How to use the QEDesign/DSPworks demo

What is on the disk?

The demonstration disk contains a reduced but working copy of both the *QEDesign* and *DSPworks* software packages.

Installing the demonstration:

The software on the disk is in a compressed form, and must be installed to run with Microsoft Windows. The installation will put the demonstration software in a directory of your specification, and install a Windows group and icons for the demo. To install the demo, follow these steps:

☞   Insert the disk in the floppy disk drive (if applicable)

☞   Type:
> **drive:\path\setup**
> then press the enter or return key

The installation process is automatic, and will ask you to confirm the location for the demo files. The installation will not work if you have insufficient disk space (about 1.5Mbytes is required).

▯**Note: The demo installation procedure will NOT change your AUTOEXEC.BAT file.**

The QEDEsign/*DSPworks* Demo Windows group with the *QEDesign* and *DSPworks* icons will then appear.

Starting the QEDesign demonstration:

🖳   To run the demo, simply point and double-click on the *QEDesign or DSPworks* icon

🖳   The appropriate window will open with menu titles appearing at the top of a blank workspace

QEDesign

QEDesign designs and analyzes digital filters. Some features of *QEDesign* are not implemented on the demonstration disk. Filters designed by *QEDesign* can be turned into assembly language code for various DSP processors, by using an optional 'code generator'.

QEDesign can also be used to analyze the transfer functions of digital systems.

With this demonstration software you can:

➢   design and analyze FIR filters

➢   design and analyze IIR filters

➢   analyze the transfer function of a digital system

A package for designing digital filters

The *QEDesign* software provides an easy to use, intuitive menu-driven system for designing and analyzing digital filters. This demo runs under Microsoft Windows although, a DOS version is also available. Benefits include:

➢   operation through simple menus with mouse control

➢   design and analysis of FIR and IIR filters, with an extensive selection of filters available

➢   analysis of Z-domain and S-domain transfer functions

➢   design of arbitrary magnitude response filters

➢   analysis of the effects of coefficient quantization

With an optional code generator, *QEDesign* can generate highly efficient filter code in assembly language for most DSP processors.

Basic Principles

QEDesign works as a standard Microsoft™ Windows application. These instructions assume some familiarity with Windows operations, but guide you through the operations step by step. If you are a complete Windows novice, you would benefit from some practice with Windows before using the demo.

Main Menu bar and pull-down menus

The main menu bar consists of the following options:

File	to load previously stored filter or system analysis specifications
Window	standard Windows operations to select graphics presentation
Design	to specify the desired filter characteristics
Analysis	to analyze the S-domain or S-domain
Options	Various options
Start	immediately starts the filter design and displays the results

Selecting the plots for analysis

QEDesign allows you to choose the analysis plots to view during filter design. For all these examples we will choose to view all plots. ***QEDesign*** will display only the plots that are relevant to the method of filter design currently in use.

☞   Click on the **Window** menu to choose the plot options

☞   Click on **Select Plot**s to choose the plots for viewing

☞   Select all plots by clicking against the line until an X appears

Designing an FIR filter

In a Finite Impulse Response (FIR) filter, each output sample is a function only of the current and previous input samples. Previous output samples do not affect the current output sample. There is no feedback, so FIR filters are always stable. There are several design methods for FIR filters. ***QEDesign*** supports the most useful - window design and equiripple (also called Parks McClellan) design.

FIR design with window method

Why are we doing this?

This example introduces some of ***QEDesign***'s features by designing FIR filters using the window method. ***QEDesign*** provides a large number of window functions from which to choose: each window has its own advantages and disadvantages.

In this example, we will design a lowpass filter using a Kaiser window. The Kaiser window designs an FIR filter whose characteristics approximate those of an analog Butterworth filter.

An Aside:

The window method of filter design is based upon the Fourier series. It is possible to represent a frequency function as a Fourier series, whose coefficients represent the coefficients of the filter. To form a causal filter, the Fourier series is truncated and shifted. Truncating the Fourier series causes a phenomenon called the "Gibbs effect"; a spike occurs wherever there is a discontinuity in the desired magnitude of the filter. To counteract this, the filter coefficients are convolved in the frequency domain with the spectrum of a 'window' function - thus smoothing the edge transitions at any discontinuity. This convolution in the frequency domain is equivalent to multiplying the filter coefficients with the window coefficients.

Selecting the filter design method

☞   Click on the **Design** menu to choose the filter design method
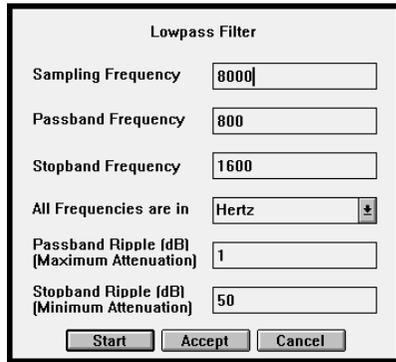
☞   Choose **FIR Design (Windows)**

This sets **QEDesign** to design FIR filters using the window method.

Choosing the filter type

The example filter is a Kaiser window lowpass
filter with an 800 kHz cutoff frequency, a
stopband from 1600 Hz and a 50 dB
attenuation. It is designed for a sampling
frequency of 8 kHz, ripple in the passband
is allowed to be up to 1 dB.

☞   Click on the **Design** menu to choose
        the filter type.

