AWARE++

Advanced Waveform Analysis Research Engine

Software Manual

Rev. 1.4

Special thanks to:
C. Lauer for providing the source code for smoothed pseudo Wigner-Ville distributions
P. Cusack and P. Bourke for the (really fast) FFT routine

Contents

1 Softwares Aim

1.1 Basic Idea

The Advanced Waveform Analysis and Research Engine (AWARE) was developed as a fast software programm, that can be used for automated calculations of Fast-Fourier-Transformation (FFT) spectra and Wavelet-Transformation (WT) coefficients. Originally the software was developed for advanced interpretation of acoustic emission signals, but the software can also be used for calculations on other transient waveform types. The latest release version AWARE++ 1.2 offers a full graphical user interface in addition to the earlier release version AWARE beta. This results in faster access to calculation parameters and faster visualization of the calculation result.

1.2 Purpose

AWARE++ is a useful tool for advanced signal analysis of ASCII-based waveforms provided in .txt format. Its main features are:

- Calculation of Fast Fourier Transforms
- Calculation of Continuous Wavelet Transforms
- Calculation of average FFT-Spectra of multiple waveforms
- Calculation of average WT-Spectra of multiple waveforms
- Automated signal convolution and deconvolution of single or multiple waveforms

The AWARE++ software can calculate FFT frequency spectra of ASCII-based signals. An implementation of several window functions is planned, but not realized yet and will be adopted in future release versions. More advantageous, the software can automatically process a large number of files. Additionally the average frequency spectra of these signals, their standard deviation, maximum and minimum values can be calculated. As a second method for signal analysis, the Continous Wavelet Transformation of ASCII-based signals is implemented. The scale and time ranges can be choosen as real values, each range adopted to the current signal's needs. The wavelet coefficients can either be saved as complex values for further calculation, or magnitude for visualization. The wavelet size is decreased from infinity to a fixed size choosen before calculation. This can boost the computation time with only a small lack of accuracy, and was already realized in the software package AGU-Vallen wavelet as well [1]. Currently only the Gabor mother wavelet is implemented, but future release versions will perhaps include some other commonly used mother wavelet types. Analogous to the FFT-options a calculation of an average WT-diagram is realized, although there is no option for standard deviation, maximum and minimum values available. This may be realized in future release versions. Apart from these basic operations, the AWARE++ software can convolute and deconvolute large number of files with a given transfer function. This could be a sensors sensitivity curve, transfer properties of a material or any other function. This option is implemented as calculation in the frequency domain, using the FFT-calculation for transformation from time to frequency space.

1.3 System Requirements

Due to the computation complexity, a fast microsoft windows based pc-system should be available. This release version was tested within an 32-bit and 64-bit Microsoft Windows XP™-environment. Fastest results will be obtained on workstations with multi-core processores or single-core CPUs with clock speeds above 2 GHz. Besides CPU-power a reasonable amount of RAM should be provided to enhance computation time. At least 1024 MB are recommended, although the maximum usable amount of 2.5 GB RAM available on 32-bit Windows-XP™systems can get used during excessive calculations. Installation requires typically less than 10 MB free disc space. In contrast, the necessary amount of free disc space during calculation depends on your input data. Many operations produce converted copies of the input data. Provide at least an amount equal to your input data size times 1.2 of free harddisc space.
Example:
Your input data are 1000 ASCII-waveforms with each 13kB datsize. This results in 13MB datasize in total. Hence you should provide at least 15.6MB free disc space for this calculation.

1.4 License Information

Part of the softwares routines are licensed to other authors, especially:

- The routine for quick folder selection was implemented by Walter Storm (c) 2006
- The FFT-routine was published by Paul Bourke 1993 [2]
- The WT-routine was published by Suzuki et al. 1996 [3]

2 Using the software

2.1 Basic Operation

Start the program executable **AWARE++.exe**. You will see the main window, like shown in fig. 2.1. Choose the option of your choice from the drop-down menu. Depending on the menu item you choose, you will be asked for several input parameters necessary for the choosen operation. For file- and folder-selection another window will pop-up, which allows you to select the path to your input data.

2.2 File

Select **File** → **View AE-file** or
Select **File** → **View FFT-file** or
Select **File** → **View WT-file**
Using "View AE-file" you can choose a single *.txt file and quickly display it. Using the other two options you can quickly re-display the results of preceeding calculations (WT-coefficents magnitude or FFT-spectra magnitude). In any case a new window will open which displays the respective content.
Select **File** → **Exit**
This option closes the application.

2.3 Calculation

For all selections a window will pop up, which allows you to select different calculation parameters. All parameters can be saved to a *.ini file using the button "Save Setup". Using the button "Load Setup" you can load an appropriate *.ini file. The default values are loaded from the **Setup.ini** file located in the installation folder.

2.3.1 Fast Fourier Transformation (single file)

Select **Calculation** → **Single FFT** from the drop-down menu. A new window will pop-up, which is shown in fig. 2.2. The only parameter necessary for the calculation is the path to the ASCII-waveform. In order to load a file click on the activated "Load" button. When you are done click the button "Calculate!". The FFT-magnitude of the calculation result is displayed in a new window. The results are saved as ASCII-data and as *.bmp file for a quick preview to the folder containing the input data.

Details on implementation:
The implementation of the Fast Fourier Transform uses procedures adopted from Paul Bourke, which were modified by Peter Cusack and are described in detail elsewhere [2].

