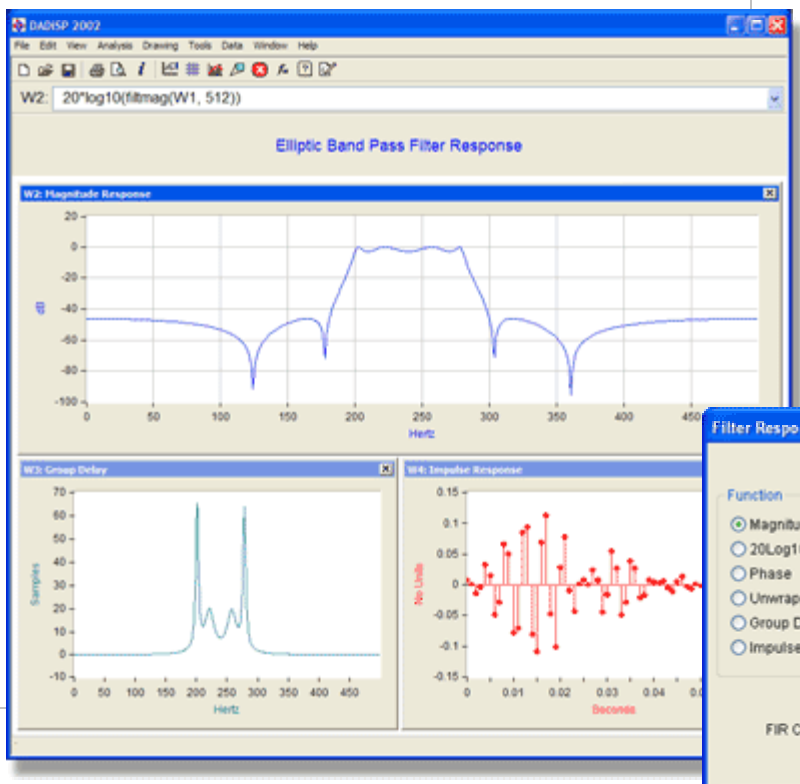
DADiSP / Filters

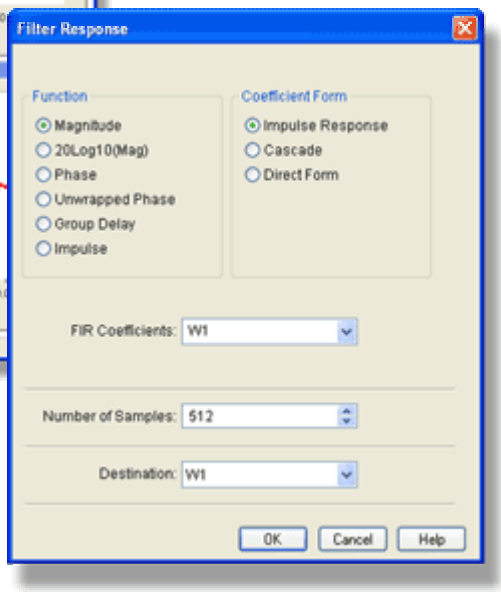
Digital Filter Design Module

DADiSP/Filters is a menu-driven module for FIR and IIR digital filter design and analysis. From any DADiSP Worksheet, you can quickly design, view and analyze both FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filters. Once you have designed a filter, you may filter the raw data, then view and analyze the filtered signal. Through the easy-to-use dialog boxes or simple one line functions, you can tune the filter iteratively, re-filtering the data until you have separated the signal from the noise cleanly.

KEY FEATURES

- Simple User Interface
- Lowpass, Highpass, Bandpass, Bandstop and Multiband Filters
- Finite Impulse Response (FIR) Filter Design
- FIR Hilbert Transforms and Differentiators
- FIR Remez Exchange and Kaiser Window Design Algorithms
- Infinte Impulse Response (IIR) Filter Design
- IIR Bessel, Butterworth, Chebychev I, Chebychev II and Elliptic Filters
- IIR Bilinear Transform and Matched Z Design Algorithms
- Magnitude, Phase, Group Delay and Impulse Response
- Output Coefficent Form Conversion





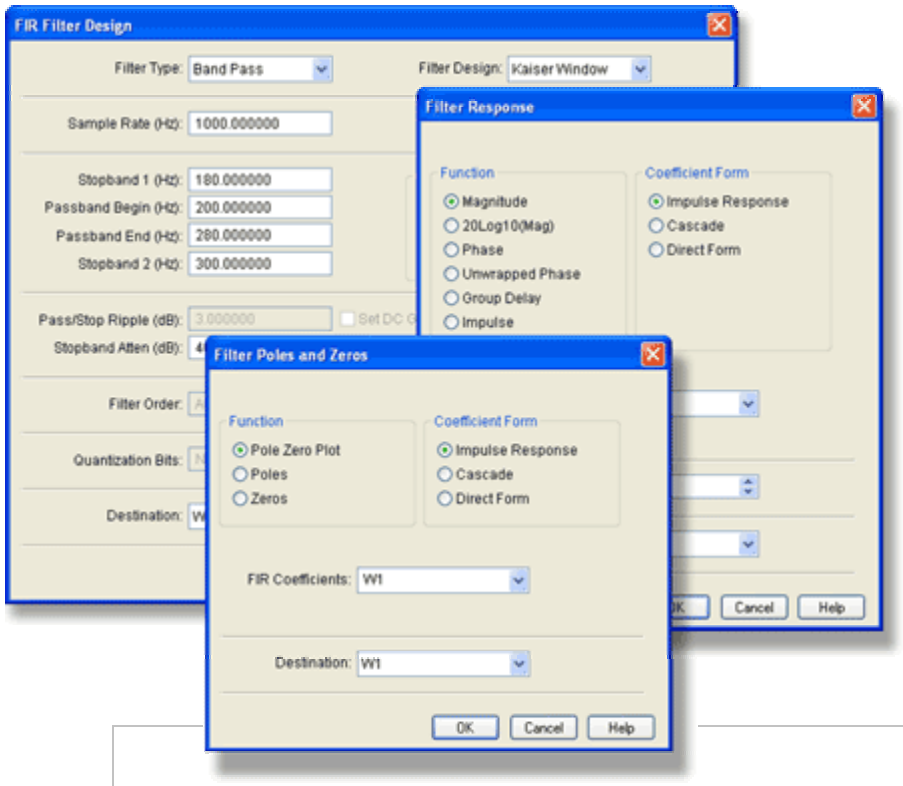
- Output Coefficent Quantization
- Time and Frequency Domain Filtering Functions
- Pole-Zero Plots

New Features

DADiSP/Filters Version 5.0 includes a completely redesigned user interface to streamline the process of designing and applying digital filters. Straightforward dialog boxes with automatic option validation simplifies both the design and analysis of filters.

Filter coefficients can be easily converted to various filter structures and quantization routines are included to help simulate DSP chipsets.

IIR Bessel filters and the Matched Z Transform design method have been added. Linear phase FIR Kaiser filters have been expanded and enhanced. Both time and frequency domain filtering routines have been optimized to provide more efficient filter processing.



Digital Filter Design Module

DADiSP/Filters is a menu-driven, digital filtering module that adds complete FIR and IIR filtering capabilities to DADiSP. DADiSP/Filters allows you to quickly design, analyze and process both FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) digital filters from easy-to-use dialog boxes or simple one line functions. On-line help and examples are also provided.

FILTERS 5.0 NEW FEATURES SUMMARY

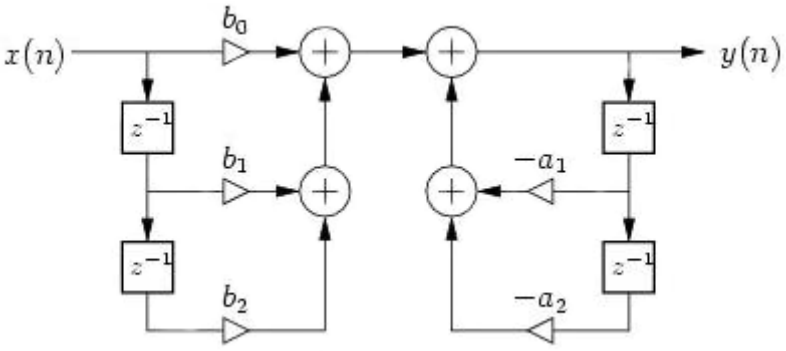
- Streamlined Interface
- IIR Bessel Filters
- IIR Matched Z Design Algorithm
- Improved FIR Kaiser Window Filters
- Coefficient Conversion and Quantization
- Optimized Filter Processing Functions

Design, Analyze and Apply

DADiSP/Filters gives you the power to easily build digital filters that emulate hardware based designs for testing and verification or perform filtering operations not possible with traditional analog methods. DADiSP/Filters allows you to remove noise generated during the data collection process. DADiSP/Filters is the perfect complement to GPIBLab, DADiSP's acquisition module for collecting data from IEEE-488 based instruments and DADiSP/AdvDSP, a sophisticated module for advanced signal processing techniques.

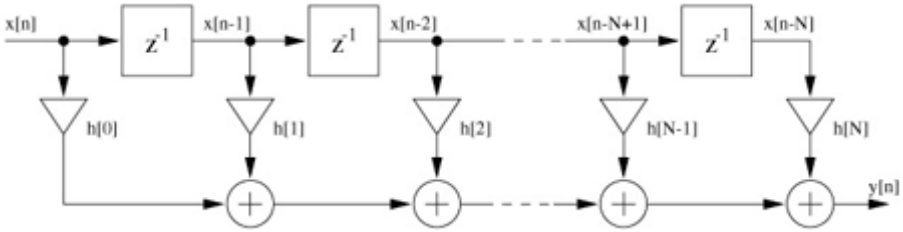
Coefficient Conversion

Filter coefficients can be converted to and from Cascade, Direct and FIR form and the coefficients can be quantized to emulate DSP chipsets.



FIR Filters

The FIR module creates linear phase lowpass, highpass, bandpass, bandstop, multiband, Hilbert transformers and differentiators using the Parks-McClellan/Remez Exchange optimal design algorithm. The filter order can be specified or automatically estimated from the particular filter specifications. The Kaiser Window method is also provided, capable of creating both very high order FIR filters and extremely tight, narrowband filters.