☞   Choose a **Lowpass** filter

A data entry screen appears. for you to enter the
desired filter characteristics. Set these as shown
on the right.

☞   Start the filter design by clicking on **Start**

Lowpass Filter

Sampling Frequency	8000
Passband Frequency	800
Stopband Frequency	1600
All Frequencies are in	Hertz
Passband Ripple (dB) (Maximum Attenuation)	1
Stopband Ripple (dB) (Minimum Attenuation)	50

[Start]   [Accept]   [Cancel]

Choosing the window function

QEDesign supports a wide selection of window functions. These are listed in a scrolling window after you click on
the Start button.

The available window functions include:

➣   Rectangular
➣   Hanning
➣   Blackman
➣   Harris Flat top
➣   3 term cosine
➣   Minimum 3 term cosine
➣   4 term cosine
➣   Minimum 4 term cosine
➣   Kaiser
➣   Dolph-Tschebyscheff
➣   Taylor
➣   Gaussian
➣   Triangular
➣   Hamming
➣   Exact Blackman
➣   Good 4-term Blackman Harris
➣   3 term cosine with continuous 3rd derivative
➣   4 term cosine with continuous 5th derivative
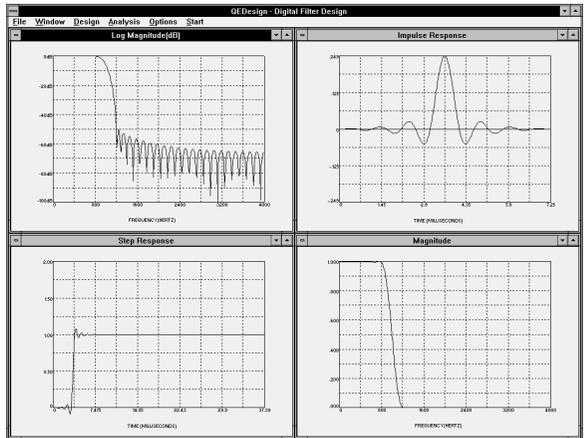
The Kaiser window is near the bottom of the list.

☞   Scroll down the list by clicking on the bottom of the vertical scroll bar

☞   Select the **Kaiser** window

QEDesign calculates the filter coefficients and displays the characteristics in the plots we selected.

Each plot can be moved, resized and closed as normal for a Microsoft Windows application.

Tiling and Cascading the display windows

☞   Open the **Windows** menu

☞   Select **Tile** to see the tiled display

☞   Open the **Windows** menu again

☞   Select **Cascade** to see the cascaded display

Each window can also be placed and sized manually.

FIR design with Equiripple method

Why are we doing this?

This example illustrates how *QEDesign* can design FIR filters using the Equiripple method. This method has some advantages (and some disadvantages) over the window method. It also demands very high numerical precision, so *QEDesign* uses 64-bit double precision arithmetic for all calculations.

In this design we will design a bandpass filter with the Equiripple method and analyze its response.

An Aside

The window design method starts with an infinite (in practice, very long) series that is truncated to the desired length. Coefficients beyond the truncation are simply ignored. The window removes even more information. The Equiripple method optimizes the series for a given number of coefficients. The method, first programmed by Parks and McClellan in FORTRAN, uses an optimization algorithm called the Remez exchange algorithm. Instead of describing the coefficients by a Fourier series, they are described using a polynomial series. This design method allows sharper transitions with better stopband attenuation than the window method - but there is a ripple in the passband. This type of design normally produces equiripple designs, where the ripples in the passbands and stopbands are of equal height within any one band.

Selecting the filter design method

☞   Click on the **Design** menu to choose the filter design method

☞   Select **FIR Equiripple Design**

This sets *QEDesign* to design FIR filters using the Equiripple method.

Choosing the filter type

The example filter is a bandpass filter with a passband from 900 Hz to 1100 Hz, stopbands up to 800 Hz and above 1200 Hz, and a 45 dB attenuation. It is designed for a sampling frequency of 6 kHz. Ripple in the passband is allowed to be up to 1 dB

☞   Click on the **Design** menu to select the filter type

☞   Select a **Bandpass** filter

A data entry screen appears for you to enter the desired filter characteristics. Enter these as shown on the right

☞   Start the filter design by clicking on **Start**

Bandpass Filter

Sampling Frequency	6000	
Passband Frequency	900	1100
Stopband Frequency	800	1200
All Frequencies are in	Hertz	
Passband Ripple (dB) (Maximum Attenuation)	1	
Stopband Ripple (dB) (Minimum Attenuation)	45	

Start   Accept   Cancel

Modifying the equiripple filter characteristics

QEDesign lets you specify the number of taps and choose functions for the shape of the transition bands. These parameters will modify the basic Equiripple characteristic. **QEDesign** can also design antisymmetric FIR filters.

QEDesign can create FIR filters that are symmetric or antisymmetric. For bandpass filters, the shape of the transition bands can be specified to be a raised cosine, or a root raised cosine.

A choice of transition band is only allowed for bandpass filters.

In this example we accept **QEDesign**'s suggestions which are:

 ▌ 93 taps

 ▌ Symmetric FIR filter

 ▌ Unconstrained transition band

 ▌ 0 dB stopband sidelobe attenuation

An Aside

Although **QEDesign** allows you to design antisymmetric filters for all filter types, in practice the only usable antisymmetric filters are bandpass and even order highpass filters: for example, a lowpass antisymmetric filter design will have a magnitude of zero at DC.

