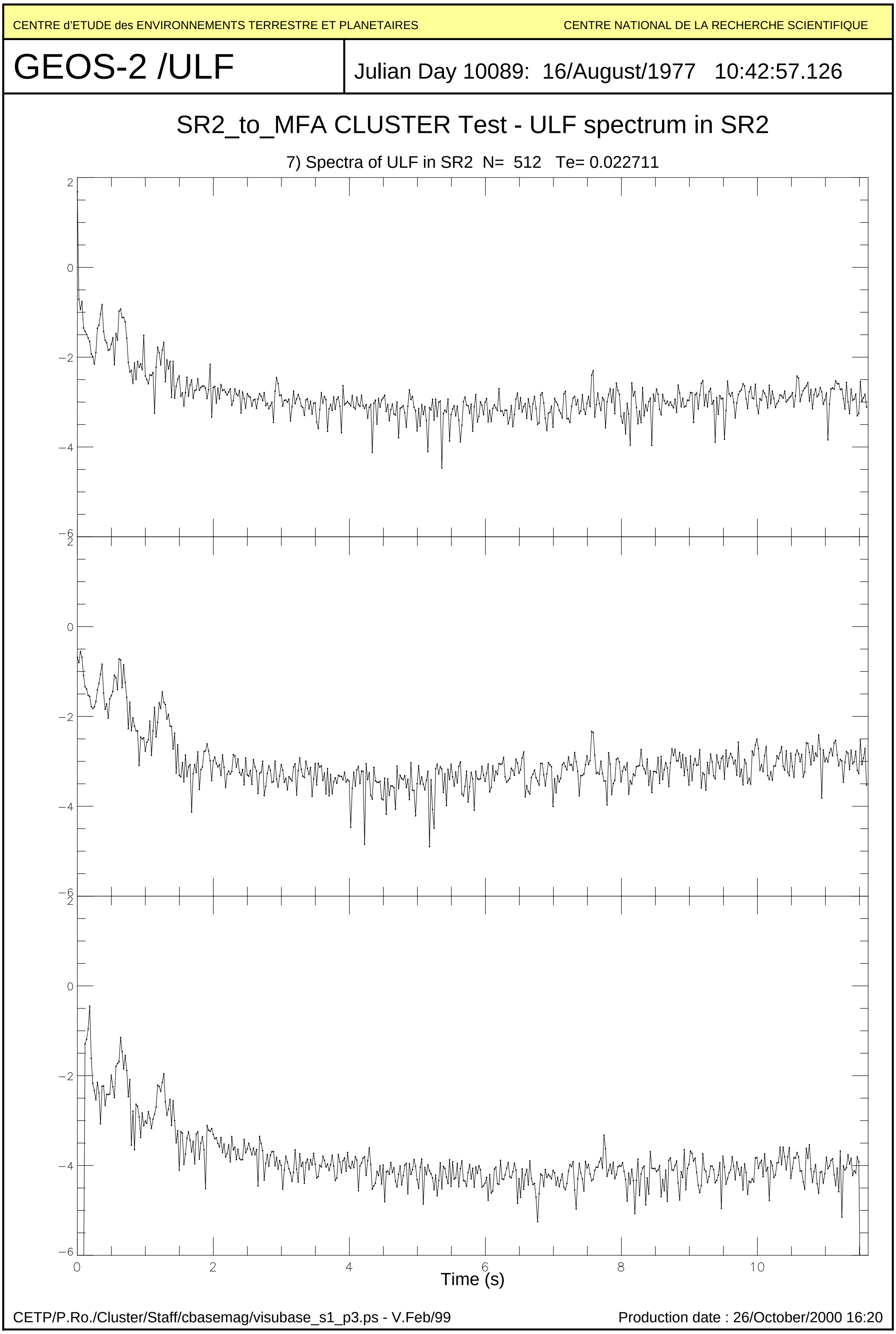
7.3.3 Figure 3: example 2 of re-sampled magnetometer data over a short time period.



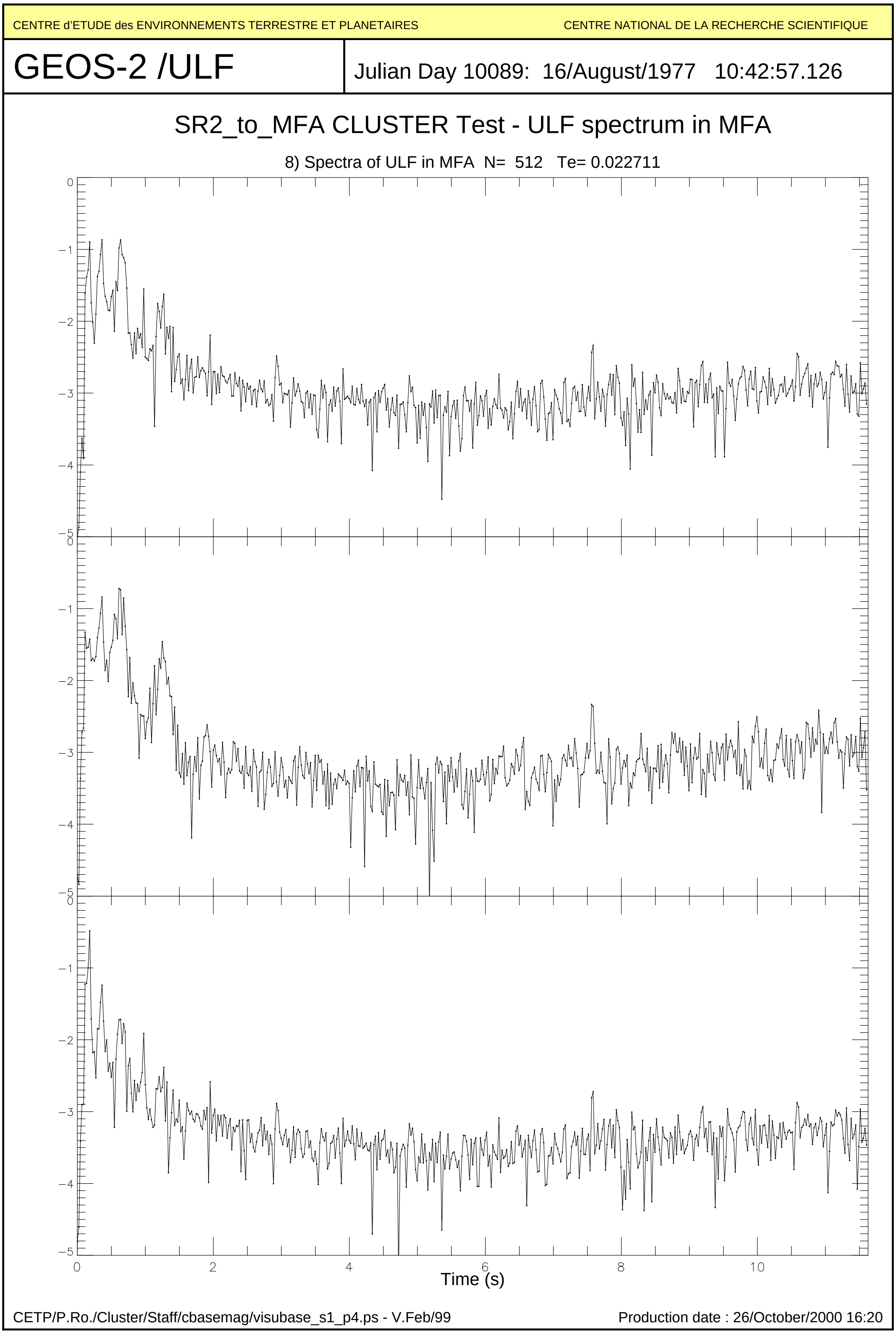
7.3.4 Figure 4: example 2 of transformed ULF waveform into the MFA system