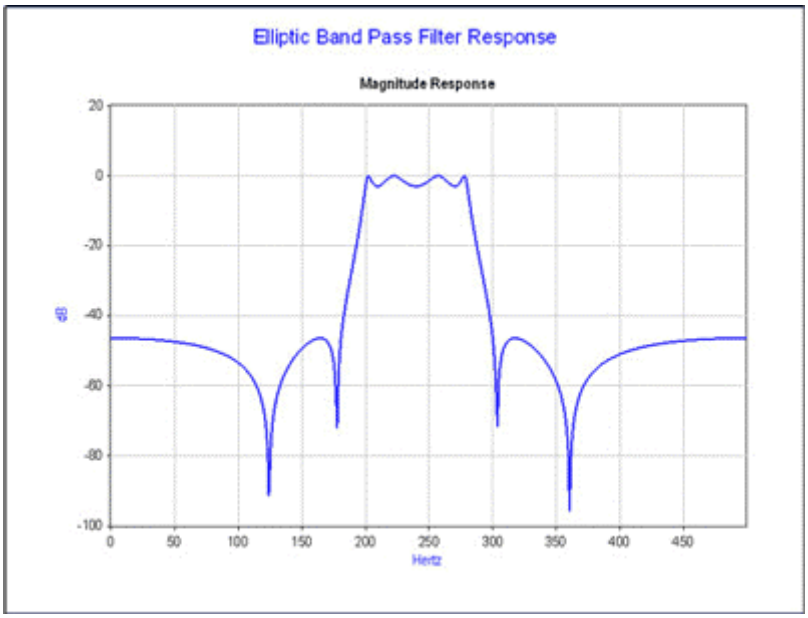


IIR Filters

The IIR module supports Bessel, Butterworth, Chebychev I, Chebychev II and Elliptic designs for lowpass, highpass, bandpass and bandstop recursive filters. The Bilinear transform method is employed and the resulting IIR coefficients can be determined in optimal multistage cascade format or traditional Direct form. The module also supports the Matched-Z transform method to approximate linear phase characteristics for IIR Bessel filters.

Filters Response

The impulse, magnitude, phase and group delay characteristics can be calculated for any filter. Pole-Zero plots are also supported. Efficent time and frequency domain filtering algorithms are provided to apply the filter to any series.



Fully Integrated

The filters module is fully integrated with DADiSP to provide a complete digital filter design, analysis, display and processing environment. The DADiSP/Filters user interface is dialog based, eliminating the need to memorize argument lists or formulas and allows quick recall of previous filter designs. The filter coefficients are automatically displayed in a DADiSP window and can be saved for further use by DADiSP or other programs.

Filter Functions

Although most users access DADiSP/Filters through the dialog based interface, DADiSP/Filters includes over standalone 50 functions. The following table is a summary of each function.

* Indicates new or improved Version 5.0 functions.

FIR Filters Functions

bandpass	Designs a FIR linear phase bandpass filter
bandstop	Designs a FIR linear phase bandstop filter
diff	Designs a FIR differentiator
fastfilter	FFT based FIR filtering
highpass	Designs a FIR linear phase highpass filter
hilbert	Designs a FIR Hilbert transformer
kwbpass*	Designs a Kaiser window FIR bandpass filter
kwbstop*	Designs a Kaiser window FIR bandstop filter
kwhpass*	Designs a Kaiser window FIR highpass filter
kwlpass*	Designs a Kaiser window FIR lowpass filter
lowpass	Designs a FIR linear phase lowpass filter
remez	Creates multiband FIR linear phase filters

Filter Coefficient Conversion Functions

cas2dir*	Converts Cascade form to Direct form
dir2cas*	Converts Direct form to Cascade form
fir2dir*	Converts FIR impulse form to Direct form
fir2cas*	Converts FIR impulse form to Cascade form

IIR Filters Functions

bessel*	Designs an IIR Bessel filter
butterworth	Designs an IIR Butterworth filter
cascade	Filters a time domain input with an IIR filter
cheby1	Designs an IIR Chebychev I filter
cheby2	Designs an IIR Chebychev II filter
elliptic	Designs an IIR Elliptical filter

Filter Response Functions

filtgrpdelay*	Calculates group delay of any filter
filtmag*	Calculates any filter magnitude response
filtimp*	Calculates any filter impulse response
filtphase*	Calculates any filter phase response
firmag	Calculates FIR filter magnitude response
firphase	Calculates FIR filter phase response
iirimp*	Calculates IIR filter impulse response
iirmag*	Calculates IIR filter magnitude response
iirphase*	Calculates IIR filter phase response
filtgrpdelay*	Calculates group delay of any filter
filtmag*	Calculates any filter magnitude response
filtimp*	Calculates any filter impulse response

Filtering Functions

dirfilter*	Apply Direct form filter in the time domain
dirfilterF*	Apply Direct form filter in the frequency domain
filtdataF*	Apply any filter in the frequency domain
filtdata*	Apply any filter in the time domain
firfilterF*	Apply FIR filter in the frequency domain
firfilter	Apply FIR filter in the time domain
iirfilterF*	Apply IIR filter in the frequency domain
iirfilter	Apply IIR filter in the time domain
dirfilter*	Apply Direct form filter in the time domain
dirfilterF*	Apply Direct form filter in the frequency domain
filtdataF*	Apply any filter in the frequency domain
filtdata*	Apply any filter in the time domain

Misc Filtering Functions

filtzeros*	Calculates zeros of any filter
filtpoles*	Calculates poles of any filter
fir	Evaluates a FIR difference equation
firpz	Creates an FIR filter zero plot
firzeros	Calculates zeros of an FIR filter
fullfir	Converts FIR filter to full band linear phase
iir	Evaluates an IIR difference equation
iirgrpdelay*	Calculates group delay of an IIR filter
iirpoles	Calculates poles of an IIR filter
iirpz	Creates an IIR filter pole-zero plot
iirzeros	Calculates zeros of an IIR filter
polecoef	Converts IIR biquad to direct pole coeff form
quantize*	Quantize filter coefficients to N bits
zerocoef	Converts IIR biquad to direct zero coeff form
unwrap	Phase unwrapping using Schafer's algorithm

DADiSP / AdvDSP

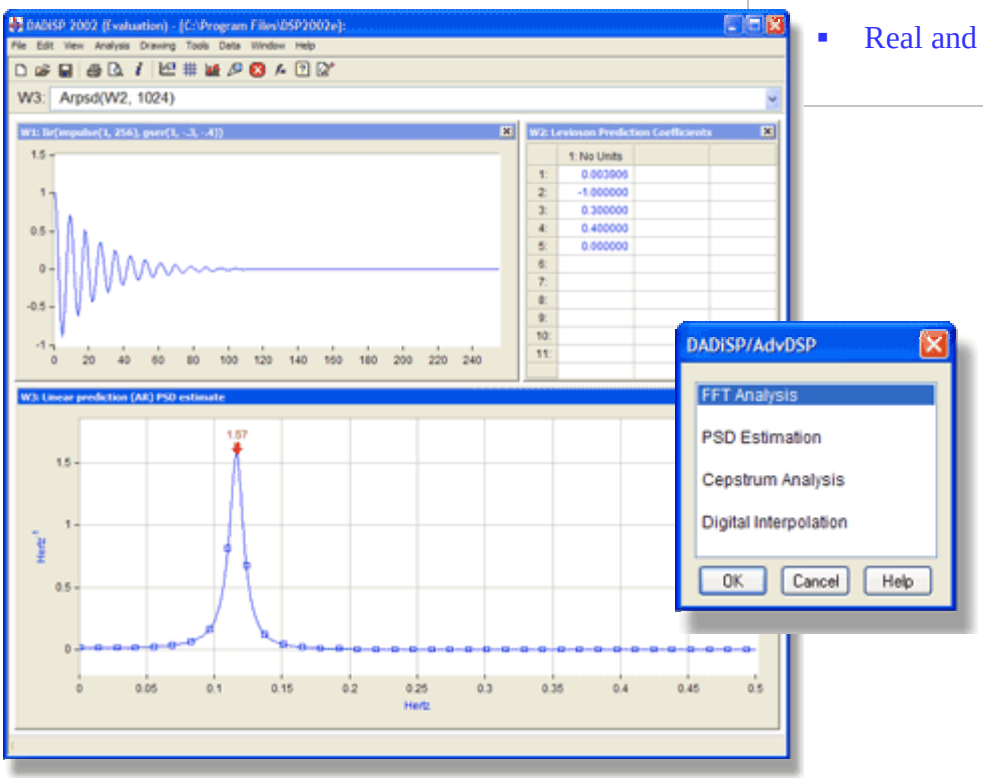
Advanced DSP Module

DADiSP/AdvDSP is a menu-driven module for DADiSP that offers a wide variety of DSP algorithms, including advanced FFT analysis, power spectral density estimation, digital interpolation and cepstrum analysis.

AdvDSP is easy-to-use and allows you to perform sophisticated signal processing without any programming. Each routine is available through a simple dialog box interface or as a direct command line function. Extensive on-line help and examples are also provided.

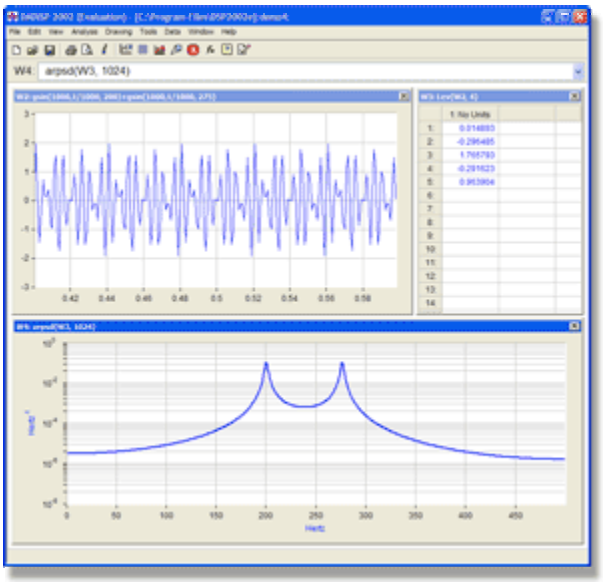
KEY FEATURES

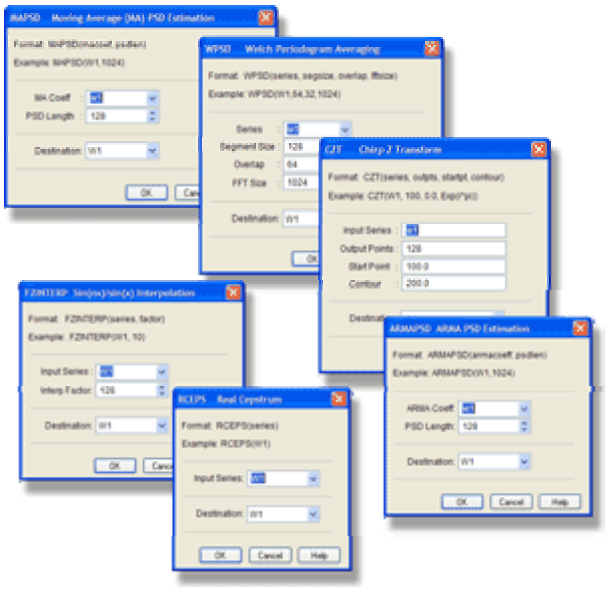
- Simple User Interface
- Classical PSD Estimation
- Parametric AR, MA and ARMA PSD Estimation
- Parametric Linear Prediction
- Zoom FFT and Chirp Z Transforms
- Sinx/x and Zero Insertion Digital Interpolation
- Transfer Function, Cross Power and Coherence Estimate
- Real and Complex Cepstrum Analysis



Advanced DSP Module

DADiSP/AdvDSP is a menu-driven, advanced signal processing module that adds classical and parametric PSD estimation, linear prediction, Zoom FFT, Chirp Z transform, digital interpolation and cepstrum routines. Each routine is available through an easy to use dialog box interface or simple command line functions. On-line help and examples are also provided.