☞ Click on the **OK** button to start the filter design.

QEDesign calculates the filter coefficients and displays the chosen analysis plots.

An Aside

The equiripple design method uses an optimization algorithm and so may fail to meet specifications in some cases. Typically, the gain will exceed 1.0 even though this was specified as the maximum gain. The magnitude and log magnitude plots will show how far from the desired value the result is.

Common causes for failure to meet specifications are:

➢ Too narrow transition band

➢ Very narrow passbands

➢ Very narrow stopbands

➢ Non-symmetric pass and stopbands
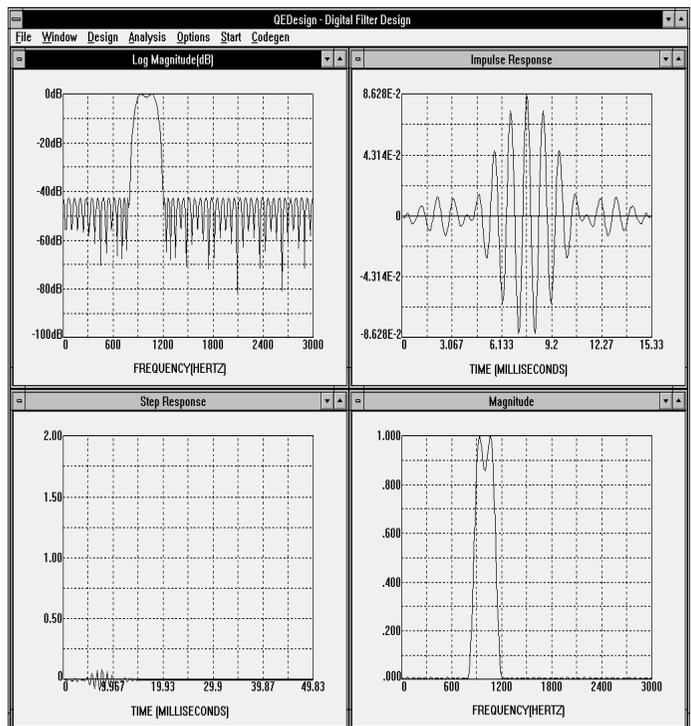
IIR filter design

Why are we doing this?

This example shows how *QEDesign* is used to design IIR filters. *QEDesign* provides five analog filter prototypes and three methods for transforming them from the analog to the digital domain. *QEDesign* also provides an allpass filter where the group delay can be specified as an arbitrary function: this filter is designed directly in the digital domain. In this example, we will design a bandpass filter by taking a classic analog filter (Butterworth) and mapping this into the digital domain.

An Aside

In an Infinite Impulse Response (IIR) filter, each output sample is a function of previous output samples, as well as of the current and previous input samples. The transfer function for such a filter has both poles and zeros. For the filter to be stable, the poles must be inside the unit circle. IIR filters can be designed in the analog domain (s-plane) using analog filter prototypes and then mapped into the digital domain (the z-plane); or they can be designed directly in the digital domain.

Selecting the filter design method.

☞   Click on the **Design** menu to choose the filter design method

☞   Choose **IIR Design**

This sets *QEDesign* to design IIR filters

Choosing the filter type

The example filter is a Bandpass filter, with a passband from 900 Hz to 1100 Hz, stopbands up to 700 Hz and above 1300 Hz, and a 25 dB attenuation. It is designed for a sampling frequency of 6 kHz. Ripple in the passband is allowed to be up to 3 dB. Since we are going to design a Butterworth filter, and the passband of a Butterworth filter is by definition flat, the 3 dB should be taken to specify the maximum attenuation of the edges of the passband as the filter starts to roll off.

☞   Click on the **Design** menu to
         choose the filter design method

☞   Select a **Bandpass** filter

A data entry screen appears for you to enter the desired filter characteristics. Enter these as shown on the right

☞   Start the filter design by clicking
on   **Start**

Bandpass Filter

Sampling Frequency	6000	
Passband Frequency	900	1100
Stopband Frequency	800	1200
All Frequencies are in	Hertz ▼	
Passband Ripple (dB) (Maximum Attenuation)	1	
Stopband Ripple (dB) (Minimum Attenuation)	45	

[Start]   [Accept]   [Cancel]

Choosing the analog filter prototype

QEDesign lets you choose the analog filter prototype. You can also specify a filter order if you wish.

The analog prototypes for IIR filters are:

➤ Butterworth   ➤ Bessel

➤ Tschebyscheff   ➤ Inverse Tschebyscheff

➤ Elliptic

In this case we will select a Butterworth filter, and accept *QEDesign*'s suggestion for the filter order.

☞ Click on the **Butterworth** filter line to choose a Butterworth filter

☞ Click on the **OK** button to start the design

QEDesign calculates the filter coefficients and displays the chosen analysis plots.

For an IIR filter the Pole/Zero plot is of interest.

An aside

The Butterworth filter is maximally flat, and shows a linear phase response in the passband.