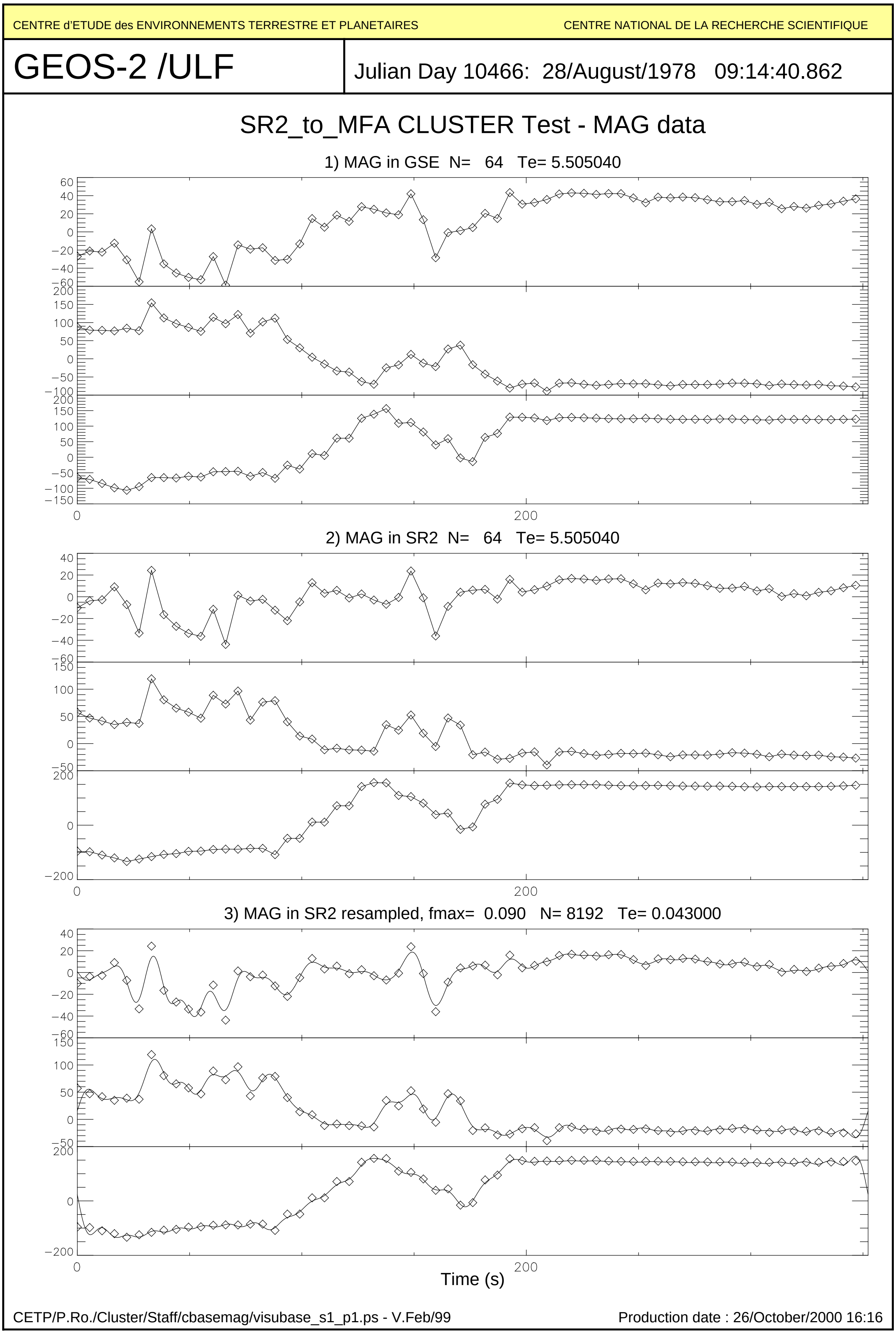
7.3.5 Figure 5: corresponding spectra in SR2 system