Classical PSD Estimation

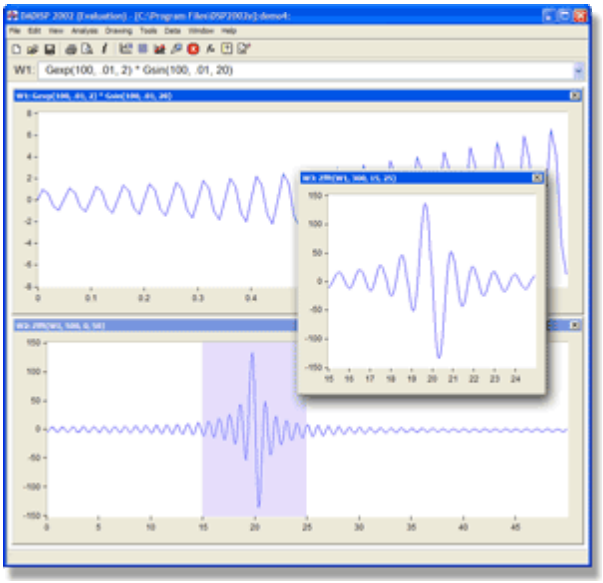
Classical Power Spectrum Estimation techniques include the standard FFT based PSD, the correlation method and the overlapped Welch method of periodogram averaging. Both the segment length and overlap size can be specified. In addition, coherence, cross-power and transfer function estimates are supported.

Parametric PSD Estimation

Parametric estimation techniques include AR - Autoregressive all pole models, MA - Moving Average all zero models, and ARMA, generalized pole and zero models. The model coefficients are computed from measured data and the coefficients can be used to estimate the PSD as well as provide time domain linear prediction results.

Digital Interpolation

The effective sample rate of any waveform can be changed using a variety of robust digital interpolation algorithms. Traditional Sinx/x bandlimited interpolation as well as non-integer multiple zero insertion routines are provided. Classical linear and cubic spline interpolation functions are also included.



Advanced FFT Analysis

AdvDSP can compute a true N point FFT. If the series length is less than the specified FFT size, the series is automatically zero padded. If the FFT size is less than the series length, rather than truncation, the series is time aliased resulting in a properly decimated FFT. The Zoom FFT computes the FFT for a selected frequency range and the Chirp Z transform computes the FFT about a generalized Z plane contour.

Cepstrum Analysis

Both real and complex cepstrum computation is available. Cepstrum analysis simplifies the task of echo identification and cancellation for speech and audio applications.

Advanced DSP Functions

Although most users access DADiSP/AdvDSP through the dialog based interface, DADiSP/AdvDSP includes over 20 standalone functions. The following table is a summary of each function.

Classical PSD Estimation Functions

apsd	Correlation method of PSD estimation
fftpsd	Standard FFT based PSD estimation
wpsd	Welch method of periodogram averaging

Misc Classical PSD Functions

wpxx	wpsd over entire FFT range
wpxy	Cross power spectrum estimate
wtxy	Transfer function estimate
wcoh	Coherence function estimate
wpsdseg	wpsd method with specified number of segments

AR Parametric PSD Estimation Functions

lev	Yule-Walker linear prediction using Levinson recursion
mem	Burg (maximum entropy) method of PSD estimation
covar	Covariance method of linear prediction
arpsd	Compute PSD estimate based on AR coefficients
arpredict	Linear prediction based on AR coefficients

MA Parametric PSD Estimation Functions

ma	Moving average method of PSD estimation
mapsd	Compute PSD estimate based on MA coefficients

ARMA Parametric PSD Estimation Functions

lsmyw	Least Squares Modified Yule-Walker ARMA calculation
armapsd	Compute PSD estimate based on ARMA coefficients

Advanced FFT Analysis Functions

czt	Chirp Z transform
nfft	N point FFT calculation
zfft	Zoom FFT

Digital Interpolation Functions

interp	Linear interpolation
spline	Cubic spline interpolation
fsxinterp	sin(x)/x bandlimited interpolation
fzinterp	Zero insertion interpolation

Cepstrum Functions

rceps	Real cepstrum computation
cceps	Complex cepstrum computation
ldunwrap	Cepstrum phase unwrapping

DADiSP / SRS

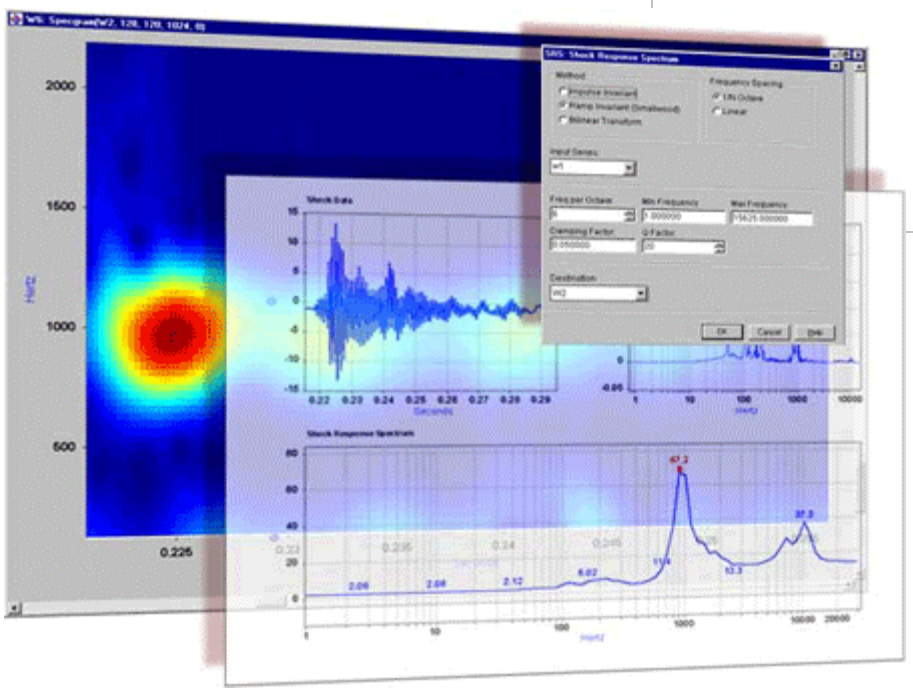
Shock Response Spectrum Module

DADiSP/SRS is a menu driven module designed for the analysis of the Shock Response Spectrum (SRS). SRS Analysis is a useful tool in minimizing the potential damage to a component due to shock. SRS is employed in industries such as aerospace engineering, automotive engineering, Department of Defense and ordnance evaluation.

Given acceleration time history data, the SRS module allows the user to choose from a variety of industry standard analysis methods and select the desired frequency range and spacing. The damping ratio or Q factor is also adjustible.

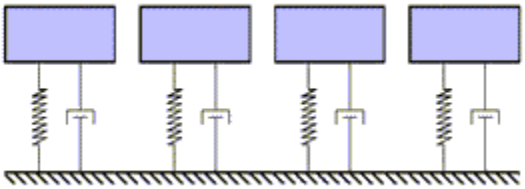
KEY FEATURES

- Ramp Invariant (Smallwood) Step Response Matching Algorithm
- Impulse Invariant Impulse Response Matching Algorithm
- Bilinear Transform Frequency Response Matching Algorithm
- Whole Octave Frequency Spacing
- Fractional (1/N) Octave Frequency Spacing
- Linear Frequency Spacing
- Adjustible Damping Ratio / Q Factor