Note that *QEDesign* allows you to choose between the bilinear transformation and the impulse invariant transformation methods for transforming from the analog filter prototype to the digital domain. The transformation method and some other useful options, are selected from the **Options** menu.
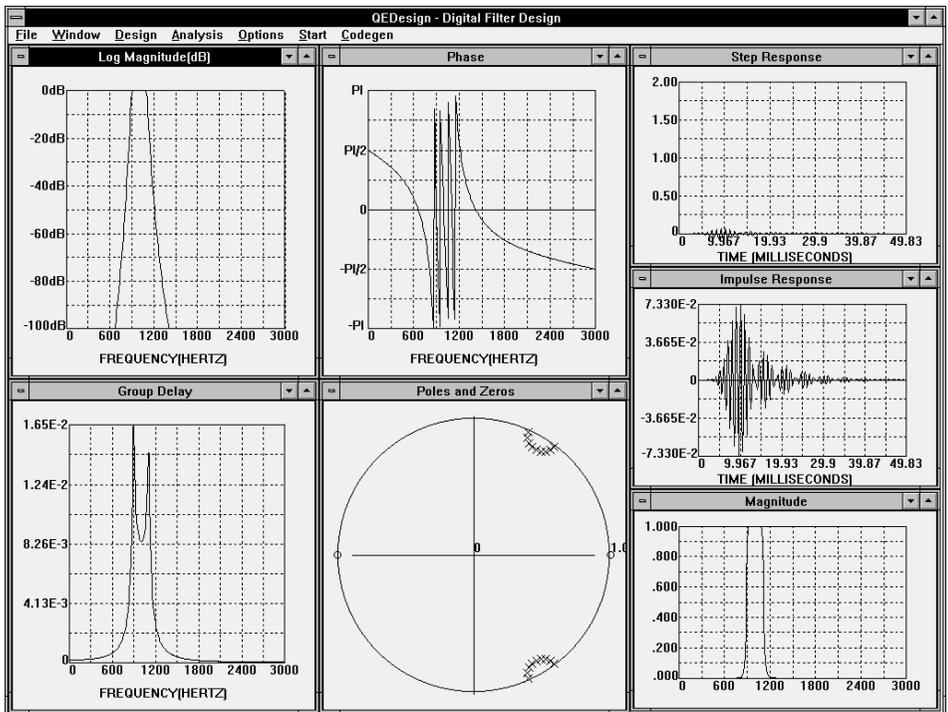
Checking the effects of Quantization

Why are doing this?

This example shows how *QEDesign* enables the user to analyze the effects of finite word length when a filter is implemented on a particular processor. We will analyze the effects of quantization for a 16-bit processor, on the Butterworth filter we just designed.

QEDesign lets you choose between a floating point or fixed point realization.

In floating point quantization, all the coefficients will have the same number of bits to represent their fractional part. This represents the best possible implementation on a processor of given word length. In fixed point quantization, the number of bits representing the smaller coefficients is less than the number of bits representing the larger coefficients. This is the more typical case for implementation on a fixed point processor, with no attempt to simulate floating point or block floating point arithmetic.

An Aside

If a filter is to be implemented using a particular processor, the coefficients will have to be 'quantized' to the word length of that processor. Quantization perturbs the location of the poles and zeros. There are also effects from limited precision arithmetic, and a need to scale the coefficients to avoid overflow.

In this example, we will choose to simulate the effects of quantization using a 16-bit fixed point processor realization of the filter.

☞   Click on the **Options** menu to choose quantization options

☞   Click on the **Quantized coefficients** to select quantization

We will investigate the effects of implementing this filter on a 16-bit processor.

☞   On the Number of Significant Bits line, enter **16** to specify the word length

☞   Click on **Fixed Point Fractional Quantization** to choose a fixed point implementation

Selecting the practical implementation

QEDesign lets you choose from various methods for the practical implementation of the filter. We will choose to simulate the filter as a 'Direct Form I' implementation (this is the transpose of Direct Form, hence its name in *QEDesign*.

☞   Click on **Cascade Transposed Second Order Sections** to choose the implementation method

☞   Click on **OK** to select the quantization options

☞   Click on the **Start** button

QEDesign calculates the effects of quantization and displays the chosen analysis plot. For this example the IIR Quantization Analysis plot is of interest.

IIR Quantization Analysis	
Input Noise Power (dB)	-155.28621
Output Noise Power (dB)	-135.30814
Noise Power Std Deviation	1.7162997e-07
Number Of Significant Bits	21
SNR before Signal Scaling (dB)	126.4326
SNR after Signal Scaling (dB)	111.32088

Digital Signal Analysis

Why are we doing this?

This example shows how *QEDesign* can be used to analyze the transfer function of a digital system. *QEDesign* can calculate the phase and frequency response; the impulse and step response; the pole/zero locations; and the group delay. In this example, we will analyze a system transfer function which is specified as an FIR filter.

Selecting Z or S domain

QEDesign can accept transfer functions specified in the S-domain (analog transfer functions), or in the Z-domain (digital transfer functions). If the transfer function is specified in the S-domain, it is first digitized. In this example, we will analyze a transfer function which is specified in the Z-domain. For the sake of simplicity, we will choose to analyze a transfer function which is specified as a symmetric FIR filter. This also lets us enter a minimum of data to specify the transfer function.

