

## **AddDistortion**

### **Contents**

AddDistortion is supplied as a zip file, adddistortion.zip. When unzipped it should contain:

```
AddDistortion.exe  
test.wav  
CT_Pro.dll
```

and this information file.

### **Description**

AddDistortion imposes a user-specified pattern of static nonlinearity, from 2<sup>nd</sup>- to 20<sup>th</sup>-order, on a nominated Wave file. The Wave file may have any number of channels and be of any sampling rate but it must be of 16-bit resolution. In addition to generating a distorted version of the chosen Wave file, AddDistortion also generates a Windows enhanced metafile which contains graphs of the transfer characteristic used to process the input file and of the transfer characteristic error (*ie* the difference between the distorted transfer characteristic and a perfectly linear mapping of input to output). The data for these graphs is also written to a text file.

To prevent an audible click occurring at the start and end of the distorted file due to DC offset introduced by even-order nonlinearity, the user is offered the option of suppressing the DC component. If this option is selected the effect will be reflected in the transfer characteristic graphs. To prevent clipping of the output file it may be necessary to attenuate the audio data: if an attenuation is applied the user is warned of this by the program.

To prevent the introduction of distortion other than that specified, the processed output data has TPDF (triangular probability density function) digital dither added to it prior to re-quantisation at the optimum amplitude of 2LSB pk-pk. This means that the output file will have a higher noise level than the input file, although in practice this may pass unnoticed. If attenuation is applied to the output data then the signal-to-noise ratio is worsened further.

### **System requirements**

AddDistortion should run under any Windows 32-bit operating system, from Windows 95 onwards. It has been tested on Windows 95, Windows ME and Windows NT4.

To run, AddDistortion requires that the supplied CT\_Pro.dll be installed in the same directory as the executable. This file contains the runtime files of Perfect Sync Inc's Console Tools Pro (<http://perfectsync.com>) which provides enhanced control over the console window's appearance and function.

The AddDistortion console window is sized for use with screen resolutions of 1024×768 or greater. If you attempt to use it with lower screen resolution you will have to navigate the console window via scroll bars.

AddDistortion is inherently a fast-running program. Its speed in practice is determined principally by disc reading/writing and memory operations, particularly when processing large Wave files. If there is intensive hard disk activity as AddDistortion reads the input file this indicates that your computer is having to use virtual (hard disk) memory. Operation will be significantly faster if all the memory operations can be achieved within RAM. To facilitate this you may need to close other applications. If virtual memory is required even when AddDistortion alone is running, a RAM

upgrade will be needed for your computer to run it faster. Or you can chose to process shorter Wave files.

## Operation

AddDistortion first asks you to identify the Wave file to be processed. If the Wave file is not in the same directory as the executable then you must specify a complete path, *eg* c:\audio\guitar.wav. Note that the .wav extension must be included. If the specified file is not found this will be reported in an error message. Error messages will also be generated if the specified file is not a Wave file (*ie* not identified as such in its header), if it is not in PCM format or if it has any resolution other than 16-bit.

If these tests are passed, AddDistortion reports the number of channels, sampling rate and length of the specified Wave file before declaring what the name of the output file will be. This is always the name of the input file with the word 'distorted' added, *ie* c:\audio\guitar distorted.wav for the example above. If you wish to re-specify the input file you can now do so by pressing <Esc>; pressing any other key brings up the next screen.

On this second screen you input the relative level of every harmonic component from 2 to 20 (as they might be read off a sine wave distortion spectrum) and the cosine polarity of each harmonic (+ or -). For example, if the second harmonic is 23dB down relative to the fundamental, you would type '-23' on the first line. Harmonic polarity information is not so readily available but for an accurate distortion simulation it is essential. The default polarity is positive – this will be inserted automatically if you press <Enter>. If the amplitude of the harmonic is undefined because it is off-scale or buried in noise, a default amplitude of -250dBr can be selected in the same way.

Note that the specified distortion amplitudes are for a full-range signal, *ie* they are the distortion amplitudes that will be present in the output file if AddDistortion is used to process a full-scale (0dBFS) sine wave. A test signal of this description is provided in the supplied test.wav. This contains a 0dBFS sine wave at 1007.8125Hz and is exactly 4096 samples long. At the sampling rate of 48kHz, this ensures that the file contains an integer number of cycles, thereby allowing the distorted output file to be spectrum analysed using a coherent FFT technique (*ie* with no windowing). If you wish to check the accuracy of the harmonic levels generated by AddDistortion, this method will give the most accurate results. If instead you use a windowed FFT, the harmonic amplitudes will be slightly in error unless a suitable correction is applied.