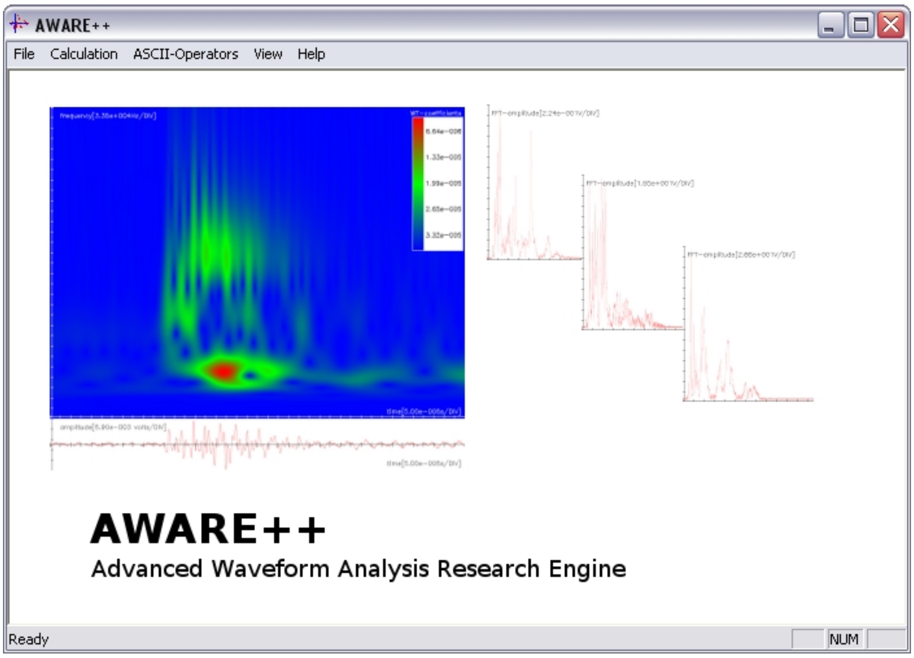

Figure 2.1: Main Window of AWARE++

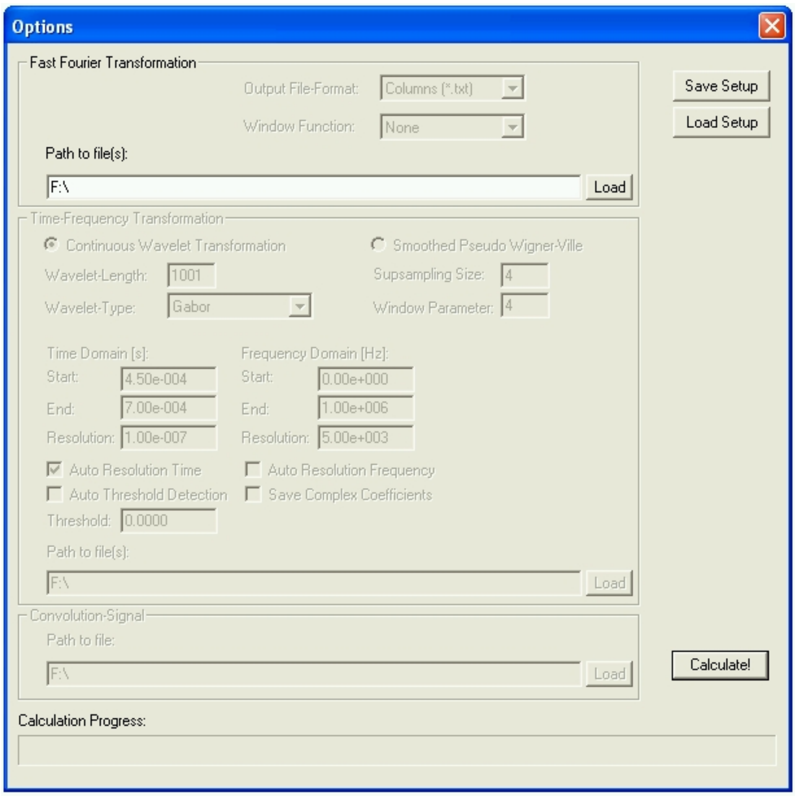

Figure 2.2: Options for FFT-calculation

2.3.2 Time-Frequency Transformation (single file)

Select **Calculation** → **Single TFT** from the drop-down menu. A new window will pop-up, which is shown in fig. 2.3. There are several parameters, that are necessary to calculate the Time-Frequency representation. Two different routines are implemented. The first one is the continuous Wavelet Transformation (WT) and the second one is the smoothed Pseudo Wigner-Ville distribution (SPWVD).
For the WT, the "Wavelet-Length" is simply the size of the wavelet in samples. In order to obtain the WT-coefficient at time position t the calculation is only performed within the interval $t \pm (waveletlength - 1)/2$. Therefore reducing the wavelet length can decrease the computation time, but should be used with care. If the wavelet length gets to short, the remaining size may not be enough to allow a convenient decay of the envelope. The next parameter to choose is the "Wavelet-Type". Currently only the Gabor mother wavelet is implemented. For the SPWVD the subsampling size and the window parameter have to be defined.
Further parameters are start, end and optionally resolution of both, time and frequency domain. These parameters should match your input data and can be entered in real values. If you use the option "Auto Resolution Time", the sampling intervall will automatically be read from the input file. Using the option "Auto Resolution Frequency" will automatically subdivide the given frequency range linearily in 100 steps. Enabling the option "Save Complex Coefficients" will store the calculation result as complex values, disabling will save the coefficients magnitude value (see also 3). In order to load a file for calculation, click on the activated "Load" button. When you are done click the button "Calculate!". The coefficients magnitude and the respective input signal is displayed in a new window. The results are saved as ASCII-data and as *.bmp file for a quick preview to the folder containing the input data.
Warning:
After the result of the calculation is displayed, the software will save the data. Depending on your input parameters (e.g. resolution) this will take some time to complete. Eventually the software will not respond! Do not force the software to close, or your calculation results may be lost!