☞ Click on the **Analysis** menu to select S or Z domain analysis

☞ Click on **Z Domain** to choose Z domain input

Choosing how the transfer function will be specified

The transfer function can be specified in a number of ways. If the transfer function is specified in the Z-domain, then it can be input as:

☞ Ratio of polynomials   ☞ Symmetric FIR Filter

☞ Poles and Zeros   ☞ Antisymmetric FIR filter

☞ Product of second order sections

☞ Click on the **Analysis** menu to select how the transfer function will be entered

☞ Click on **Symmetric FIR Filter** to choose the transfer function specified as an FIR filter

In this example, the FIR transfer function is specified for a sample rate of 8 kHz, and consists of 16 coefficients.

☞ Click on the **Accept** button to confirm the entered parameters

➢ Each of the different methods for specifying the transfer function has their own data entry screen.

Entering the transfer function

Since this transfer function is specified as an FIR filter, *QEDesign* provides a screen to enter the filter coefficients. For a symmetric FIR filter, we only need enter half the coefficients (another reason we chose this example).

The coefficients for this example are entered as follows:

H(1) = .1495361328E-01 = H(16)
H(2) = -.4577636719E-02 = H(15)
H(3) = .3570556641E-01 = H(14)
H(4) = -.4891967773E-01 = H(13)
H(5) = -.1205444336E-01 = H(12)
H(6) = .8135986328E-01 = H(11)
H(7) = .1974792480E+00 = H(10)
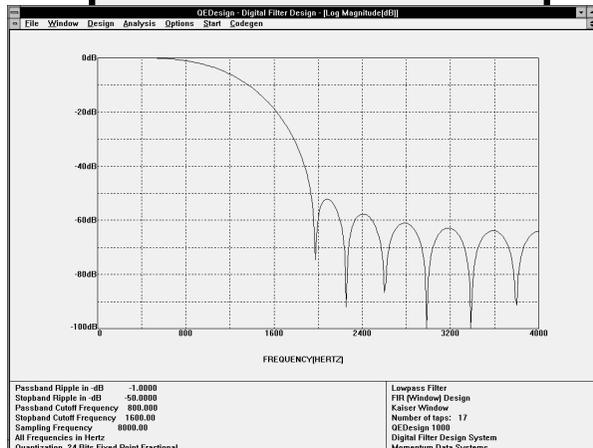H(8) = .2779846191E+00 = H(9)

☞ Click on the **Accept** button

QEDesign calculates the characteristics of the system and displays the chosen analysis plots.
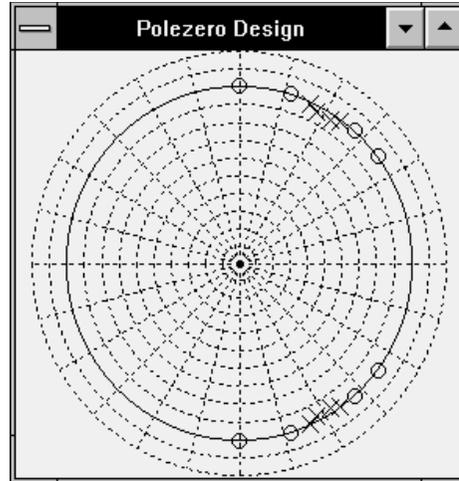
Pole/Zero Design

This section will demonstrate how to interactively move the poles and zeros of an IIR design. For convenience, we will design a standardBandpass filter and then copy these poles and zeros and move them to change the design.

☞   Under the**Design** menu, select*IIR design*.

☞   Select *Load Filter Specifications*under the **File** menu.

☞   Select the*Bandpass* filter and click on*Start* to initiate the design.

☞   Click *OK* on the filter order screen

The plot for thebandpass filter will display at this point

☞   Under the**Design** menu, select*Pole/Zero*Design

☞   The panels displayed on this page will then appear

☞   Select Polar on the Pole/Zero Toolbox to display the Pole/Zero diagram in polar coordinates

☞   Click the Copy button on the toolbox to copy the bandpass design onto the Pole/Zero diagram

☞   The operation mode is currently in "move" mode. A Pole or zero can be selected by placing the cursor over the desired Pole or Zero and pushing the left button down, hold the left button down to move, upon releasing the left button, the responses will be automatically recomputed.

☞   To zoom the Pole/Zero diagram about a specified point, place the cursor at the desired point and click the right button.

What Next?

The *QEDesign* demonstration software does allow you to experiment for yourself. Some suggestions:

➢   Try loading some of the provided filter specification files

➢   Analyze some of the provided system transfer function files

➢   Experiment withquantization

DSPworks

Digital Signal Processing Capabilities

DSPworks is a general purpose signal processing system. Signals can be acquired via a DSP board with external I/O capabilities or generated from one of the built-in signal generators. All operations or commands work on the entire signal. Thus, the *Multiply* command of two signals multiplies corresponding samples in the first file to the second file results in a new output file. Note that *DSPworks* can both import and export Microsoft .WAV files. An extensive variety of operations or commands are available. Script files are available for repetitive operations and a file import/export capability can be used to facilitate interfacing with other systems. All commands use a typical Windows interface with pull down menus and popup dialog boxes. Menus and commands have been organized for very intuitive use.

The following are four guided examples demonstrating some of the capabilities of *DSPworks*. These examples are also automated in script files so you can have a quick look without manually following all the steps described below. The four script files are : 'signals.scr', 'correlat.scr', 'filter.scr', and 'ratechg.scr'.

Installing the demonstration:

Please refer to the introductory section under *QEDesign*.