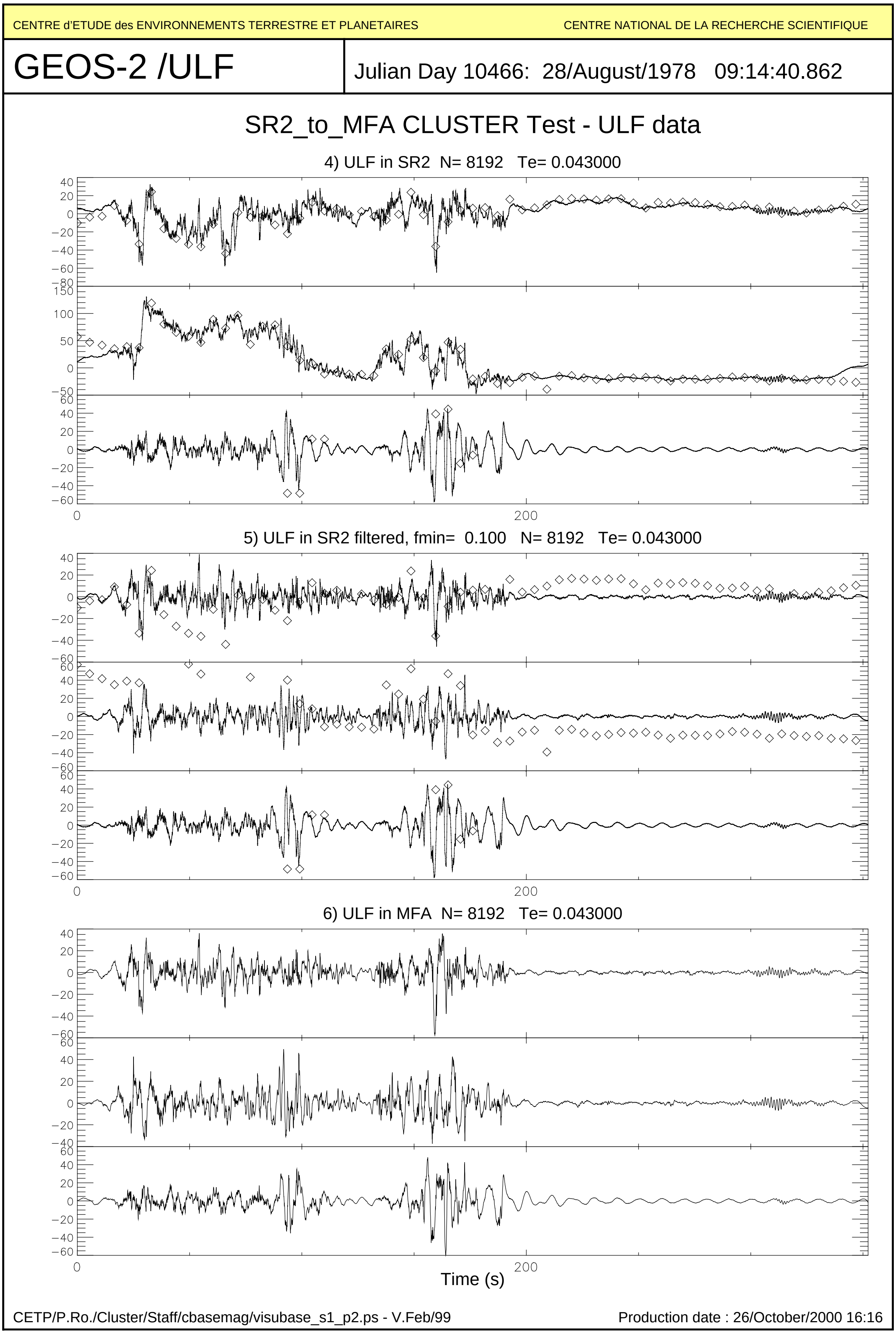
7.3.6 Figure 6: corresponding spectra in MFA system



7.3.7 Figure 7: example 3 of re-sampled magnetometer data but over a long time period.

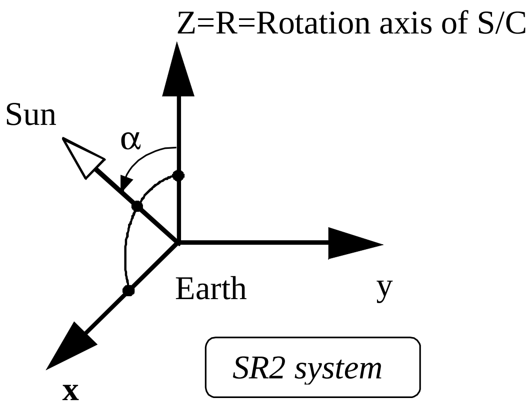


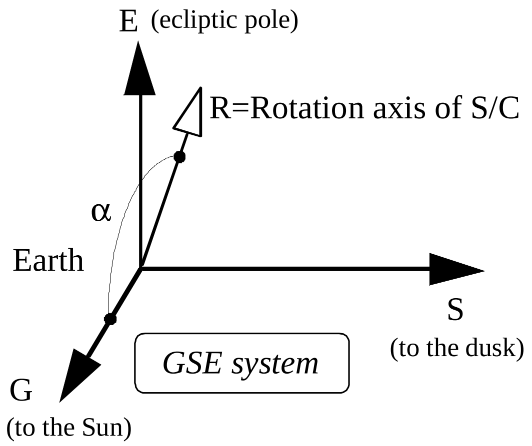
7.3.8 Figure 8: example 3 of transformed ULF waveform into the MFA system



8 APPENDIX 1: TRANSFORMATION OF GSE SYSTEM TO SR2 SYSTEM

The SR2 coordinate system (Spin Reference #2 system), also called DS system (Despun Satellite), is a system where Z is along the spin axis, and X in the plane defined by the spin axis and the direction of the Sun (see figure below). The GSE system has the X axis toward the Sun, and the Y axis in the ecliptic plane, to the dusk (see figure below).





The transformation of any vector expressed in GSE system into the SR2 system can be written as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2} = \begin{pmatrix} X1 & X2 & X3 \\ Y1 & Y2 & Y3 \\ Z1 & Z2 & Z3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{GSE}$$

where ($X1$, $X2$, $X3$) are the components of the **X** axis of the SR2 system expressed in GSE coordinates, and in the same way for **Y** and **Z** axis.