Details on implementation:
The routine for WT-calculation follows the source-code published by Suzuki et al. and is described in detail there [3].

2.3.3 Average FFT (multiple files)

Select **Calculation** → **Average FFT** from the drop-down menu. A new window will pop-up, which is shown in fig. 2.2. In order to calculate an average FFT-spectra of several waveform files within a folder, the only parameter necessary is the path to the folder which contains the ASCII-waveforms. To load a folder for calculation, click on the activated "Load" button. When you are done click the button "Calculate!". The average FFT-magnitude is displayed in a new window. The resulting average is saved as ASCII-file **average_fft.txt** and as picture-file **average_fft_preview.bmp** for a quick preview into the subfolder avgFFTData.

Details on implementation:
The average complex FFT-amplitude $\langle A \rangle$ is defined as FFT-Amplitude A at frequency f

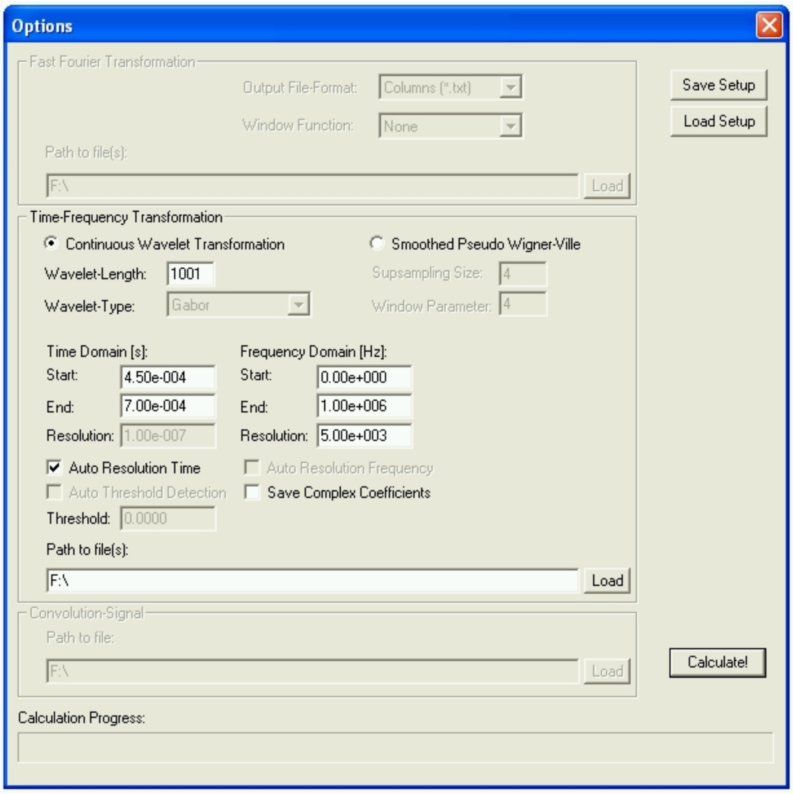

Figure 2.3: Options for WT-calculation

summed over all waveforms and divided by the number of waveforms N:

$$\langle A(f) \rangle = \frac{1}{N} \sum_{i=1}^{N} A_i(f) \tag{2.1}$$

The corresponding real and imaginary FFT-amplitudes are defined as real and imaginary part of $\langle A \rangle$ respectively. The average FFT-Magnitude $\langle M \rangle$ is defined as absolute value of the average FFT-amplitude $\langle A \rangle$:

$$\langle M(f) \rangle = \frac{1}{N} \sum_{i=1}^{N} |A_i(f)| \tag{2.2}$$

The extremal values at a frequency value f are defined as:

$$\langle M_{min}(f) \rangle = min(|A_i(f)|) \tag{2.3}$$

$$\langle M_{max}(f) \rangle = max(|A_i(f)|) \tag{2.4}$$

The standard deviation σ of $\langle M \rangle$ is calculated applying the parallel axis theorem to the conventional definition as follows:

$$\sigma(f) = \sqrt{\frac{1}{N} \left(\sum_{i=1}^{N} \left| A_i^2(f) \right| - \langle M(f)^2 \rangle \right)} \tag{2.5}$$

2.3.4 Average TFT (multiple files)

Select **Calculation** → **Average TFT** from the drop-down menu. A new window will pop-up, which is shown in fig. 2.3. In order to calculate average Time-Frequency coefficents of several waveform files within a folder, the same parameters are necessary as described in section 2.3.2. In addition the option "Auto threshold detection" is now enabled. Using this option together with an entry in the field "Threshold" can improve calculation results on time-shifted waveforms. In this case the onset of each waveform is not fixed at a specific position within the dataset, but scatteres arbitrary from waveform to waveform. To overcome this problem, the specific onset of each waveform is determined by the first crossing of the threshold value given by "Threshold" in units of the waveforms amplitude. The software now shifts the waveforms within the dataset to start at their new (common) zero position. This is defined as 20% of the timeframe given by the values for time start and end. This option should be used with care, as unrealistic threshold values can significantly influence the outcome of the calculation results! To load a folder for processing, click on the activated "Load" button. When you are done click the button "Calculate!". The average Time-Frequency coefficients magnitude and the respective input signal is displayed in a new window. The resulting average is saved as ASCII-data and as *.bmp file for a quick preview to a subfolder.
Warning:
After the result of the calculation is displayed, the software will save the data. Depending on your input parameters (e.g. resolution) this will take some time to complete. Eventually the software will not respond! Do not force the software to close, or your calculation results may be lost!