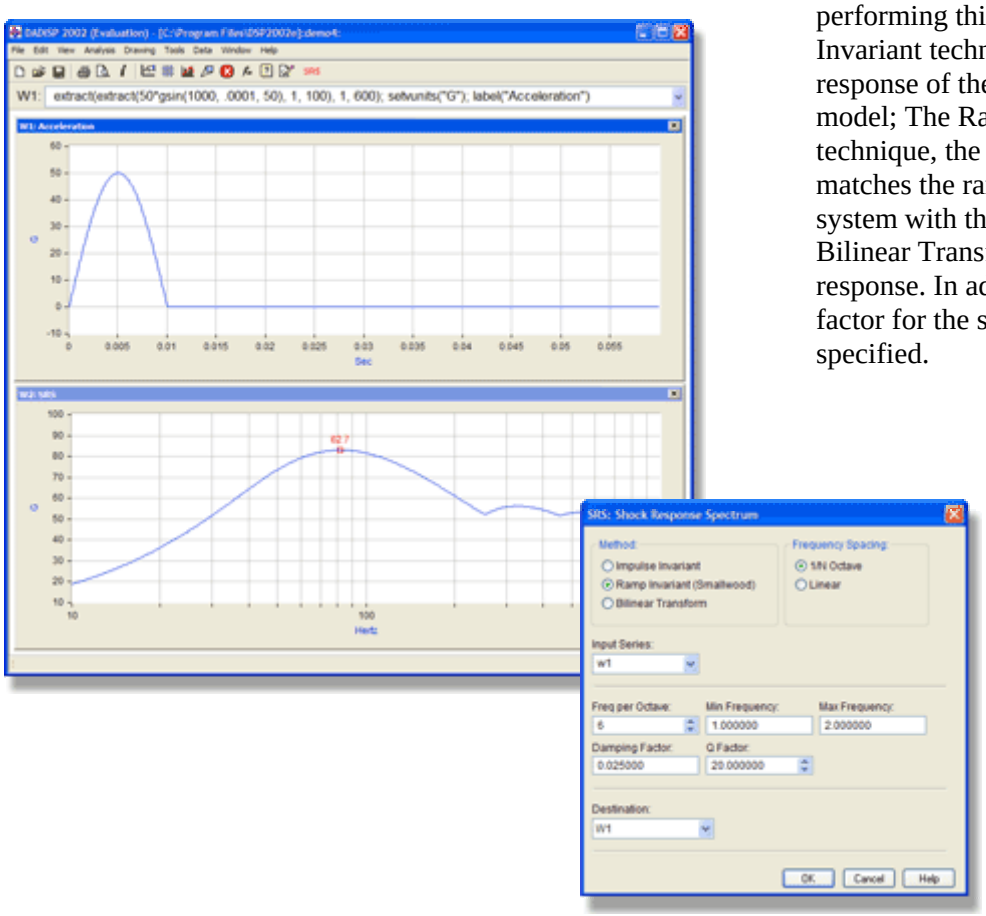
Shock Response Spectrum Module

The Shock Response Spectrum, or SRS, is used in modeling a mechanical component as a series of single degree of freedom (SDOF) spring-dashpot subsystems each with a constant damping ratio and varying natural frequency. Each spring-dashpot subsystem is considered a 2nd order linear system and is converted into the digital domain. The absolute maximum response of each spring-dashpot subsystem is returned as the SRS result for the corresponding natural frequency of the subsystem. A plot of the absolute maximum responses for all the natural frequencies is the Shock Response Spectrum.



Multiple Analysis Methods

To calculate the SRS, each analog 2nd order spring-dashpot subsystem is converted into the digital domain. DADiSP/SRS supports three industry standard methods of performing this transformation: the Impulse Invariant technique matches the impulse response of the analog system with the digital model; The Ramp Invariant (Smallwood) technique, the most common approach, matches the ramp response of the analog system with the digital model; and the Bilinear Transform matches the frequency response. In addition, the damping ratio or Q factor for the spring-dashpot network can be specified.

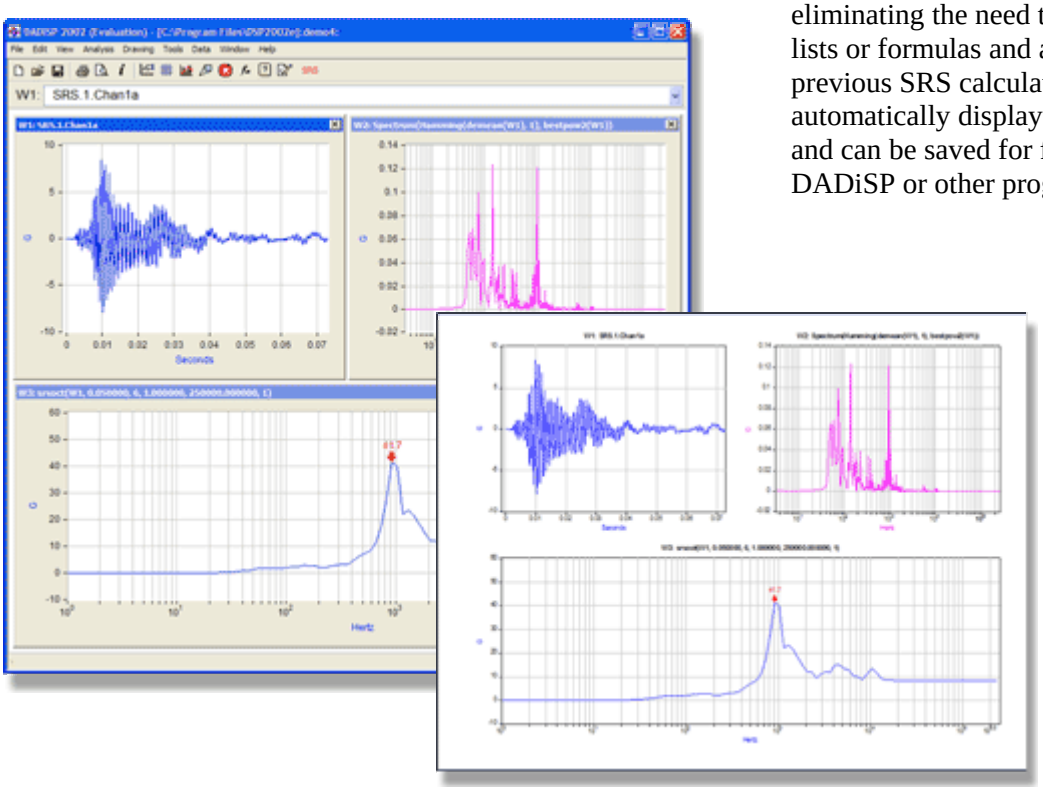


Octave or Linear Natural Frequencies

The natuaral frequency range of interest can be specified in octave, fractional octave or linear bands to produce comprehensive SRS results. Whole octave and 1/N fractional octave bands provide SRS information over a wide natural frequency range with a minimum of computation. Linear frequency ranges are useful to produce high resolution SRS results for a narrow band of frequencies. Once computed, any SRS can be plotted with linear, log or log-log axes.

Fully Integrated

The SRS module is fully integrated with DADiSP to provide a complete shock analysis, display and processing environment. The DADiSP/SRS user interface is dialog based, eliminating the need to memorize argument lists or formulas and allows quick recall of previous SRS calculations. The SRS results are automatically displayed in a DADiSP window and can be saved for further processing by DADiSP or other programs.



SRS Functions

srsoct	SRS calculation for octave spaced frequencies
srslin	SRS calculation for linear spaced frequencies
srscoef	Convert SRS SDOF analog system to digital domain

DADiSP / WAV

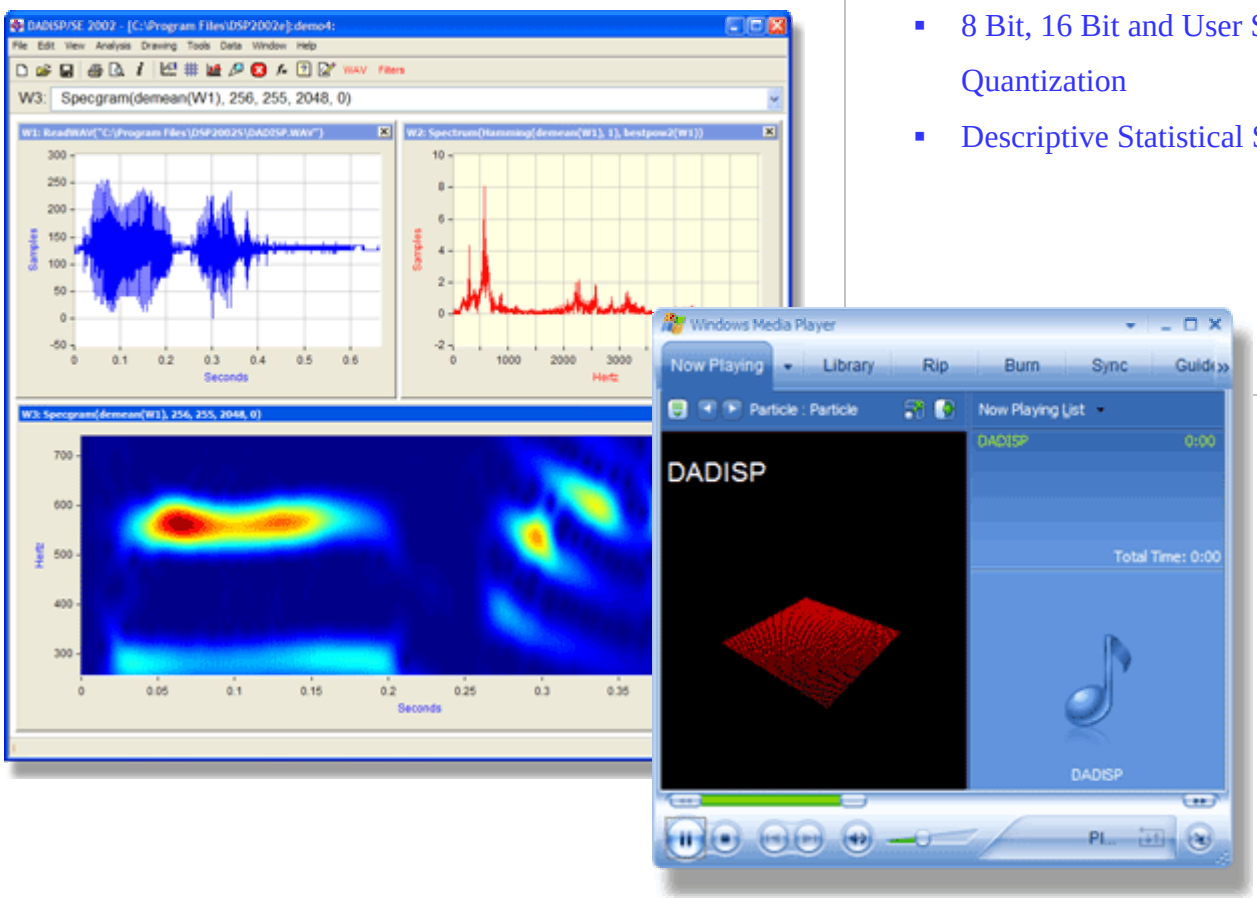
WAV Audio Module

DADiSP/WAV is fully integrated with the DADiSP Worksheet to provide a complete digital audio data analysis, display and processing environment. DADiSP/WAV allows the user to quickly read, write, and edit digital audio data from WAV format files via easy-to-use pop-up menus or simple one line functions.

The DADiSP/WAV module supports standard PCM WAV file format for 8 and 16 bit mono and stereo data files giving users the flexibility to work directly with their data and take full advantage of the WAV file format.