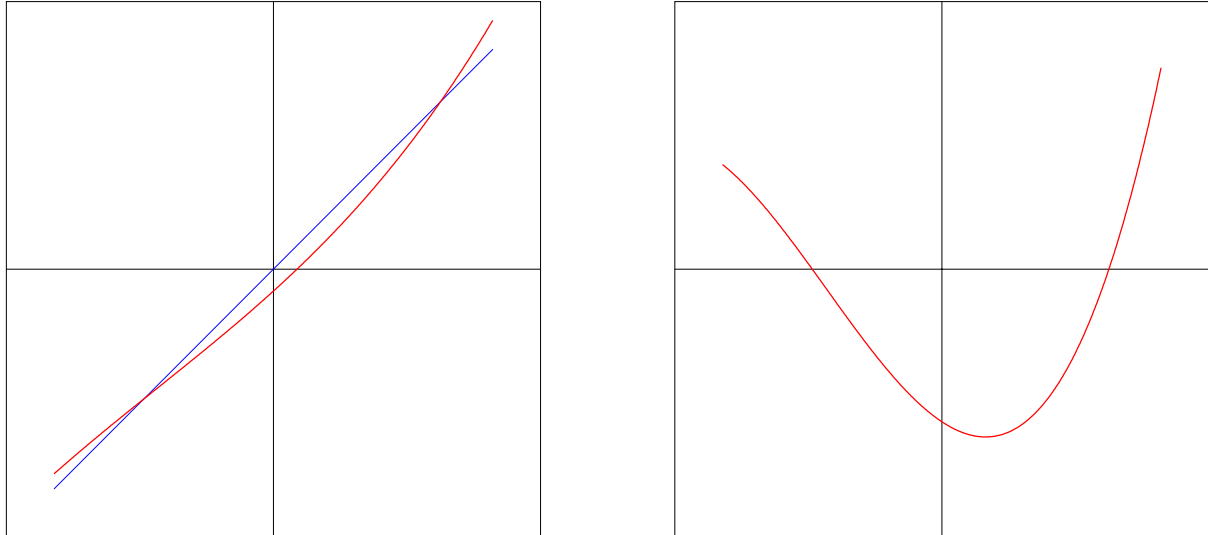
Note also that in general it is only possible to generate 2<sup>nd</sup>- and 3<sup>rd</sup>-order distortion in isolation; higher orders involve the addition of lower-order components as well. For example, AddDistortion can be commanded to add 7<sup>th</sup>-order distortion only, and on a full-scale sine wave input only the 7<sup>th</sup> harmonic will appear. As soon as the amplitude of the input sine wave is lowered, however, 3<sup>rd</sup> and 5<sup>th</sup> harmonic components will appear. This is not an artefact of AddDistortion's processing: it is an inevitable consequence of the mathematics involved. If you were to design an amplifier circuit that produced only 7<sup>th</sup> harmonic distortion at a particular sine wave input amplitude, it would behave in exactly the same fashion as the signal level was reduced.

On complex signals like music, AddDistortion will add intermodulation components as well as harmonic components. Again this mimics the behaviour of a real nonlinear device. What AddDistortion does not mimic is the frequency-dependent distortion behaviour that is typical of audio components: it imposes the same level and pattern of distortion regardless of frequency (although at higher input frequencies some of these will be removed as a result of the bandwidth limitation imposed by the signal's sampling rate).

When all the harmonic amplitudes and polarities have been specified, AddDistortion asks the user if any DC component should be suppressed. As already described, removing the DC component prevents the addition of clicks at the beginning and end of the output file. The downside is that this action may increase the amount of attenuation that has to be applied to prevent clipping of the output file.

AddDistortion now performs the calculations necessary to perform the processing and writes both the enhanced metafile containing the transfer characteristic graphs (filename transchar.emf) and a text file containing the same data (transchardata.txt) which can be imported into graphing software if required. The figure below shows an example metafile generated by AddDistortion. The left-hand graph shows the calculated transfer characteristic in red, with a blue line for comparison

representing zero distortion. When small harmonic amplitudes are specified, the red line can appear perfectly straight on this scale because its perturbations are so small. To overcome this, the right-hand graph displays on its vertical axis the difference between the red and blue lines in the transfer characteristic graph. This display is scaled according to the size of the error so that the main features of the transfer characteristic can always be visualised.



*Metafile example*

The enhanced metafile can be viewed at this stage, if desired, to check the transfer characteristic before proceeding to import the data from the input file, process it and write the output file. If it proves necessary to re-specify the harmonic amplitude and/or polarity data for any reason, this can be done now by pressing <Esc>. Pressing any other key will set in train the reading, processing and writing operations. If any attenuation is applied to the output data, this is reported once the output file has been written.

### **Licence**

This software may be freely distributed provided that it is unaltered and distributed in its entirety, including the supplied DLL, Wave file and this information file. It may not be used for any commercial purpose. No support or warranty is implied or given but if you encounter any bugs or have suggestions for improvements you are invited to post them via [freeware@audiosignal.co.uk](mailto:freeware@audiosignal.co.uk).

Keith Howard  
August 2003