Starting the DSPworks demonstration:

Double-click on the *DSPworks* icon to start the demo. The *DSPworks* window will open, with menu titles at the top and a blank workspace.

Basic principles for use

DSPworks works as a standard Windows application. These instructions assume some familiarity with Windows operation, but guide you through the operations step by step. If you are a complete newcomer to Windows programs you may benefit from some practice with Windows before using the demonstration.

Main Menu bar and pull-down menus

File	Standard file operations including, Record/Play Script, Import/Export Files, Print and Exit functions and information on *DSPwork*s
Edit	Wave form graphical editing functions
Generator	Wave form generation for standard wave forms
Operation	Mathematical operations on time domain wave frame, graphical and real time displays
DSP	Additional DSP operations on time domain & Frequency domain
Display	Wave form displays & Real-time displays
Acquire	Real-time functions, record signal to disk, playback signal from disk and selection of hardware accelerator functions
Utilities	System utility functions such as file format conversion, number type conversion, graph control
Window	Selection of Windows for display and order of windows, Font & Color Selection

Signal Generation

This example corresponds to the script file 'signals.scr'.

Why are we doing this?

This example demonstrates how to use the built-in generators to create various signals and combine several signals to form a composite signal. In this example we are going to generate several signals, add and multiply two signals, and join two signals to form longer signals.

Generate a high frequency sinusoidal signal 'sin60.tim'.

☞   Click on the **Generator** menu to select a generator.

☞   Choose *Sinusoidal* for sinusoidal wave generator.

➢   A sinusoidal wave generator data entry dialog box appears. These entries specify various characteristic of the wave form to be generated. Set these entries as shown on the right. For 'output file name' field click on the button, and a standard file save dialog box will pop up. Enter 'sin60' and click *'Save'*; '.tim' may be omitted and will automatically be added. Use this method to enter all file names that appear on a button for all subsequent dialog boxes.

☞   Go ahead to generate the wave form by clicking on the *OK* button.

An aside

To add Random Noise to signals, select *Random Noise* type for one of the available Noise distributions.

Results of all generators and operations are automatically displayed at the completion of the generations or operations. Colors and fonts of various graph components are user selectable using the *'Graph Colors...'* menu item under **Window**.

Generating Sinusoidal Wave

Signal frequency	60	Hertz
Sampling rate	600	
Number of samples	600	
Output file name	sin60.tim	
Angular phase delay	0	Degrees
Zero to peak amplitude	0.25	
D. C. offset	0	
Output file format	Binary	
Output number type	32-bit floating	
Random noise type	No noise	

Cancel   OK

Generate a medium frequency triangular signal 'tri6a.tim':

☞   Click on the **Generator** menu to select a generator.

☞   Choose *Triangular* for triangular wave generator.

☞   Set the entries in the dialog box as shown on the right.

☞   Click the OK button to generate the wave form.

Generating Triangular Wave

Signal frequency	6	Hertz
Sampling rate	600	
Number of samples	600	
Output file name	tri6a.tim	
Angular phase delay	0	Degrees
Zero to peak amplitude	0.5	
D. C. offset	0	
Output file format	Binary	
Output number type	32-bit floating	
Random noise type	No noise	

Cancel   OK

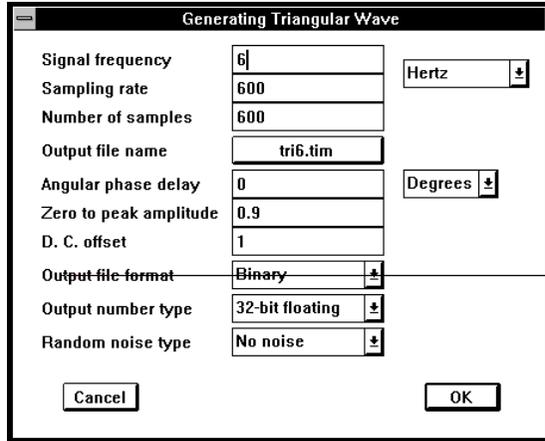## Generate an offset medium frequency triangular signal 'tri6.tim'

☞ Click on the **Generator** menu to select a *Generator.*

☞ Choose *Triangular* for triangular wave generator.

☞ Set the entries in the dialog box as shown on the right.

☞ Click the *OK* button to generate the wave form.

An Aside

'Tri.tim' has an amplitude of 0.9 and an offset of 1.0, i.e. its average value is 1.0 and minimum vale is 0.1. This wave form will be used as the offset signal to amplitude modulate the high frequency sinusoidal wave.

Generating Triangular Wave		
Signal frequency	6	Hertz ±
Sampling rate	600	
Number of samples	600	
Output file name	tri6.tim	
Angular phase delay	0	Degrees ±
Zero to peak amplitude	0.9	
D. C. offset	1	
Output file format	Binary ±	
Output number type	32-bit floating ±	
Random noise type	No noise ±	
Cancel		OK

Multiply 'sin60.tim' by 'tri6.tim'

☞ Click on the **Operation** menu and select *Arithmetic* menu item.

☞ Choose *Multiply two signals*

☞ Set the entries in the dialog box as shown on the right.

☞ Click the *OK* button to do the multiplication.