KEY FEATURES

- Simple User Interface
- PCM WAV File Support
- Read, Write and Play any WAV File
- Fully Integrated with Media Player
- Mono and Stereo Support
- 8 Bit, 16 Bit and User Specified Quantization
- Descriptive Statistical Summary



New Features

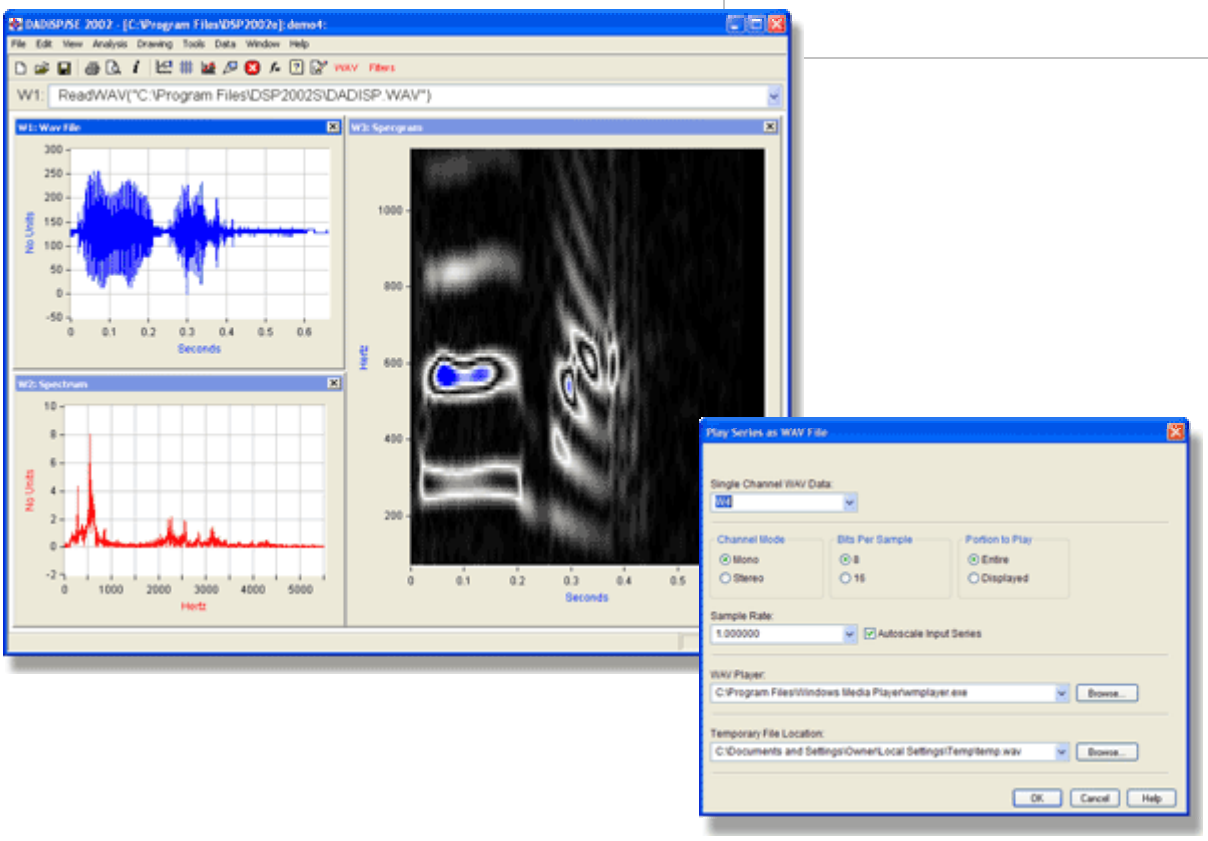
DADiSP/WAV Version 2.0 includes a completely redesigned user interface to streamline the process of creating, reading, writing and playing PCM .WAV files.

Waveforms from any DADiSP Window can be played directly with Windows Media Player or any WAV compatible application. The data is automatically scaled to provide optimal signal to noise ratio.

Series can be quantized to standard 8 bit and 16 bit resolution or user determined quantization and bit format can be specified for custom applications.

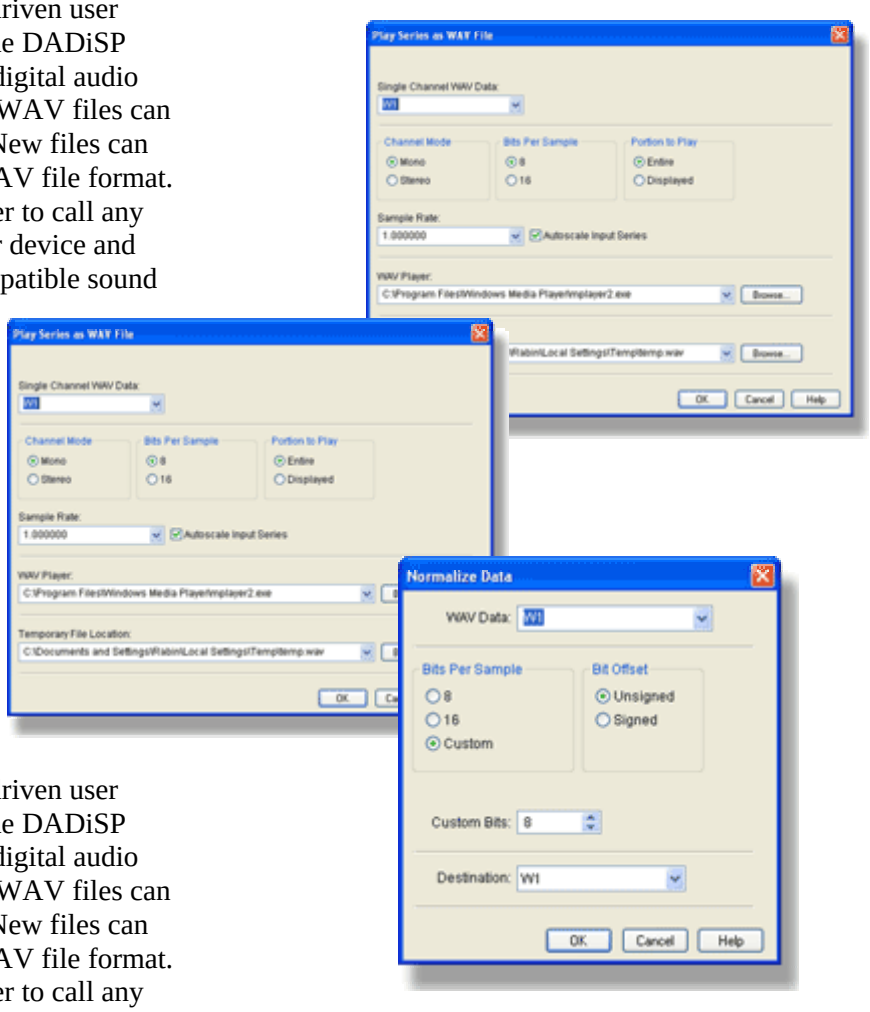
WAV 2.0 NEW FEATURES SUMMARY

- Streamlined Interface
- Direct Integration with Media Player
- Automatic Waveform Scaling
- Standard and User Defined Quantization and Format



WAV Digital Audio Module

DADiSP/WAV provides a menu-driven user interface integrated directly into the DADiSP Worksheet to provide a complete digital audio processing environment. Existing WAV files can be read, processed and modified. New files can be generated and written in the WAV file format. DADiSP/WAV also allows the user to call any Windows compatible WAV player device and play WAV data on any WAV compatible sound card.



Read, Write, Create and Play WAV Data

DADiSP/WAV provides a menu-driven user interface integrated directly into the DADiSP Worksheet to provide a complete digital audio processing environment. Existing WAV files can be read, processed and modified. New files can be generated and written in the WAV file format. DADiSP/WAV also allows the user to call any Windows compatible WAV player device and play WAV data on any WAV compatible sound card.

Clean Interface

Each routine is available through DADiSP's pop-up menus and as a direct command line function. Simple "fill-in-the-blank" menu fields provide a user-friendly interface to help get the job done quickly and efficiently.

Open Source

DADiSP/WAV is written entirely in SPL code, giving quick execution within DADiSP. These routines can be incorporated into user-defined functions and menus to further customize DADiSP to specific applications. All SPL source code, variable definitio
ns and menus are supplied in ASCII text format and can be easily modified to meet the needs of custom applications.

WAV Functions

DADiSP/WAV includes several functions to read, write, play and scale data in the WAV file format.

WAV Functions

readwav	Read a WAV file directly into a Window
writemono	Write a series to a mono channel WAV file
writestereo	Write two series to a stereo channel WAV file
playmono	Play a series in a mono channel WAV format
playstereo	Play two series in stereo channel WAV format
normalize	Normalize a series for N bit WAV file quantization

DADiSP / MAT File

MATLAB File Import Module

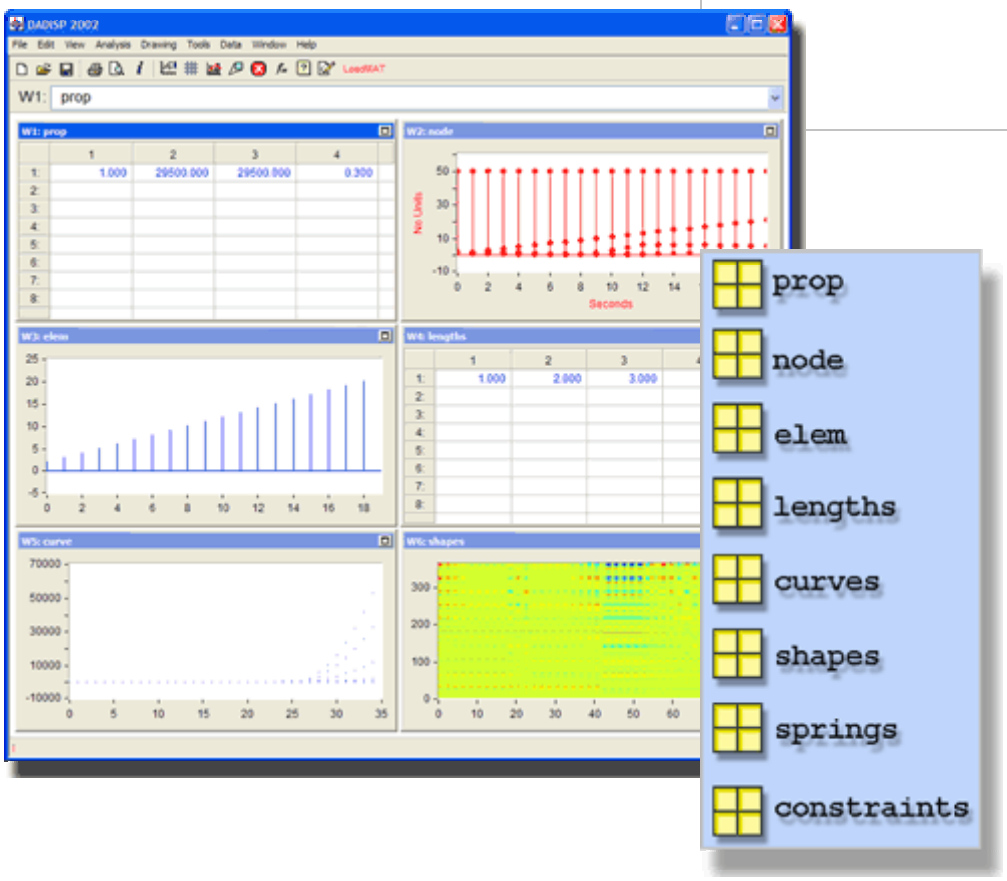
DADiSP/MAT File is a simple dialog based module designed to easily import data files saved in the MAT file format created by MATLAB (1) and similar programs. MAT files Version 4.0 and higher are supported.