In SR2 system, Z is the rotation axis of the spacecraft, and is known in GSE system from auxiliary SPD data. We can write it as:

$$\mathbf{Z} = \begin{pmatrix} Z1 \\ Z2 \\ Z3 \end{pmatrix} = \mathbf{R} = \begin{pmatrix} Rx \\ Ry \\ Rz \end{pmatrix}_{GSE}$$

The components of the Y axis can be found by the relation **Y = (R** X **S)** / | **R** X **S** | which is very simple since the direction of the Sun **S** is the X axis of GSE system, expressed as (1,0,0), and we deduce:

$$\mathbf{Y} = \frac{1}{\sqrt{Ry^2 + Rz^2}} \begin{pmatrix} 0 \\ Rz \\ -Ry \end{pmatrix}_{GSE}$$

Finally we complete the system by computing **X** = **Y** X **Z**, so:

$$\mathbf{X} = a\begin{pmatrix} Ry^2 + Rz^2 \\ -RxRy \\ -RxRz \end{pmatrix}_{GSE} \quad \text{with} \quad a = \frac{1}{\sqrt{Ry^2 + Rz^2}}$$

Thus, the transformation of GSE coordinate into SR2 coordinates can be written as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2} = \begin{pmatrix} a(Ry^2 + Rz^2) & -a(RxRy) & -aRxRz \\ 0 & aRz & -aRy \\ Rx & Ry & Rz \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{GSE}$$

where ***R*** vector is the spin axis (or Rotation axis) expressed in GSE coordinate system, and extracted from auxiliary SPD data.

Note that, if useful, the reverse transform is simply:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{GSE} = \begin{pmatrix} a(Ry^2 + Rz^2) & 0 & Rx \\ -aRxRy & aRz & Ry \\ -aRxRz & -aRy & Rz \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2E}$$

<u>Literal expression of the transformation:</u>

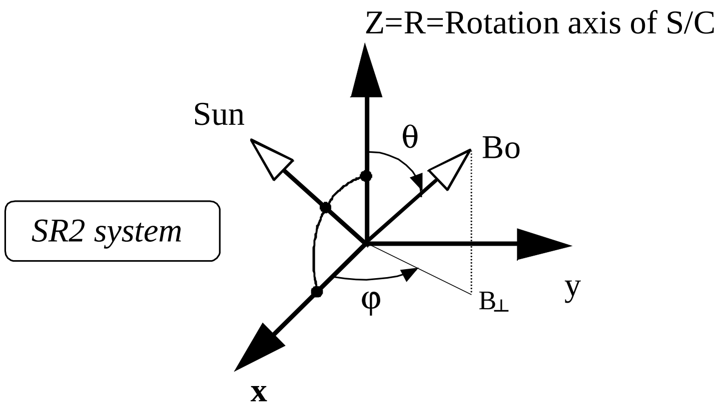
For programming use, the above matrix products can be expressed as:

```
a = 1./SQRT(Ry*Ry + Rz*Rz)

X1=  a*(Ry*Ry + Rz*Rz)
X2= -a*Rx*Ry
X3= -a*Rx*Rz

Y1=  0.
Y2=  a*Rz
Y3= -a*Ry

Z1=  Rx
Z2=  Ry
Z3=  Rz

XSR2=  X1*XGSE + X2*YGSE + X3*ZGSE
YSR2=  Y1*XGSE + Y2*YGSE + Y3*ZGSE
ZSR2=  Z1*XGSE + Z2*YGSE + Z3*ZGSE

XGSE=  X1*XSR2 + Y1*YSR2 + Z1*ZSR2
YGSE=  X2*XSR2 + Y2*YSR2 + Z2*ZSR2
ZGSE=  X3*XSR2 + Y3*YSR2 + Z3*ZSR2
```

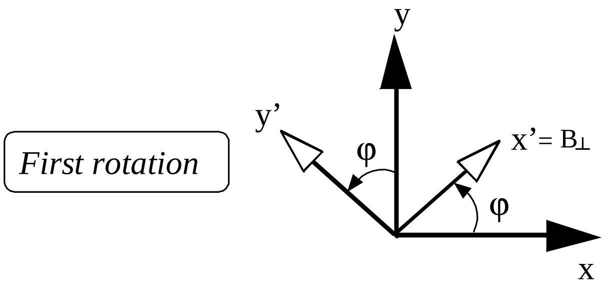
Note that `(Rx,Ry,Rz)` must be a normalised to 1 vector.

9 APPENDIX 2: TRANSFORMATION OF SR2 SYSTEM TO MFA SYSTEM

The Magnetic Field Aligned system (noted MFA hereafter) is defined as a system where Z is along the DC magnetic field, and X in the plane containing the DC magnetic field vector and the direction of the Sun (see figure below). A problem of definition can appear when the DC-MF is aligned with the direction of the Sun, which is not an unlikely case. When this situation arrives, the direction of the ecliptics could be used instead of the direction of the Sun (*** new matrix TBD).

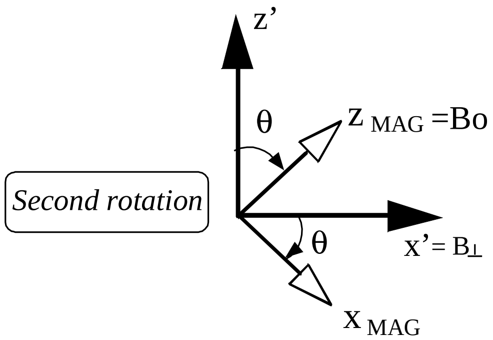


The Magnetic Field Aligned system (MFA) must have The Z axis aligned with the DC local magnetic field, noted as Bo. To obtain this alignment, we apply two rotations. The first one, in the x-y plane, align the x axis with the perpendicular component of Bo in the x-y plane:



$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

A second rotation, in the x'-z' plane, leads the new z axis to be aligned with Bo:



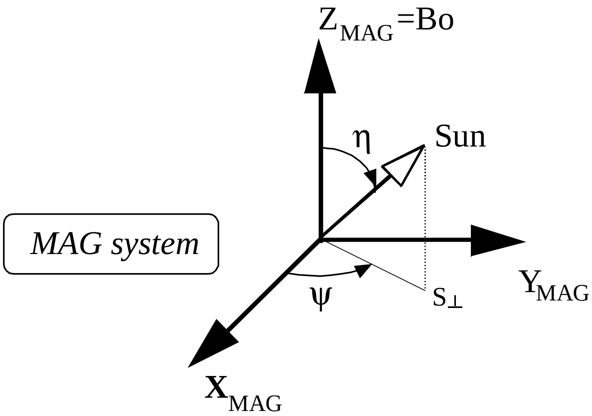
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MAG} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix}$$

and the transformation from SR2 to a new MAG is given by:

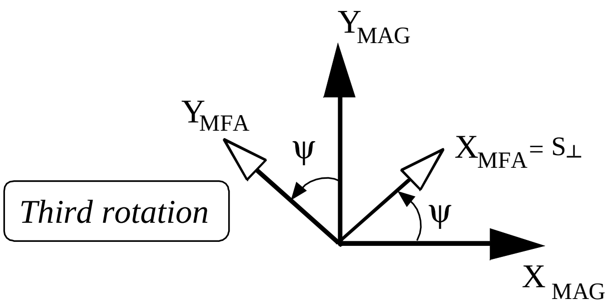
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MAG} = \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} cos\varphi & sin\varphi & 0 \\ -sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

At this level, the Z axis of new MAG system is aligned with the Bo DC local magnetic field, but the MAG system is still not the MFA system, because the MFA system is defined as having the X in the plane containing the DC magnetic field vector and the direction of the Sun. So we must apply a third rotation to pass from the MAG system to the MFA system.

This third rotation in the x-y plane depends of the direction of the Sun in the MAG system, defined by its spherical angles η and ψ :



and we pass from the MAG system to the MFA system by a rotation of ψ angle around the Z axis:



$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MFA} = \begin{pmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MAG}$$

So, the total transformation from SR2 system to MFA system is:

$$\boxed{\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MFA} = \begin{pmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} cos\varphi & sin\varphi & 0 \\ -sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}}$$

Computation of the angles θ and φ

θ and φ are respectively the polar and azimuthal angles of the Bo DC magnetic field vector in SR2 system, which has been computed in section 4. If we note Bx, By and Bz the components of Bo in SR2 system, we have:

$$sin\theta = \frac{\sqrt{Bx^2 + By^2}}{Bo}\ ,\quad cos\theta = \frac{Bz}{Bo}$$

$$sin\varphi = \frac{By}{\sqrt{Bx^2 + By^2}}\ ,\quad cos\varphi = \frac{Bx}{\sqrt{Bx^2 + By^2}}$$

Computation of the angle ψ

ψ is the azimuthal angle of the direction of the Sun in MAG system . If we note Sx_M, Sy_M, Sz_M the Cartesian components of this direction in MAG system, we have:

$$sin\psi = \frac{Sy_M}{\sqrt{Sx_M{}^2 + Sy_M{}^2}}\ ,\quad cos\psi = \frac{Sx_M}{\sqrt{Sx_M{}^2 + Sy_M{}^2}}$$

Computation of Sx_M, Sy_M, Sz_M:

To compute the components of the direction of the Sun in MAG system, we use the transformation from SR2 to MAG, computed in the beginning of this annexe, as:

$$\begin{pmatrix} Sx_M \\ Sy_M \\ Sz_M \end{pmatrix}_{MAG} = \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} cos\varphi & sin\varphi & 0 \\ -sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Sx \\ Sy \\ Sz \end{pmatrix}_{SR2}$$

where Sx, Sy, Sz are the components of the direction of the Sun in SR2 system, which leads to:

$$\begin{pmatrix} Sx_M \\ Sy_M \\ Sz_M \end{pmatrix}_{MAG} = \begin{pmatrix} Sx\ cos\theta\ cos\varphi - Sz\ sin\theta \\ -Sx\ sin\varphi \\ Sx\ sin\theta\ cos\varphi - Sz\ cos\theta \end{pmatrix}$$

Sx, Sy, Sz are deduced by the transformation of GSE to SR2 given in annexe 1, thus:

$$\begin{pmatrix} Sx \\ Sy \\ Sz \end{pmatrix}_{SR2} = \begin{pmatrix} a(Ry^2 + Rz^2) & -a(RxRy) & -aRxRz \\ 0 & aRz & -aRy \\ Rx & Ry & Rz \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}_{GSE} \quad \text{with } a = \frac{1}{\sqrt{Ry^2 + Rz^2}}$$

R vector being the Rotation axis expressed in GSE coordinate. Note that in GSE, the direction of the Sun is simply (1,0,0), thus the direction of the Sun in SR2 system is:

$$\begin{pmatrix} Sx \\ Sy \\ Sz \end{pmatrix}_{SR2} = \begin{pmatrix} \sqrt{Ry^2 + Rz^2} \\ 0 \\ Rx \end{pmatrix}$$

which leads to the result:

$$\boxed{\begin{pmatrix} Sx_M \\ Sy_M \\ Sz_M \end{pmatrix}_{MAG} = \begin{pmatrix} \sqrt{Ry^2 + Rz^2}\ \cos\theta\ \cos\varphi - Rx\ \sin\theta \\ -\sqrt{Ry^2 + Rz^2}\ \sin\varphi \\ \sqrt{Ry^2 + Rz^2}\ \sin\theta\ \cos\varphi - Rx\ \cos\theta \end{pmatrix}}$$

Literal expression of the transformation:

For programming use, the above matrix products can be expressed as follow; bx, by, bz are the components of MAG vector in SR2 system (each sample being dependant of time), while Rx, Ry, Rz are the components of the rotation axis in GSE system. The input vector is xsr2, ysr2, zsr2, the output one is xmfa, ymfa, ymfa.

```
     b0   = sqrt(bx*bx + by*by + bz*bz)
     bperp= sqrt(bx*bx + by*by)

*** first rotation

     sinphi= by/bperp
     cosphi= bx/bperp

     xp=  cosphi*xsr2 + sinphi*ysr2
     yp= -sinphi*xsr2 + cosphi*ysr2
     zp=  zsr2

*** second rotation

     sintet= bperp/b0
     costet= bz/b0

     xmag= costet*xp - sintet*zp
     ymag= yp
     zmag= sintet*xp + costet*zp

*** third rotation

     b= sqrt(ry*ry + rz*rz)

     sxm=  b*costet*cosphi - rx*sintet
     sym= -b*sinphi
     szm=  b*sintet*cosphi - rx*costet

     smperp= sqrt(sxm*sxm + sym*sym)

     sinpsi= sym/smperp
     cospsi= sxm/smperp

     xmfa=  cospsi*xmag + sinpsi*ymag
     ymfa= -sinpsi*xmag + cospsi*ymag
     ymfa=  zmag
```