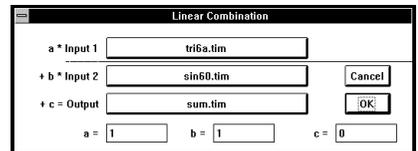| Multiply Two Signals	
a * Input 1	sin60.tim
* Input 2	tri6.tim	Cancel
= Output file	am.tim	OK
a =	1
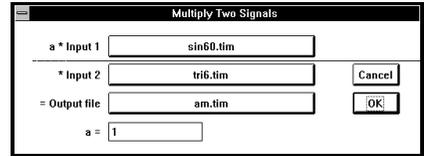
Result

This produces a high frequency wave form not of a constant amplitude but varying like a triangular wave. This is amplitude modulation (A.M.)

More information

This operation multiplies the first sample in 'sin60.tim' with the first sample in 'tri6.tim' and the constant 'a' (1.0 in this case) and store the product as the first sample of the output file. This is repeated for the second and all subsequent samples. Most other operations in *DSPworks* work on sample by sample in a similar fashion.

Add 'tri6a.tim' to 'sin60.tim'

☞ Click on the **Operation** menu and select the Arithmetic menu item.

☞ Choose *Linear Combo...'*

☞ Set the entries in the dialog box as shown on the right.

Linear Combination			
a * Input 1	tri6a.tim		
+ b * Input 2	sin60.tim	Cancel	
+ c = Output	sum.tim	OK	
a = 1	b = 1	c = 0	

More information.

This operation implements the following equation on sample by sample basis:

$$Y = a * X1 + b * X2 + c$$

where Y is the output file, X1 and X2 are two input files, and a, b, and c are arbitrary constants and b may be zero, in which case X2 is not required.

This operation can be used to perform several specialized operations such as offset existing signal (a = 1.0, b = 0.0, c = offset,) inversion (a = -1.0, b = 0.0, c = 0.0), amplify/attenuate (a = gain, b = 0.0, c = 0.0,) etc.

➢ Click the *OK* button to perform the multiplication.

Generate a low frequency sinusoidal signal 'sin1.tim'

➢ Click on the **Generator** menu to select a generator.

➢ Choose *Sinusoidal* for sinusoidal wave generator.

➢ Set the entries in the dialog box as shown.

➢ Click the OK button to generate the waveform.

Generating Sinusoidal Wave		
Signal frequency	1	Hertz
Sampling rate	600	
Number of samples	600	
Output file name	sin1.tim	
Angular phase delay	0	Degree
Zero to peak amplitude	1	
D. C. offset	0	
Output file format	Binary	
Output number type	32-bit floating	
Random noise type	No noise	

Cancel   OK

Generate a low frequency square signal 'sqr1.tim'

➢ Click on the **Generator** menu to select a generator.

➢ Choose *Triangular* for triangular wave generator.

➢ Set the entries in the dialog box as shown.

➢ Click the *OK* button to generate the waveform.

Generating Square Wave		
Signal frequency	1	Hertz
Sampling rate	600	
Number of samples	600	
Output file name	sqr1.tim	
Angular phase delay	0	Degrees
Zero to peak amplitude	1	
D. C. offset	0	
Output file format	Binary	
Output number type	32-bit floating	
Random noise type	No noise	

Cancel   OK

Join (append) 'sqr1.tim' to 'sin1.tim'

➢ Click on the **Operation** menu and select *Join*.

➢ Set the entries in the dialog box as shown on the right.

➢ Click the *OK* button to join the wave forms.

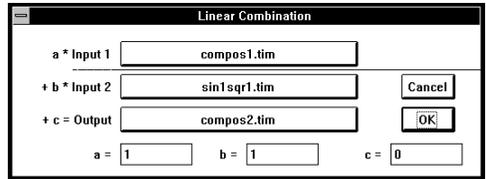| Join Two Signals		
Input 1 file	sin1.tim	
& Input 2 file	sqr1.tim	Cancel
= Output file	sin1sqr1.tim	OK

An Aside

The preceding method of joining two signals is primarily meant for scripting. In addition, the two original signals are not modified; the output is written into a new file. The join operation can also be done graphically by means of the graphical cut/copy and paste editing capability. However, graphical operations cannot be put into script file and they also directly modify signal files by overwriting the originals.

Join (append) 'sum.tim' to 'am.tim'

➢ Click on the **Operation** menu and select *Join*

➢ Set the entries in the dialog box as shown on the right.

➢ Click the *OK* button to join the wave forms.

Join Two Signals		
Input 1 file	am.tim	
& Input 2 file	sum.tim	Cancel
= Output file	compos1.tim	OK

Add 'sin1sqr1.tim' to 'compos1.tim' to make the final composite signal

➢   Click on the **Operation** menu and select *Arithmetic* menu item.

➢   Choose *'Linear Combo..'*

➢   Set the entries in the dialog box as shown on the right.

➢   Click the *OK* button to add the wave forms.

	Linear Combination	
a * Input 1	compos1.tim	
+ b * Input 2	sin1sqr1.tim	Cancel
+ c = Output	compos2.tim	OK
a = 1	b = 1	c = 0

An Aside

Viewing signals with too many sample points can be a problem.