MAT files can be of any size and contain any number of variables. The entire contents of the file can be imported or a subset of specific variables can be selected. Array variables are optionally plotted automatically.

(1) MATLAB is a registered trademark of The MathWorks, Inc.

KEY FEATURES

- Simple Dialog Box User Interface
- Loads Version 4.0 MAT Files and Higher
- Handles both Real and Complex Arrays
- 1x1 Arrays Imported as Scalars
- Supports String Variables
- Loads Structures and Cells as Global Variables
- Automatic Plotting of Array Variables
- Does Not Require MATLAB

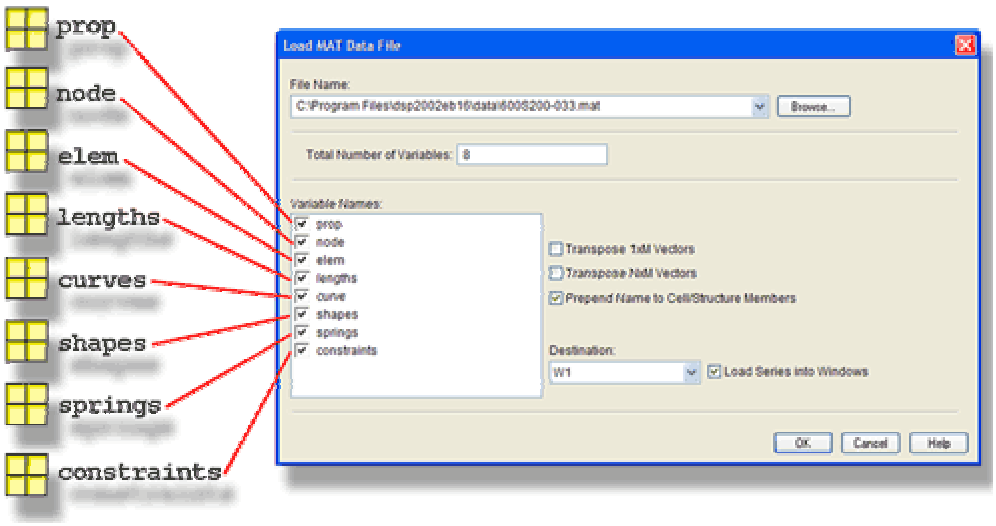


MATLAB MAT File Import Module

A MAT file is a binary data file format created by MATLAB and similar programs. MAT files can contain any number of program variables produced by a MATLAB interactive session or automated script. These variables represent several MATLAB data types including strings, arrays, cells and structures.

Simple Interface

Importing a MAT file is as simple as pressing a button and selecting the file. After a file is chosen, a list of the contained variables is displayed. The entire contents of the file can be imported or a smaller subset of specific variables can be selected. Arrays can be imported "as is" or transposed on the fly to take advantage of DADiSP's optimized column orientation.



MAT File Formats

Although most MAT files are often identified by the .MAT file extension, a variety of differing internal formats exist depending on the version of MATLAB that produced a particular file. Rather than deciphering each format directly, DADiSP/MAT File relies on MATLAB's own MAT DLL library to isolate format details and eliminate file incompatibilities. As a consequence, all MAT files from Version 4.0 and higher are supported and MATLAB itself is *not required*.

Data Type Support

A MAT file variable is imported into DADiSP as a global variable with the same name and data type as the original. Real and complex arrays are supported. 1x1 arrays are returned as scalars. Members of cell and structure variables are imported as individual global variables prefixed with the original cell or structure name. String variables are also supported.

Requirements

DADiSP/MAT File requires DADiSP 6.0 B16 or higher. MATLAB is not required. Contact us for information about updating your current version of DADiSP.

MAT File Functions

DADiSP/MAT File is a fully menu driven module. However, the following functions can be used on a standalone basis to read MAT files.

MAT File Functions

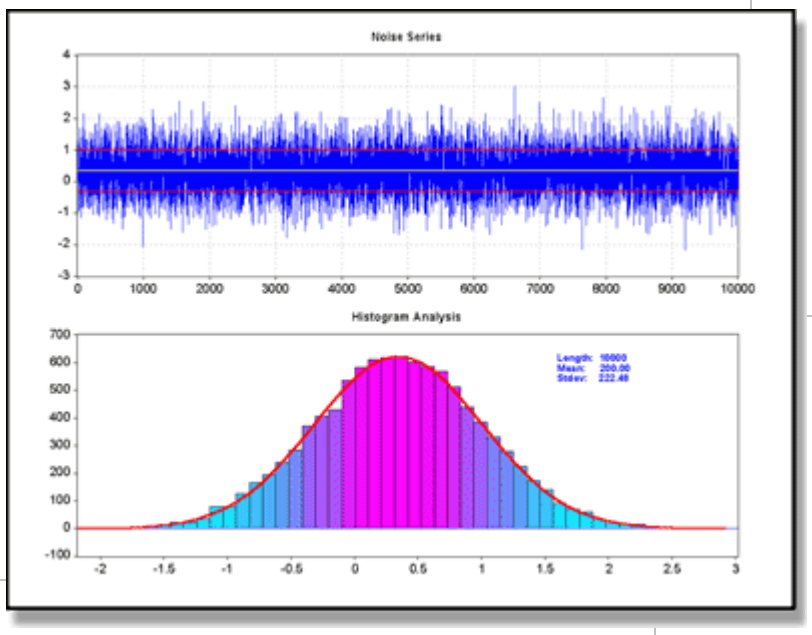
guimat	Displays the dialog box interface to read MAT files.
loadmat	Load selected variables from a MAT file.
matinfo	Returns the number of variables and variable names in a MAT file.

DADiSP / Stats

Statistics Module

DADiSP/Stats is a menu-based module designed to perform a variety of statistical tests and present statistical information graphically. The DADiSP/Stats module computes statistical measures, calculates probability values based on standard distributions, displays summary reports and graphics, and performs hypothesis tests on sample data.

Each menu option pops up a dialog box that provides you with a description of the menu feature and prompts for the location of your sample data. The results of the statistical test are displayed automatically on the screen.

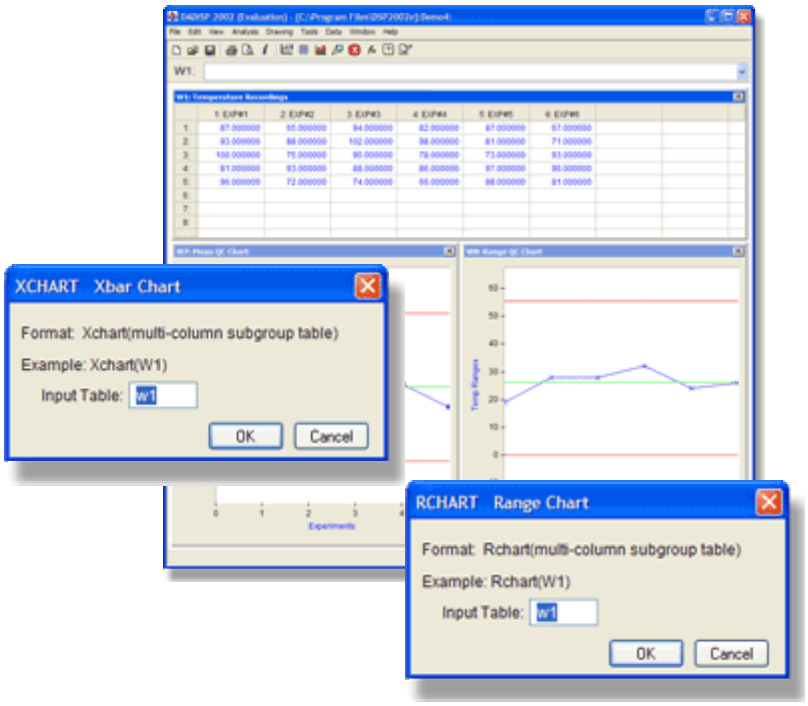


KEY FEATURES

- Simple User Interface
- Descriptive Statistics
- Column and Row Statistics
- One and Two Tailed T Tests
- Chi Square Tests for Multiple Categories
- F Tests
- Two Sample Comparison T Tests for Pooled Variances
- Two Sample Comparison T Tests for Unequal Variances
- One Way ANOVA
- Hypothesis Testing
- Multiple Regression
- Residual Plots
- Scatter Plots
- Histograms
- Box and Error Plots

Statistics Module

DADiSP/Stats is a menu-driven module for DADiSP that computes statistical measures, calculates probability values based on standard distributions, displays summary reports and graphics, and performs hypothesis tests on sample data.



Industry Standard Measurements

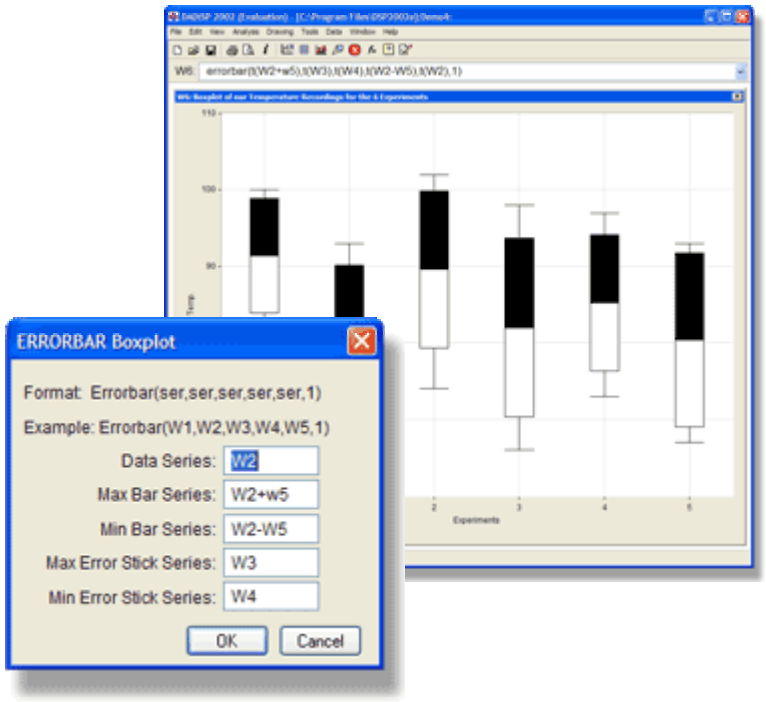
DADiSP/Stats uses DADiSP's familiar graphical Worksheet and pull-down menus. All of the DADiSP analysis and graphics features are available to operate on original input data and on the results of statistical operations. The statistics module offers an additional dimension to DADiSP, providing scientists and engineers with accepted industry statistical measurements to analyze quantitative data.