Details on implementation:

If the option for "Auto threshold detection" is disabled the average coefficients are obtained in a similar manner like the average FFT-magnitude. First of all, the average FFT-magnitude of all waveforms is calculated, as defined in equation 2.2. Afterwards, the inverse Fourier-Transformation is applied to yield the average waveform amplitude in time domain. This signal is now used to calculate the associated coefficients. This is numerically identical to a direct averaging of each waveforms coefficients $C_\psi(a,b)$, but computationally much more efficient. In the following, the mathematical proof is given for the Wavelet-Transformation.
The WT-Transformation of a single waveform i with signal amplitude a is defined as:

$$C_{i,\psi}(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} a_i(t)\psi\left(\frac{t-b}{a}\right) dt \tag{2.6}$$

Here $\psi\left(\frac{t-b}{a}\right)$ denotes the wavelet-function at scale position a and location parameter b. After application of the convolution theorem the corresponding WT-Transformation in the frequency domain can be obtained as:

$$C_{i,\psi}(a,b) = \frac{\sqrt{a}}{2\pi} \int_{-\infty}^{\infty} A_i(f)e^{jbf} \langle\psi(af)\rangle\, df \tag{2.7}$$

Here $A(f)$ denotes the Fourier-Transformation of $a(t)$ and $e^{jbf}\langle\psi(af)\rangle$ the respective Fourier-Transformation of the wavelet-function. Consequently, the average WT-coefficient at position a and b of N waveforms is:

$$\frac{1}{N}\sum_{i=1}^{N} C_{i,\psi}(a,b) = \langle C_{i,\psi}(a,b)\rangle = \frac{1}{N}\sum_{i=1}^{N} \frac{\sqrt{a}}{2\pi} \int_{-\infty}^{\infty} A_i(f)e^{jbf} \langle\psi(af)\rangle\, df \tag{2.8}$$

Using the linearity of the integral for finite boundaries f_{low} and f_{high} the summation can be switched with the integration, which results in:

$$\langle C_{i,\psi}(a,b)\rangle = \frac{\sqrt{a}}{2\pi} \int_{f_{low}}^{f_{high}} \frac{1}{N}\sum_{i=1}^{N} A_i(f)e^{jbf} \langle\psi(af)\rangle\, df \tag{2.9}$$

Hence, using the definition of the average FFT-amplitude from equation 2.1 this can finally be formulated as:

$$\langle C_{i,\psi}(a,b)\rangle = \frac{\sqrt{a}}{2\pi} \int_{f_{low}}^{f_{high}} \langle A(f)\rangle\, e^{jbf} \langle\psi(af)\rangle\, df \tag{2.10}$$

This is the Wavelet-Transformation of the waveforms averaged FFT-amplitude, which is identical to the average of the signals wavelet-coefficients for finite boundaries f_{low} and f_{high}.
If the "Auto threshold detection" is enabled, the only the time-shift of the original waveforms changes as described above, before the Fourier-Transformation is applied.

2.3.5 Convolution and Deconvolution

Select **Calculation** → **Signal Convolution** or
Select **Calculation** → **Signal Deconvolution** from the drop-down menu.
A new window will pop-up, which is shown in fig. 2.4. In order to (de)convolute several waveform files within a folder, click on the activated upper "Load" button to load a folder path. To load the respective (de)convolution signal, click on the activated lower "Load" button to

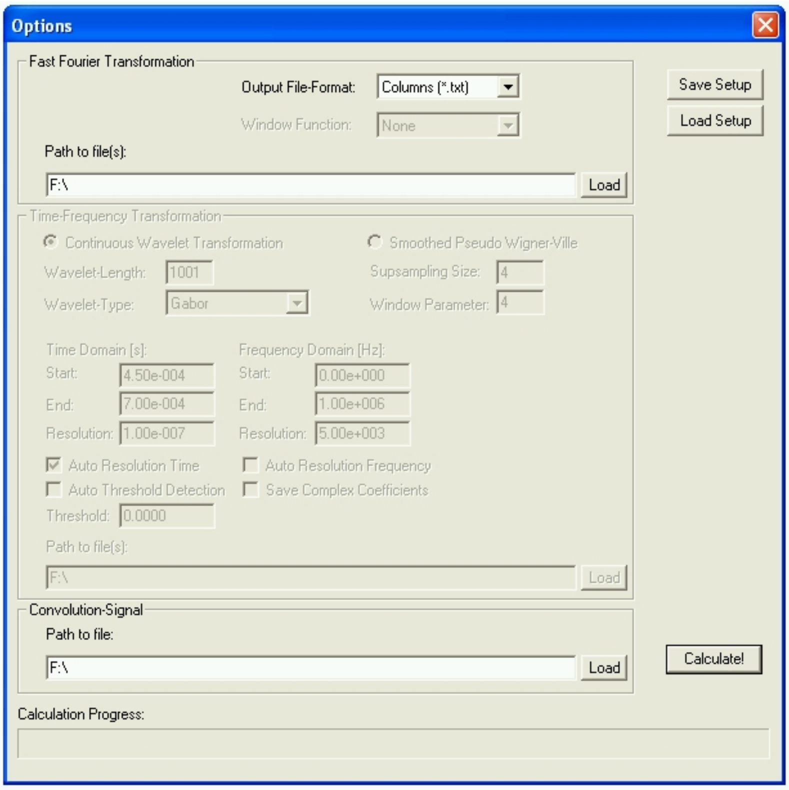

Figure 2.4: Options for signal (de)convolution

load a signal file. For the output file format two options are available. Using "Single *.txt" (de)convolutes each waveform file and stores the result into the procData subfolder. Using "Columns *.txt" appends the (de)convoluted waveform to a column separated single file instead (see also section 3), where each column corresponds to the result of a single waveform.