10 APPENDIX 3: USEFULL PARTS OF FORTRAN CODE

```
      program ulf_to_mfa
c
c     ---------------------------------------------------------------+--
c      CLUSTER  DATA  PROCESSING
c
c     transformation of ULF data into a Magnetic Field Aligned system
c
c     Patrick ROBERT, CETP, May 2000
c     ---------------------------------------------------------------+--
c
c *** dimension of ULF, MAG and ROT array
c
c     ULF data are in SR2 system, MAG in GSE system, and ROT is the
c     rotation axis of the S/C in GSE system.
c
c input:
c
c     ulfsr2: ulf data in SR2 system
c     maggse: mag data in GSE system
c     rotgse: rot data in GSE system
c
c output:
c
c     ulfmfa: ulf data in MFA system (1024 or 8192 pt for GEOS data examples)
c
      parameter (Nu=1024, N2=Nu/2, Nm=Nu/128)
      parameter (Nr=   1)
c
      real ulfsr2(3,Nu), maggse(3,Nm), rotgse(3,Nr)
      real ulfrem(3,Nu), magsr2(3,Nm), magres(3,Nu)
      real ulfmfa(3,Nu), workar(3,Nm)
      real spesr2(3,N2), spemfa(3,N2)
      real bigtab(Nu,3,16)
c
      complex spe(Nu)
c
      character*8 cfmi, cfma
c
c      ------------------------------
c
c *** set time sampling of ULF, MAG and ROT
c
      dtu= 0.0430003
      dtm= 5.505038
      dtr=60.
c
c *** maximum frequency for resampling MAG data
c
      print*, 'fmin ULF, fmax MAG ? (ex: 0.,0.1)'
      read *,  fmin, fmax
      print*,  fmin, fmax
c
c *** fill rotgse data
c
      call fillrot(rotgse,Nr)
c
c *** fill ULF and MAG array with experimental data
c
      call fillar(ulfsr2,Nu,workar,Nm,bigtab)
      call sr2_to_gse(workar,Nm,rotgse,Nr,maggse)
c
```

```
c
c *** transform mag data into SR2 system
c
      call gse_to_sr2(maggse,Nm,rotgse,Nr,magsr2)
c
c *** re-sampling mag data from Nm points to Nu points
c
      call resampling(magsr2,Nm,dtm,magres,Nu,dtu,spe,fmax)
c
c *** removing the DC part of ULF components
c
      call remove_dc(ulfsr2,spe,Nu,ulfrem,dtu,fmin)
c
c *** transform ULF data into MFA system
c
      call sr2_to_mfa(ulfrem,magres,Nu,rotgse,Nr,ulfmfa)
c
c *** computation of ULF spectra in SR2 and MFA system
c
      call co_spectra(ulfsr2,spe,Nu,dtu,spesr2,N2,dfu)
      call co_spectra(ulfmfa,spe,Nu,dtu,spemfa,N2,dfu)
c
c *** create result file for visualization
c
      open(1,file='ulf_to_mfa.resu')
c
      call flotochar(fmin,cfmi,'(f7.3)')
      call flotochar(fmax,cfma,'(f7.3)')
c
      call whead(1)
      call wdata(1,maggse,Nm,dtm,'1) MAG in GSE')
      call wdata(1,magsr2,Nm,dtm,'2) MAG in SR2')
      call wdata(1,magres,Nu,dtu,'3) MAG in SR2 resampled, fmax='//cfma)
      call wdata(1,ulfsr2,Nu,dtu,'4) ULF in SR2')
      call wdata(1,ulfrem,Nu,dtu,'5) ULF in SR2 filtered, fmin='//cfmi)
      call wdata(1,ulfmfa,Nu,dtu,'6) ULF in MFA')
      call wdata(1,spesr2,N2,dfu,'7) Spectra of ULF in SR2')
      call wdata(1,spemfa,N2,dfu,'8) Spectra of ULF in MFA')
c
      close(1)
c
      stop 'ulf_to_mfa: normal ending'
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine gse_to_sr2(agse,n,rotgse,nr,asr2)
c
      real agse(3,n),rotgse(3,nr),asr2(3,n)
c
c     ---------------------------------------------------------------+--
c     transform agse array into asr2 array using rotgse data
c     ---------------------------------------------------------------+--
c
c *** set transform matrix with R supposed here undependant of time
c
      j=1
      rmod= sqrt(rotgse(1,j)**2 + rotgse(2,j)**2 + rotgse(3,j)**2)
c
      rx=rotgse(1,j)/rmod
      ry=rotgse(2,j)/rmod
      rz=rotgse(3,j)/rmod
c
      a= 1./sqrt(ry*ry + rz*rz)
c
      x1= (ry*ry + rz*rz)*a
      x2= -rx*ry*a
      x3= -rx*rz*a
c
```

```
      y1=  0.
      y2=  rz*a
      y3= -ry*a
c
      z1= rx
      z2= ry
      z3= rz
c
c *** tranform the waveform from gse to sr2
c
      do 10 i=1,n
c
      xgse=agse(1,i)
      ygse=agse(2,i)
      zgse=agse(3,i)
c
      asr2(1,i)= x1*xgse + x2*ygse + x3*zgse
      asr2(2,i)= y1*xgse + y2*ygse + y3*zgse
      asr2(3,i)= z1*xgse + z2*ygse + z3*zgse
c
   10 continue
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine sr2_to_gse(asr2,n,rotgse,nr,agse)
c
      real asr2(3,n),rotgse(3,nr),agse(3,n)
c
c     ---------------------------------------------------------------+--
c     transform asr2 array into agse array using rotgse data
c     ---------------------------------------------------------------+--
c
c
c *** set transform matrix with R supposed here undependant of time
c
      j=1
      rmod= sqrt(rotgse(1,j)**2 + rotgse(2,j)**2 + rotgse(3,j)**2)
c
      rx=rotgse(1,j)/rmod
      ry=rotgse(2,j)/rmod
      rz=rotgse(3,j)/rmod
c
      a= 1./sqrt(ry*ry + rz*rz)
c
      x1= (ry*ry + rz*rz)*a
      x2= -rx*ry*a
      x3= -rx*rz*a
c
      y1=  0.
      y2=  rz*a
      y3= -ry*a
c
      z1= rx
      z2= ry
      z3= rz
c
c *** tranform the waveform from sr2 to gse
c
      do 10 i=1,n
c
      xsr2=asr2(1,i)
      ysr2=asr2(2,i)
      zsr2=asr2(3,i)
c
```

```
      agse(1,i)= x1*xsr2 + y1*ysr2 + z1*zsr2
      agse(2,i)= x2*xsr2 + y2*ysr2 + z2*zsr2
      agse(3,i)= x3*xsr2 + y3*ysr2 + z3*zsr2
c
   10 continue
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine sr2_to_mfa(ulfsr2,magsr2,nu,rotgse,nr,ulfmfa)
c
      real ulfsr2(3,nu), magsr2(3,nu), ulfmfa(3,nu),rotgse(3,nr)
c
c     ---------------------------------------------------------------+--
c     transform ulfsr2 array into ulfmfa array using magsr2 and rotgse
c     ---------------------------------------------------------------+--
c
c
c *** normalise R supposed here undependant of time
c
      j=1
      rmod= sqrt(rotgse(1,j)**2 + rotgse(2,j)**2 + rotgse(3,j)**2)
c
      rx=rotgse(1,j)/rmod
      ry=rotgse(2,j)/rmod
      rz=rotgse(3,j)/rmod
c
c *** tranform the waveform from sr2 to mfa
c
      do 10 i=1,nu
c
c     set transform matrix with time-dependant terms
c
      xsr2=ulfsr2(1,i)
      ysr2=ulfsr2(2,i)
      zsr2=ulfsr2(3,i)
c
      bx=magsr2(1,i)
      by=magsr2(2,i)
      bz=magsr2(3,i)
c
      bperp= sqrt(bx*bx + by*by)
      b0= sqrt(bx*bx + by*by + bz*bz)
c
c     first rotation
c
      sinphi=by/bperp
      cosphi=bx/bperp
c
      xp=  cosphi*xsr2 + sinphi*ysr2
      yp= -sinphi*xsr2 + cosphi*ysr2
      zp=  zsr2
c
c     second rotation
c
      sintet= bperp/b0
      costet= bz/b0
c
      xmag= costet*xp - sintet*zp
      ymag= yp
      zmag= sintet*xp + costet*zp
c
c     third rotation
c
      b= sqrt(ry*ry + rz*rz)
c
```

```
      sxm=  b*costet*cosphi - rx*sintet
      sym= -b*sinphi
      szm=  b*sintet*cosphi - rx*costet
c
      smperp= sqrt(sxm*sxm + sym*sym)
c
      sinpsi= sym/smperp
      cospsi= sxm/smperp
c
      ulfmfa(1,i)=  cospsi*xmag + sinpsi*ymag
      ulfmfa(2,i)= -sinpsi*xmag + cospsi*ymag
      ulfmfa(3,i)=  zmag
c
   10 continue
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine resampling(magsr2,nm,dtm,magres,nu,dtu,spe,fmax)
c
      real magsr2(3,nm),magres(3,nu)
c
      complex spe(nu)
c
c     ---------------------------------------------------------------+--