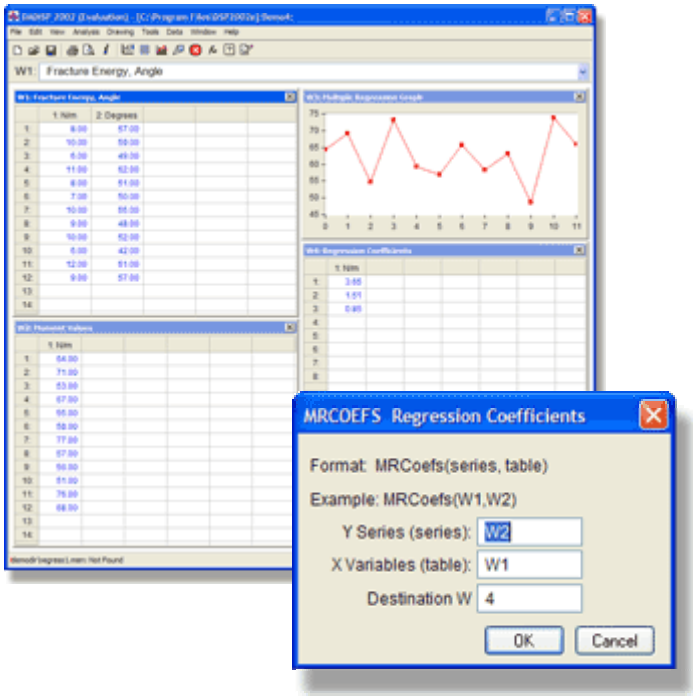
Descriptive Statistics

Sample size, mean, median, variance, standard deviation, standard error, maximum, minimum, and range values are immediately avaliable at the click of a button. Any computation can be used as a stand alone value for further computation or presented as a group in a summary report. Statistical computations can be performed on single datasets or tables with row or column orientation.

Graphical Statistical Analysis

Mean, median, variance, error bar and box plots provide quick visual summaries of common statistical properites. Xbar, Range and combined Xbar / Range plots are appropriate for quality control and process management applications. Histograms, scatter plots and residual plots further complement the statistical visualization features of DADiSP/Stats.



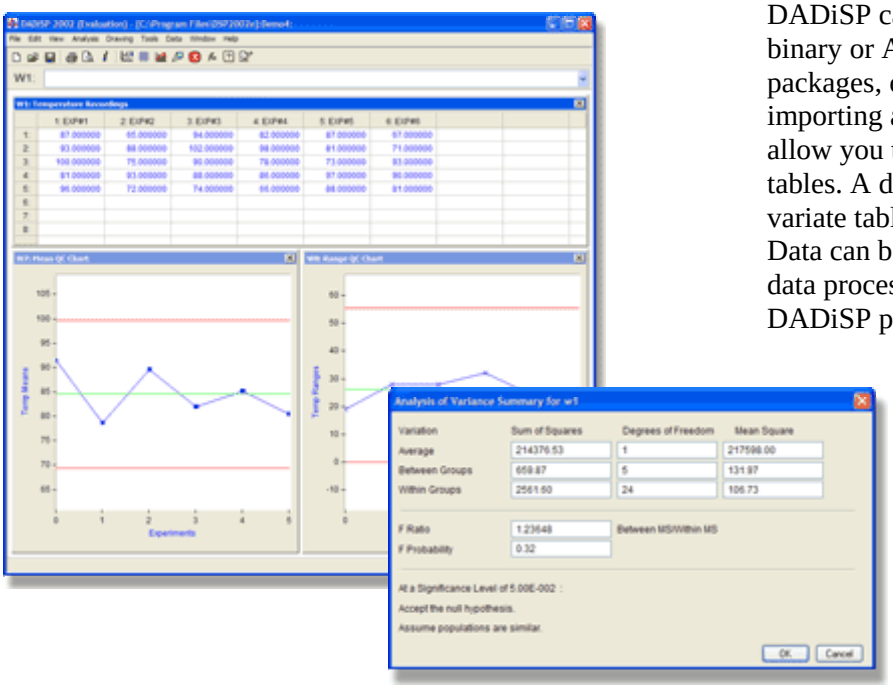


Regression Analysis

Linear and Polynomial fitting offer straightforward statistical modeling capabilities. Sophisticated multiple regression techniques model data with a linear combination of almost any user defined terms. Regression statistics, residual and comparison plots aid in the interpretation of regression results. Chi square goodness of fit statistics provide standard acceptance probabilities and significance levels for regression coefficients.

Data Management

DADiSP can access data from external binary or ASCII files, from other software packages, or directly from the keyboard. The importing and data management features allow you to read single or multi-column tables. A dataset may be stored as a multi-variate table or as a set of individual series. Data can be read into DADiSP for a one-time data processing session or stored the data in DADiSP permanently.



Anova

DADiSP/Stats includes powerful Analysis of Variance routines. Variation, sum of squares, degrees of freedom and mean squared statistics are computed for average, between and within sample groups. F ratio, F probability and significance levels for hypothesis testing are automatically displayed.

Data Compression

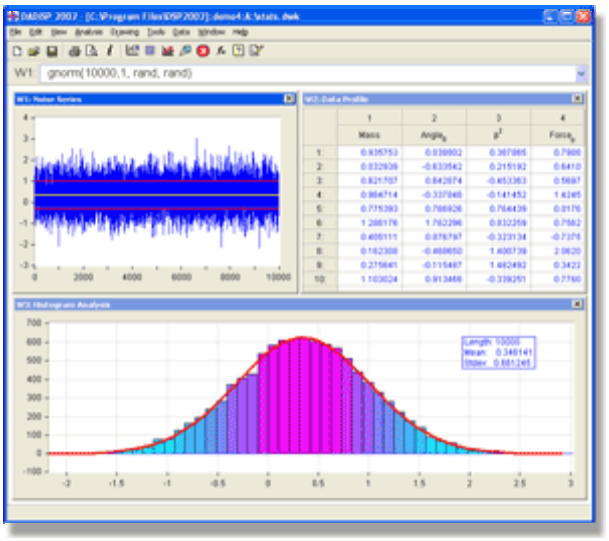
T test sample comparison for pooled and unequal variances compute significance levels for variance testing. One and Two Tailed T tests calculate statistical measurements for mean testing. F Ratio test, cross-correlation, R and R^2 correlation coefficents provide additional tools for statistical data comparison.

Probability Generation

Chi squared, Normal, T Test and F probability density functions are easily generated to provide common standards for several statistical measurements. Lookup functionality is optimized to produce fast and accurate results.

Integrated Results

Each menu option in the statistics menus pops up a dialog box that provides a description of the menu feature and prompts the location of the sample data. The results of each statistical test are displayed automatically on the screen. Each menu option generates a presentation quality screen report or graph that contains the statistical results and text summaries. Each report or graph can be incorporated into a document or sent directly to the printer.



Verification and Automation

The underlying statistical macros and menus can be viewed to verify the statistical algorithms or adapt the individual routines to particular applications. DADiSP/Stats was developed using DADiSP's custom menu and macro extension language. The menu and macro files are simple ASCII text files. The statistical functions are fully documented, including formulas and source references. While DADiSP/Stats is completely menu-driven, any of the new statistics features can be accessed from the command line or from one of DADiSP's automated session scripts.

Stats Functions

DADiSP/Stats includes several functions that encompass a wide variety statistical processing and analysis.

Graphical Statistics Functions

boxplot	Box plot of minimum, maximum, minimum error and maximum error
comboplot	Combined plot of sample mean, median and variance
errorplot	Error bar plot
hist	N bin histogram
meanplot	Plot of sample means
medianplot	Plot of sample medians
polygraph	Polynomial fit plot
rchart	Statictical quality control range chart
varplot	Plot of sample variances
xchart	Statictical quality control mean chart
xrchart	Combined statictical quality control mean and range chart
xyplot	Scatter plot

Regression Functions

mrcoefs	Compute multiple regression coefficients
mrgraph	Plot multiple regression results
mulregsum	Multiple regression computatiuon and summary
polyfit	Linear and polynomial regression

ANOVA Functions

anova	Analysis of Variance summary
correlatesum	cross correlation summary

Descriptive Statistics Functions

max	Sample maximum
min	Sample minimum
mean	Sample average value
median	Sample median value
sersize	Sample size
stderr	Sample standard error
stdev	Sample standard deviation
var	Sample variance

Probability and Hypothisis Test Functions

chimultsum	Chi square goodness of fit for 2xN matrix
chitestsum	Chi square goodness of fit for 2x2 matrix
comp2sum	Two series pooled variance T test
compnoeqsum	Two series separate variance T test
ftestsum	F Ratio test
ttestsum	One tailed T test vs mean
ttest2sum	Two tailed T test vs mean

Probability Density Functions

chisum	Chi square probability
fprobsum	F probability
normsum	Normal (Gaussian) probability
tprobsum	T probability