Details on implementation:
The software uses the application of the convolution theorem, which means the convolution takes place in the frequency domain:

$$A^*(f) = A(f) \cdot H(f); \tag{2.11}$$

Here $A(f)$ is the Fourier-Transformation of the waveforms amplitude $a(t)$ and $H(f)$ the respective convolution signal in the frequency domain. The deconvolution procedures is defined equivalently as:

$$A^*(f) = \frac{A(f)}{H(f)}; \tag{2.12}$$

2.3.6 Feature Extraction

Select **Calculation** → **Feature Extraction** from the drop-down menu.
A new window will pop-up, which is shown in fig. 2.5. This window is used to select the features which will be extracted from all waveforms included within a folder. The waveforms

processed have to be conform with the format described in section 3. The definition of the features extracted is given in table 2.1. The first column holds all features extracted from the time domain of the signals. The first feature is the (initial) "Arrival Time" of the signal, which is calculated by the AIC-picker (see below). The "Peak Amplitude" feature referes to the maximum absolute signal amplitude. The "Signal Energy" feature is calculated as defined in 2.1 using the input impedance Ω of the system. The calculation assumes a waveform as electrical signal, measured in volts and an electric impedance in Ohms to derive the correct electrical energy in Joule. The second column is for features extracted from the Fast-Fourier Transformation of the signals. The first three features "Peak Frequency" refer to the frequency position of the first, second and third intensity maximum in the FFT-magnitude $A(f)$. The peak search is done using an iterative search, were the positions of the previous peaks are ignored within an interval of $f_{peak} \pm c$. The constant c is dependent on the bandwidth of the recorded signals. Thus the results should be checked manually on selected signals before automatic application of the method. The "Frequency Centroid" is defined as "center of gravity" of the frequency distribution as given in table 2.1. The features "Weighted Peak-Frequency" refer to the respective squareroot of the combinations of "Frequency Centroid" and the three "Peak Frequencies". The "Partial Powers" are normalized intensity contributions to frequency intervals as selected within the fields near the checkbox. The frequency interval f_{start} and f_{end} used for normalization is the first and last frequency entered in the value fields.

The second column holds features extracted from the Continuous Wavelet Transformations of the signals. If any of the checkboxes within this column is activated, the WT-coefficients $C_\psi(a, b)$ of each signal are calculated within the time and frequency range given in the third column "Range settings". This will slow down the feature extraction process significantly and should only be used if these features are required. The calculation ranges should thus be optimized beforehand to capture only the significant part of the signal in time and frequency domain. The first feature "Modulus Maximus (Time)" is the abscissa of the modulus maximus as defined in table 2.1. The second feature "Modulus Maximus (Frequency)" is the ordinate and the third finally the modulus maximus itself. The "WT Peak Coefficients" are used to find the maxima of $C_\psi(a, b)$ in the time-frequency domain. The search is carried out within the frequency intervals specified in the fields next to the checkboxes, but throughout the complete time interval given in the "Range settings". Thus some information on the positions of the maxima are required beforehand and should be evaluated before automatic application of the method. The checkbox "Use AIC" in the "Range settings" column is an alternative method to define the signal onset, which uses the Akaike Information Criterion. More information on the algorithm used for AIC is given in [4]. The duration of the signal used for the investigation is then specified relative to the onset using the value entered in the "Duration" field.

For extraction of the selected signal features click on "Extract!" A new window will open to select the folder which holds the waveform fiels to be processed.

2.4 ASCII-Operators

The main purpose of this section is the treatment of several kinds of ASCII input data, to obtain special output formats, without changing the data content itself.

Parameter	Definition	Shortcut
Peak Amplitude [Hz]	A_{peak}	AMP
Signal Energy [Hz]	$\int_{t_{end}}^{t_{start}} \frac{A^2}{\Omega} dt$	ENE
Peak Frequency [Hz]	f_{peak}	PF
Frequency Centroid [Hz]	$f_{centroid} = \frac{\int f \cdot A(f) df}{\int A(f) df}$	FC
Weighted Peak-Frequency [Hz]	$\langle f_{peak} \rangle = \sqrt{f_{peak} \cdot f_{centroid}}$	WPF
Partial Power [%]	$\int\limits_{f_1}^{f_2} A^2(f)\, df \Bigg/ \int\limits_{f_{start}}^{f_{end}} A^2(f) df$ Frequency range of interest [f_1;f_2] Frequency range of investigation [f_{start};f_{end}]	PP
Modulus Maximus [units]	$max\,(C_\psi(a,b))$	MM
Modulus Maximus (Time) [s]	$t\,(max\,(C_\psi(a,b)))$	MMT
Modulus Maximus (Frequency) [Hz]	$f\,(max\,(C_\psi(a,b)))$	MMF
WT Peak Coefficient [units]	$max\,(C_\psi\,(a,[f_1;f_2]))$ Frequency range of interest $[f_1; f_2]$	WTC

Table 2.1: Definition of transient signal features in frequency and time-frequency domain