c     re-sampling of magsr2 array into magres array with nu/nm >1
c     ---------------------------------------------------------------+--
c
c
      do 60 j=1,3
c
c *** computation of mean value of each component
c
      som=0.
c
      do 10 i=1,nm
      som=som +magsr2(j,i)
   10 continue
      rmoy= som/real(nm)
c
c *** option linear interpolation between each point of magsr2 waveform
c
      nr= nu/nm
c
      do 40 i=1,nm
      iu=(i-1)*nr +1
c
      do 50 ir=1,nr
      ku= iu +ir -1
      b1= magsr2(j,i)
      if(i.lt.nm) then
                  b2=magsr2(j,i+1)
                  else
                  b2=magsr2(j,nm)
                  endif
      magres(j,ku)= b1 +(ir-1)*(b2-b1)/real(nr-1)
   50 continue
   40 continue
c
c *** filtering above fmax
c
c     substract mean value and fill complex array spe(nu)
c
      do 20 i=1,nu
      spe(i)= magres(j,i) - rmoy
   20 continue
c
```

```
c     filtering by FFT and FFT-1
c
      call filtfon(spe,nu,dtu,0.,fmax)
c
c     re-add mean value and fill magres array
c
      do 30 i=1,nu
      magres(j,i)=real(spe(i)) + rmoy
   30 continue
c
   60 continue
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine remove_dc(ulfsr2,spe,nu,ulfrem,dtu,fmin)
c
      real ulfsr2(3,nu), ulfrem(3,nu)
c
      complex spe(nu)
c
c     ---------------------------------------------------------------+--
c     remove the DC component of the ULF waveform
c     ---------------------------------------------------------------+--
c
      df=1./(nu*dtu)
      fmax= nu*df/2.
c
      do 60 j=1,3
c
c *** computation of mean value of each component
c
      som=0.
c
      do 10 i=1,nu
      som=som +ulfsr2(j,i)
   10 continue
      rmoy= som/real(nu)
c
c *** filtering below fmax
c
c     substract mean value and fill complex array spe(nu)
c
      do 20 i=1,nu
      spe(i)= ulfsr2(j,i) - rmoy
   20 continue
c
c     filtering by FFT and FFT-1
c
      call filtfon(spe,nu,dtu,fmin,fmax)
c
c     fill output array ulfrem
c
      do 30 i=1,nu
      ulfrem(j,i)=real(spe(i))
   30 continue
c
   60 continue
c
      return
      end
c
```

```
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine whead(ifc)
c
      common /datim/ jul1,jj1,mm1,ian1,ih1,im1,is1,ims1
c
c     ---------------------------------------------------------------+--
c     write in file ifc vectim data
c     ---------------------------------------------------------------+--
c
      write(ifc,100) 'results of ulf_to_mfa.exe'
      write(ifc,100) 'Sat/Station:'
      write(ifc,100) 'GEOS-2'
      write(ifc,100) 'nbseq:'
      write(ifc,100) ' 1'
      write(ifc,100) 'Date:'
      write(ifc,200) jul1,jj1,mm1,ian1
      write(ifc,100) 'Time:'
      write(ifc,200) ih1,im1,is1,ims1
c
  100 format(a)
  200 format(4i5)
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine wdata(ifc,vectim,n,dt,com)
c
      real vectim(3,n)
      character*(*) com
c
c     ---------------------------------------------------------------+--
c     write in file ifc vectim data
c     ---------------------------------------------------------------+--
c
      write(ifc,100) com
      write(ifc,100) 'Delta T:'
      write(ifc,300) dt
      write(ifc,100) 'Nb. curves:'
      write(ifc,100) ' 3'
      write(ifc,100) 'Nb. points'
      write(ifc,200) n
      write(ifc,300) ((vectim(i,j),i=1,3),j=1,n)
c
  100 format(a)
  200 format(i6)
  300 format(3e14.6)
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine fillar(ulfsr2,nu,maggse,nm,bigtab)
c
      real ulfsr2(3,nu),maggse(3,nm)
c
      character*64 titpan(16)
      integer numpan(16),nbcpan(16),nbppan(16)
      real dtpan(16), bigtab(nu,3,16)
c
      common /datim/ jul1,jj1,mm1,ian1,ih1,im1,is1,ims1
c
c     ---------------------------------------------------------------+--
c     fill ulfsr2 and maggse arrays with GEOS data (need geoslib)
c     ---------------------------------------------------------------+--
c
```

```
      print 100
      print 100, 'readind caliulf.resu file'
      print 100

      close(1)
      open(1,file='caliulf.resu')
c
      call r_head_caliulf(igeos,nbp,f1,f2,nbseq,nbpan)
c
      do 10 iseq=1,1
      call r_sequ_caliulf(iseq,jul1,jj1,mm1,ian1,ih1,im1,is1,ims1,
     &            fs,bigtab,nbp,nbpan,numpan,titpan,nbcpan,nbppan,dtpan)
   10 continue
c
      print 100
      print 100, 'fill ulf data with ',titpan(1)
c
      do 20 i=1,nbppan(1)
      do 20 j=1,nbcpan(1)
      ulfsr2(j,i)=bigtab(i,j,1)
   20 continue
c
      print 100, 'fill mag data with ',titpan(2)
c
      do 30 i=1,nbppan(2)
      do 30 j=1,nbcpan(2)
      maggse(j,i)=bigtab(i,j,2)
   30 continue
c
      close(1)
c
  100 format(2a)
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine fillrot(rotgse,nr)
c
      real rotgse(3,nr)
c
c     ---------------------------------------------------------------+--
c     fill 1st element of rotgse array supposed undependant of time
c     fixed value is for GEOS2 at 28/08/1978 9:14, lat,lon= -0.49, 36.16
c     ---------------------------------------------------------------+--
c
      rotgse(1,1)=  0.178
      rotgse(2,1)= -0.364
      rotgse(3,1)=  0.914
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine co_spectra(ulfmfa,spe,nu,dtu,spemod,nu2,dfu)
c
      real ulfmfa(3,nu), spemod(3,nu2)
c
      complex spe(nu)
c
c     ---------------------------------------------------------------+--
c     compute spectra of ulfmfa array and fill spemod (need filtlib)
c     ---------------------------------------------------------------+--
c
      dfu=1./(nu*dtu)
c
```

```
      do 60 j=1,3
c
      do 20 i=1,nu
      spe(i)= ulfmfa(j,i)
   20 continue
c
      call fftdirp(spe,nu)
c
      do 30 i=1,nu2
      rmod= max(cabs(spe(i)),1.e-7)
      spemod(j,i)= alog10(rmod)
   30 continue
c
   60 continue
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
      subroutine flotochar(c,cx,format)
c
      character*(*) cx, format
c
c     ----------------------------------------------------------------+--
c     converts float value to character value
c     ----------------------------------------------------------------+--
c
      write(cx,format) c
c
      return
      end
c
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0XX
c
```

11 REFERENCE

[1] Requirements Specification WEC/ISDAT Coordinate Transformations, System Functional Requirements and Architectural Design Considerations, WEC Data Coordinate Systems Working Group, *by C. Harvey, P.A. Lindqvist, A. Lundgren, M. Parrot, P. Robert, S. Walker, J. Zender, Document CWD-OBSPM-DD-002, April 18, 2000, Issue 1, Rev. 5.*

[2] Coordinate Transformation Library, *by P. Robert, CRPE document DT/CRPE/1231, July 1993.*

[3] Analysis Methods for Multi-Spacecraft Data, *by Göetz Paschmann and Patrick W. Daly (Eds), International Space Science Institute,ISSI/ESA, 1998..*

12 ACKNOWLEDGEMENTS

The authors are very grateful to Simon Walker (walker@acse.shef.ac.uk) for carefully reading this document and for check and coherence with the fortran computer code.