The graphical display can display a maximum of 600 samples at once. For signals with more than 600 samples only the first 600 samples are displayed when the graph first appear. However, two methods are available to display other sample points.
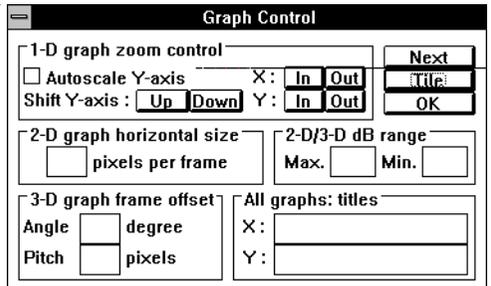
Some Useful Tools

Scroll bar

Graphs of signals with more than 600 sample points will display only 600 samples in one screen and will have a horizontal scroll bar at the bottom of the graph window. Click the right scroll arrow to pan the display window right 1/5 screen full; clicking on the scroll bar between the right scroll arrow and the scroll box will advance the display window right one screen full. The left scroll arrow and the scroll bar left of the scroll box pans to the left.

Zoom control

All graphs are initially displayed at 1:1 ratio, i.e. every sample is displayed. It is possible to change the zoom ratio to display the value ranges of more samples. Samples are grouped into clusters and the maximum and the minimum values of each cluster are displayed. In this way a good idea about the signal values distribution can be given. The zoom controls are on the 'Graph Control' dialog box. The dialog box can be invoked by selecting the 'Graph Control' menu item under the 'Window' menu. The 'Graph Control' dialog box is shown below:

The lower half of the dialog box is hidden off screen. They relate to more advanced graphical control functions and will not be discussed here. The zoom control of interest right now are the X axis zoom in and zoom out control buttons located to the left of 'Tile' button. Make sure the 'compos2.tim' is currently selected:

Graph Control

1-D graph zoom control
☐ Autoscale Y-axis   X : In Out
Shift Y-axis : Up Down   Y : In Out

Next
Tile
OK

2-D graph horizontal size   2-D/3-D dB range
pixels per frame   Max.   Min.

3-D graph frame offset   All graphs: titles
Angle   degree   X :
Pitch   pixels   Y :

➢   Click in the graph 'compos2.tim' to bring it to the top and to select it.

The 'Tracking cursor co-ordinate' dialog box may appear.

➢   Click the X zoom out button (the top right one immediately to the left of the 'Tile' button)

Tracking cursor co-ordinate			
Ch: 0	X: 498.33	Y: 0.0105	1:1.00
delta:	x: -500.00	y: 0.0209	f:2.000Hz
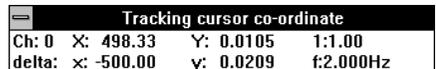
The graph will zoom out to show the entire signal. The horizontal scroll bar is removed to indicate that the entire signal is shown.

Second Example

This corresponds to the script file 'filter.scr'.

Why are we doing this?

This example demonstrates the filtering capabilities of *DSPworks*. First, a square wave is filtered by an FIR filter and an IIR filter and the results contrasted. Then the impulse responses of an FIR filter and an IIR filter are obtained and the frequency responses of both filters are computed using FFT. And lastly the square wave is convolved with the FIR impulse response and the IIR impulse response.

☞   Close all graph windows from previous exercise so that the limit of 36 opened graph windows will not be exceeded.

This time instead of stepping through all the steps, we will execute the script file:

➢   Click on the **File** menu and select *'Play Script...'* menu item.

➢   Choose *'filter.scr'* and click the *'Open'* button to start the design.

➢   Whenever the Info dialog box appears, click the *'OK'* button to continue. If during Info dialog box display *DSPworks* is brought to the front it will obscure the Info dialog box and *DSPworks* will patiently wait for the 'OK' button in the Info dialog box to be clicked. When this happens, do the following:

   Double click on the desktop, the 'Task List' dialog box will appear.

   Double click on 'INFO' task, the Info dialog box will appear.

   Click *'OK'* button to continue.

Third Example

This correspond to the script file 'correlat.scr'.

Why are we doing this?

This example demonstrates the auto-correlation and cross-correlation functions of *DSPworks*.

➢   Close all graph windows from previous exercise

➢   We will again execute the script file: Click on the **File** menu and select 'Play Script...' menu item.

➢   Choose 'correlat.scr' and click the 'Open' button to start.

➢   Whenever Info dialog box appears, click the 'OK' button to continue.

Fourth Example

This corresponds to the script file 'ratechg.scr'.

Why are we doing this?

This example demonstrates how *DSPworks* can be used to change the sampling rate of a signal.

An aside

Normally, if a signal sampled at 8 kHz sampling rate were to be played back at 10 kHz sampling rate, there will be a shift in signal frequency corresponding to the ratio of the recording sampling rate to the playback sampling rate. Sampling rate conversion technique uses Digital Signal Processing theories to change the sampling rate without causing a shift in signal frequency.

➢   Close all graph windows from previous exercise.

➢   We will again execute the script file: Click on the **File** menu and select *'Play Script..'* menu item.

➢   Choose *'ratechg.scr'* and click the *'Open'* button to start.

➢   Whenever Info dialog box appears, click the *'OK'* button to continue.

This software is specially designed to let you make a proper evaluation of QEDesign and DSPworks.

It is concise and informative. It will take you about five minutes to get a good 'feel' for the software, but you can take as long as you like. Find out why most of our business comes from repeat customers.

If you have any questions about our software, please call and we'll be glad to take the time to discuss your requirements with you.

Also available - our Advanced Series of QEDesign systems including:

Multirate Signal Processing

Sample Rate Conversion System

Quadrature Mirror Filter Banks