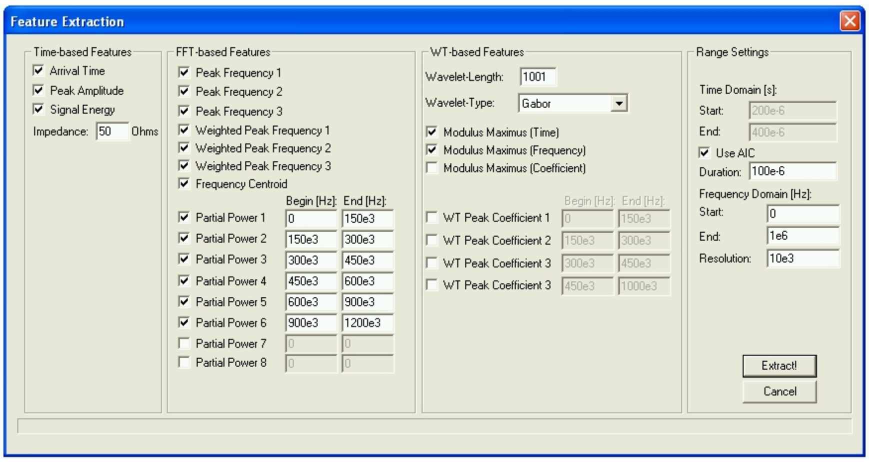

Figure 2.5: Options for Feature Extraction

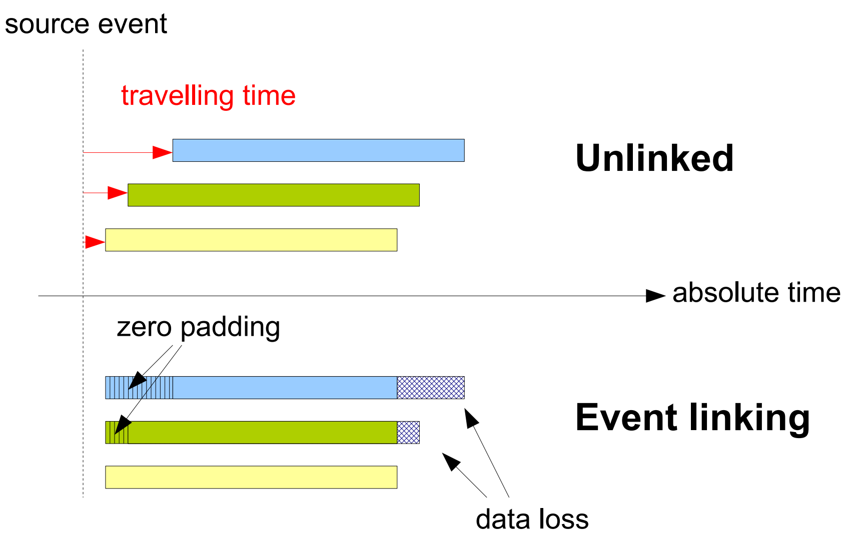

Figure 2.6: ASCII-handling scheme for event linking

2.4.1 Event Linking

Select **ASCII-Operators** → **Event Linking** and select the path to the folder, which contains the files to be linked together.
Multi-channel recording equipment for ultrasonic waves usually either uses independent triggering for each channel or one guard channel as trigger. Due to the varying source-sensor distance, the waveforms arrive at different times at each sensor and are recorded for a fixed length (i. e. fixed data size). To obtain a time-linked datafile for each singular source event recorded with all sensors, the timelines of each sensor is shifted, to obtain a common timeline for all sensors. This is shown in fig. 2.6 in an example for three sensors. The output is saved in the subfolder convData and one file per source event with incremental filenames.

Details on implementation:
At first, the absolute recording times are read from each waveforms file header. Only waveforms belonging to one source event as defined in the filenames (see section 3) are taken into account. Afterwards, a threshold based arrival time extraction is used to obtain the waveforms onset relative to the absolute recording time. The first waveform arrival is choosen as beginning of the new common time line. Relative to this waveform, the corresponding sensors datasets are shifted in time using zero padding at the beginning and data removal at the end. This is only possible if the overall recording time is sufficiently large, so that data removal only takes place after the waveforms decay. It should be noted, that the maximum loss of data points can be estimated from the maximum difference in arrival time with respect to the source-sensor distance.

2.4.2 Columns

Select **ASCII-Operators** → **Columns** and select the path to the folder, which contains the files to be linked together.
All waveforms inside the folder are pasted into one single ASCII-file. Inside the new file, each waveform corresponds to one tabulator separated column with a header definition as described in section 3. The output is saved in the convData subfolder to the file **Columns_ASCII_waveforms.txt**

2.4.3 Accumulate

Select **ASCII-Operators** → **Accumulate** and select the path to the folder, which contains the files to be accumulated.
Each waveforms time of record is read out and is assumed to reflect the waveforms time of occurence on a macroscopic scale. This neglects the time difference between time of record and the real arrival time, which could for instance be defined by threshold crossing time. In comparison to the timespan of a recording this time difference is often negligible. The software now calculates the cumulative number of waveforms over time. The output is saved in the convData subfolder to the file **Accumulated_waveforms.txt** as two tabulator separated columns.

2.4.4 AEwin (x,y,z,t) to ASCII

Select **ASCII-Operators** → **AEwin (x,y,z,t) to ASCII** and select the path to the file, which contains the (x,y,z,t) data to be processed.
This option is used to extract the positions of localized acoustic emission sources available within the program AEwin™ from Physical Acoustics Ltd. The program converts the given (x,y,z,t) datasets to a *.dat file compatible with the free software program Density Ville [5]. To this purpose it automatically detects if x-, y-, and z-positions are available and replaces non-available positions by zero-values.
To export the (x,y,z,t) data from localized acoustic emission sources, please confirm to the AEwin™ manual or follow this short description:
Inside AEwin draw a diagram showing the result of the localization. On the top menu bar a icon with a green rectangle is shown ("Area Hit/ Event Linking Mode"). Use this tool to select all localization you want to export. A new Pop-up window appears showing detailed information about your localization process. In the top menu bar use the icon that displays ("View all hits/events (unmodified)"). Another window appears displaying the ASCII-results of your localization process. Use a right-click and select the option "Copy All Text to Clipboard Ctrl+c" to save all displayed information to the clipboard. Now use notepad or similar text-editors and insert the clipboard information using "Ctrl+v" or right-click and "Insert". To be compatible with AWARE++, save the data as *.txt-file in ANSI-format option. *Example: The resulting text-file for program input should look like this example for a two-dimensional case. The number of waveform parameters (RISE, COUN, ENER, ...) or the number of sensors used to calculate the position does not influence the conversion process.*

```
 ID DDD HH:MM:SS.mmmuuun  CH  RISE  COUN  ENER  DURATION AMP A-FRQ    RMS ASL PCNTS R-FRQ I-FRQ SIG STRNGTH
* Gp# 1[1,4,3] x =    41.04, y =    20.75 dT[     0,     4] Src Amplitude = 70.0
*     0 00:18:08.8534103   1    41    54    10       356  70   152 0.0000  27     7   149   170  85.827E+03
*     0 00:18:08.8534103   4    35    65    10       398  69   163 0.0000  27    14   140   400  73.901E+03
*     0 00:18:08.8534145   3    39    50    10       373  71   134 0.0000  24     8   125   205  94.428E+03
* Gp# 1[2,1,4,3] x =    28.62, y =    8.956 dT[    11,    16,   23] Src Amplitude = 84.0
```

...

2.4.5 VisualAE (x,y,z,t) to ASCII

Select **ASCII-Operators** → **VisualAE (x,y,z,t) to ASCII** and select the path to the file, which contains the (x,y,z,t) data to be processed.
This option is used to extract the positions of localized acoustic emission sources available within the program VisualAE™from Vallen Systeme GmbH. The program converts the given (x,y,z,t) datasets to a *.dat file compatible with the free software program Density Ville [5]. To this purpose it automatically detects if x-, y-, and z-positions are available and replaces non-available positions by zero-values.
To export the (x,y,z,t) data from localized acoustic emission sources, please confirm to the VisualAE™manual.

Example:
The resulting text-file for program input should look like this example for a two-dimensional case. Make sure that only the columns "X", "Y" and "Z" (if available) are exported. Additional exported waveform parameters can influence the software functionality. The number of sensors used to calculate the position does not influence the conversion process.

```
Id      DSET   HHMMSS      MSEC          X          Y
             [hhmmss] [ms.µs ]       [cm]       [cm]
La Label 19: '11:16 PclBr'
LE      4329 00:33:12 151,6519      11,58       4,43
Ht      4330 00:33:12 151,6579
Ht      4331 00:33:12 151,6683
Ht      4332 00:33:12 151,6702
LE      4436 00:33:38 112,0811       9,73       3,89
Ht      4437 00:33:38 112,0832
Ht      4438 00:33:38 112,0957
Ht      4439 00:33:38 112,0991
LE      4954 00:33:59  38,8268       4,92       9,10
 ...
```

2.5 View

Select **View** → **Status Bar**
Enabling or disabling the option enables or disables the status bar on the bottom of the window.

2.6 Help

Select **Help** → **About AWARE++...**
Show version and copyright information.

3 Compatibility

This section should familiarize the user to the ASCII-formats used for calculation routines. Unless otherwise stated each character is important. Especially white-spaces, row- and column delimiting characters should be used exactly in the form as shown here.

3.1 Data formats

The input waveforms ASCII-format should exhibit the following form:

```
SOURCE FILE NAME: C:\test\test.dta
DATE: Friday, April 18, 2008
TIME: 17:13:15
SAMPLE INTERVAL (Seconds):   0.0000001000
UNITS: volts
CHANNEL NUMBER: 1
HIT NUMBER: 5
TIME OF TEST:     1058.7284030000

-0.00091553
 0.00030518
 0.00030518
 ...
```

This is the standard ASCII-format obtained by exporting the proprietary data format used by Physical Acoustics Ltd. software AEwin^{TM}. As the primary reason for this software was treatment of these data format, no other input type formats are supported right now, but may be adopted in the future, if requested.
The FFT-calculation on a single waveform results in the following data structure:

```
AWARE - Single FFT - (c) Markus Sause 2008
Channel: 1
Time of Record: 1.058728e+003
Timeresolution: 1.000000e-007
Frequency: Real: Imaginary:
0.000000e+000 -6.277465e-001 0.000000e+000
6.103516e+002 -3.027721e-001 4.537087e-001
1.220703e+003 1.373423e-001 3.093146e-001
...
```

In comparison the average FFT calculation has four columns in addition:

```
AWARE - Average FFT - (c) Markus Sause 2008
Frequency: Real: Imaginary: Amplitude: Standard Deviation: Minimum: Maximum:
0.000000e+000 -1.089320e+000 0.000000e+000 2.900359e+000 1.037893e+000 4.705822e-001 2.354950e+001
6.103516e+002 5.511800e+000 2.012331e+000 6.053717e+000 1.220635e+000 4.009856e+000 1.734421e+001
1.220703e+003 5.921892e+000 9.640371e-001 6.014118e+000 1.162506e+000 4.580478e+000 1.289064e+001
...
```

The output file for a single waveforms wavelet coefficients magnitude looks like this:

```
Channel: 2
Time of Record: 5.377976e+002
Timeresolution: 1.000000e-007
Frequency[Hz]/Time[s] 1.000000e-007 2.000000e-007 3.000000e-007 4.000000e-007 5.000000e-007 ...
1.100000e+004 3.164919e-001 3.162870e-001 3.160431e-001 3.158377e-001 3.155941e-001 ...
2.200000e+004 4.486387e-001 4.484015e-001 4.481202e-001 4.478821e-001 4.476018e-001 ...
3.300000e+004 5.514869e-001 5.512926e-001 5.510643e-001 5.508667e-001 5.506385e-001 ...
...
```

Here, the tabulator separated columns are the time domain, while the rows reflect the frequency domain. The first line holds the values for the time scaling, while the first column holds the frequency scaling.

The output file for a single waveforms complex wavelet coefficients should look like this:

```
Channel: 2
Time of Record: 5.377976e+002
Timeresolution: 1.000000e-007
Frequency[Hz]/Time[s] 1.000000e-007 2.000000e-007 ...
5.000000e+003 -2.684406e-007,2.936274e-007 -2.703595e-007,3.021117e-007 ...
1.000000e+004 -2.693933e-007,2.978707e-007 -2.703595e-007,3.021117e-007 ...
1.500000e+004 -2.684406e-007,2.936274e-007 -2.703595e-007,3.021117e-007 ...
...
```

The structure of the file is the same, like for coefficient magnitudes, but each coefficient entry is written in complex form. Real and imaginary part are separated by a comma.
The ASCII-structure of a transfer signal used for (de)convolution should be provided, as follows:

```
Real Imaginary
6.70484E-4 6.70484E-4
7.18419E-4 7.18419E-4
6.60511E-4 6.60511E-4
6.50789E-4 6.50789E-4
7.71938E-4 7.71938E-4
6.70374E-4 6.70374E-4
...
```

The first column provides the real part of the signal, the tabulator separated second column provides the imaginary part. In this option, the signal length (the number of rows) is crucial! The signal length has to match the padded size of the waveforms FFT-spectra. During FFT-calculation the size of the waveform is increased until it matches the next binary based length, e.g. the natural waveform length of 15360 is padded with zeros at the end until it reaches $2^{14} = 16384$. In order to use the (de)convolution described in section 2, the respective (de)convolution sgnal should exhibit the same size of 16384 datapoints. The output format for (de)convolution with option Columns(*.txt) looks like this:

```
1 2 ...
5.330227e+002 5.410450e+002 ...
1000 1000 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
0 0 ...
15360 15360 ...
3.0518E-4 2.0234E-4 ...
-3.0518E-4 8.0743E-4 ...
-3.0518E-4 -4.3454E-4 ...
-6.1035E-4 2.4590E-4 ...
...
```

The first row holds the channel, which recorded the waveform, the second row holds the time of record in seconds. The third row is the sampling rate in kHz. The following 16 rows hold 0 as entry. Row 20 finally holds the number of samples belonging to the waveform. The following rows hold the time ordered values for the signal amplitude.

ASCII-Event linking results in the following data:

```
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 6.103500e-004 0.000000e+000
5.000000e-007 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 3.051800e-004 0.000000e+000
1.000000e-006 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
1.500000e-006 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 -3.051800e-004 0.000000e+000
2.000000e-006 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
2.500000e-006 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
3.000000e-006 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
...
```

The above shows an example for six recording channels. The first column holds the time information. The other six tabulator separated columns correspond to channel one to six, respectively. Each column holds the waveforms amplitude in units as read from the source files. The input files have to be in the format **yourfilename_M_N.txt**. Here the value of N is the associated incremental source event and M is the recording channel. The file **yourfilename_1_1.txt** has to exist within the folder. The output filenames are order by events. The first linked event is saved to file **00000001.txt** and so on.

The output format for ASCII-Columns is the same like for (de)convolution with Columns(*.txt) option.

ASCII-Accumulate results in two tabulator separated columns, the first one holds the number of the waveform, the second the respective recording time.

3.2 Using other data formats

To begin calculation using data from other software, you have to change the ASCII-format according to the entries like mentioned above. If this is impossible, feel free to ask, if further formats can be included within AWARE++.

3.3 Errors and Bug Reports

For any compatibility issues, errors, bug reports or suggestions do not hesitate to contact me via mail at markus.sause@physik.uni-augsburg.de

New in AWARE++ Version 1.1:

- Added conversion procedure from AEwin™(x,y,z,t) data to DensityVille compatible format
- Added conversion procedure from VisualAE™(x,y,z,t) data to DensityVille compatible format
- Changed vertical direction of color-bar in preview-graphics of WT-diagrams
- Remove content of WT-diagrams after calculation (no more conflicts with adjacent calculations without restarting the software)

- Fixed several memory leakages

New in AWARE++ Version 1.2:

- Fixed detection of missing (x,y,z,t) columns
- Average WT-coefficients are now averaged only within investigated time-frame
- Fixed several memory leakages

New in AWARE++ Version 1.3:

- Added calculation of pseudo Wigner-Ville distribution
- Fixed detection of interpunctation in (x,y,z,t) columns
- Fixed several memory leakages

New in AWARE++ Version 1.4:

- Added option for feature extraction

Bibliography

[1] Aoyama Gakuin University (Tokyo Japan) Vallen Systeme GmbH (Munich, Germany), 2001. http://www.vallen.de/wavelet/index.html.

[2] P. Cusack P. Bourke. Fast fourier transformation, 1998. http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/dft/index.html.

[3] Y. Hayashi M. Takemoto H. Suzuki, T. Kinjo and K. Ono. Wavelet transform of acoustic emission signals. *Journal of Acoustic Emission*, 14:69–84, 1996.

[4] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[5] University Augsburg (Augsburg Germany) W. Skopalik, M. G. R. Sause, 2009. http://www.physik.uni-augsburg/exp2/downloads.en.shtml.