DADiSP / Controls

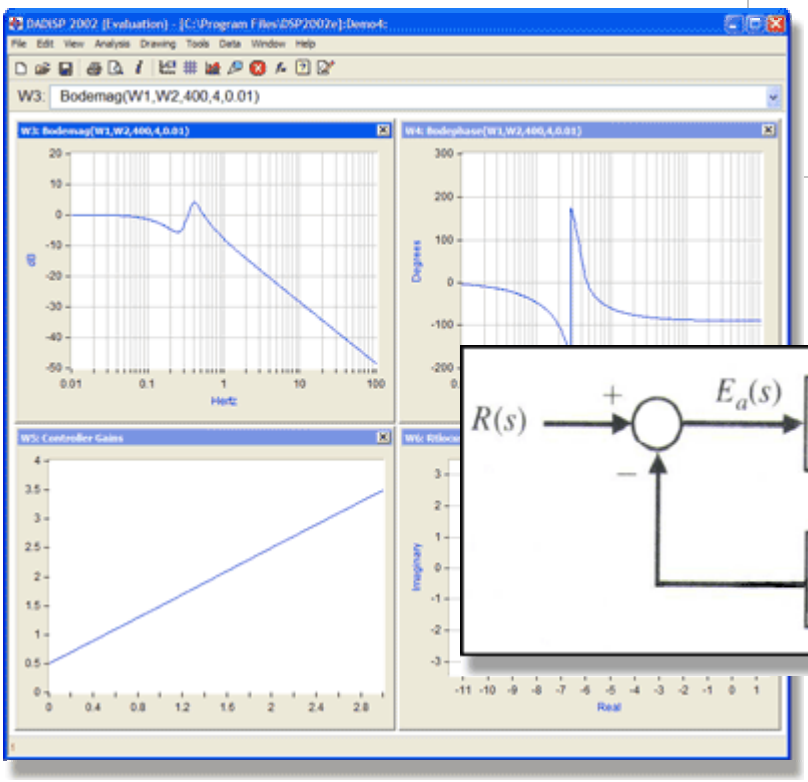
Analog and Digital Controls Module

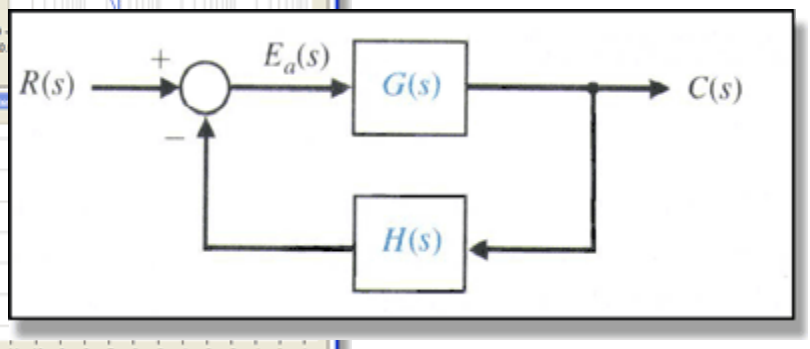
DADiSP/Controls is a menu-driven module that offers easy and accurate design, analysis, and simulation of both discrete and continuous linear time invariant single-input/single-output (SISO) controllers.

The controls module includes menus for the quick design of the most common controllers (PID), simultaneous open and closed loop frequency and time response design of continuous 2nd order systems and the iterative design of lag and lead compensators.

KEY FEATURES

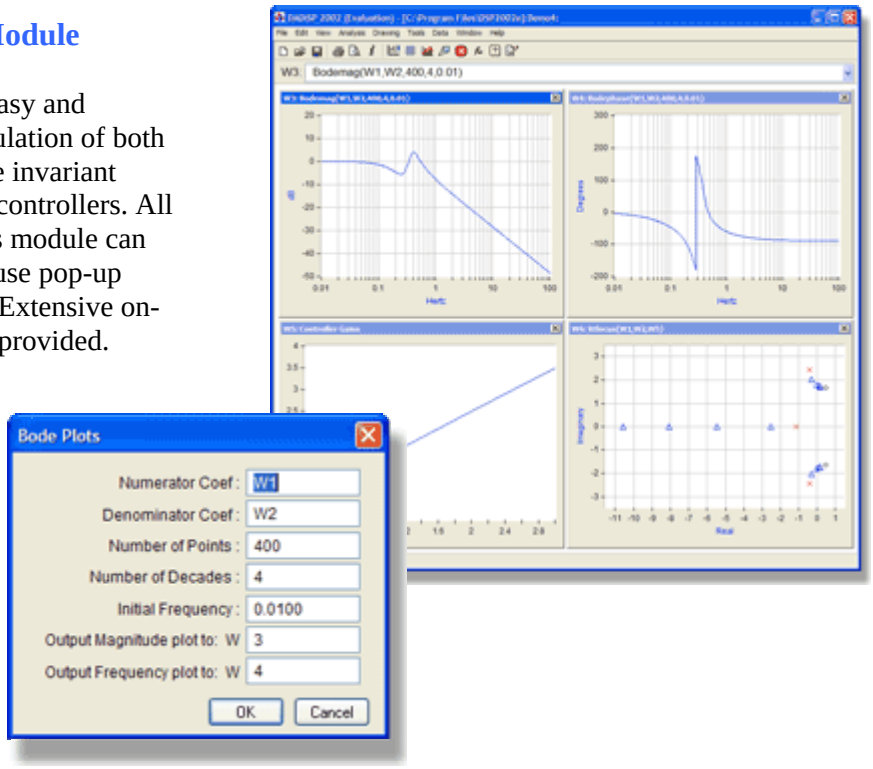
- Simple User Interface
- Iterative Design Method for Common Controllers
- Impulse, Step, Ramp and Frequency Response Calculations
- Bilinear, Backwards Integration and Zero Order Hold Models
- Bode, Nyquist, Root-Locus and Pole-Zero Plots
- Open Loop and Closed Loop Conversion
- PID, PI and PD Designs
- Delay Elements, Lag and Lead Compensators
- 2nd Order Continuous System Design





Analog and Digital Control Module

DADiSP/Controls allows for the easy and accurate design, analysis, and simulation of both discrete and continuous linear time invariant single-input/single-output (SISO) controllers. All of the functionality of the Controls module can be accessed through both easy-to-use pop-up menus and single line commands. Extensive on-line help menus and examples are provided.



Quick and Easy Control Design

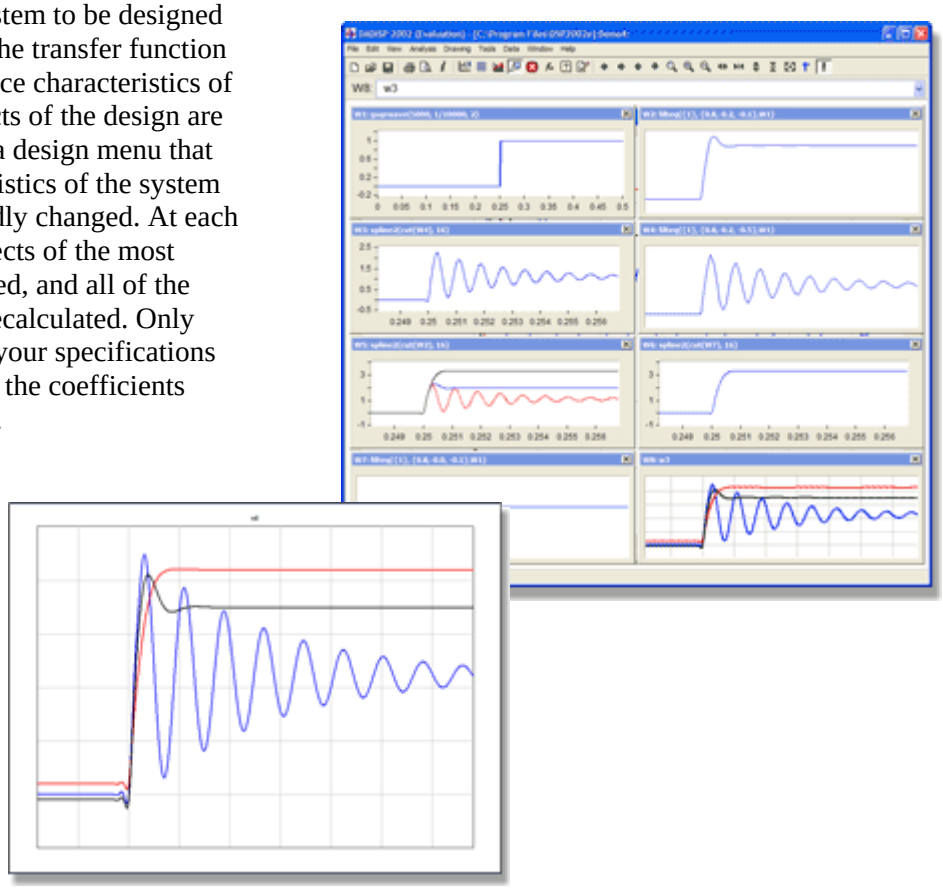
Fully integrated with the DADiSP, DADiSP/Controls is display oriented to show you what is going on in your model. DADiSP/Controls includes menus for the quick design of the most common controllers (PID), simultaneous open and closed loop frequency and time response design of continuous 2nd order systems, and the iterative design of lag and lead compensators.

Continuous to Discrete & Discrete to Continuous Transformation

DADiSP/Controls has a variety of methods to calculate the discrete equivalent of a continuous system as well as the continuous equivalent of a discrete system. Methods include: zero-order hold, bilinear method (Tustin Transform or trapezoidal rule), backward integration method, and zero order hold with processing delay. DADiSP/Controls models delays in continuous systems with either the first or second order Padé approximation to an exponential.

Iterative Design of Common Systems

DADiSP/Controls introduces an iterative method for the design of the most common types of control systems (continuous and discrete phase compensators and continuous 2nd order systems). This method allows for the system to be designed through specification of both the transfer function coefficients and the performance characteristics of the system. The iterative aspects of the design are made possible through use of a design menu that echoes the dominant characteristics of the system and allows them to be repeatedly changed. At each step during the design, the effects of the most recent modification are included, and all of the characteristics of the system recalculated. Only when the entire system meets your specifications and the design is accepted, are the coefficients output to the desired windows.



Simulation with Initial Conditions

To develop a continuous simulation with initial conditions, DADiSP/Controls uses state space realization and eigenvector methods for solving differential equations. Examples provided explain which functions should be used, and demonstrate the proper procedure for developing this type of simulation.

Controls Functions

DADiSP/Controls includes over 40 standalone functions. The following table is a summary of each function.

Off the Shelf Controllers

pid	Design a proportional plus integral plus derivative controller
pi	Design a proportional plus integral controller
pd	Design a proportional plus derivative controller
lagleadm	Design a lag or lead compensator
dpid	Design a discrete proportional plus integral plus derivative controller
dpi	Design a discrete proportional plus integral controller
dpd	Design a discrete proportional plus derivative controller compensator
dlagleadm	Design the discrete equivalent of a continuous lag or lead
dsgn2ordm	Design a 2nd order continuous system

Model Transformation Functions

connect	Produce one composite model from two smaller ones
cloop	Transform open-loop model into its closed-loop equivalent
cloopf	Produce closed-loop transfer fcn for a system with open-loop & feedback dynamics
delay	Model a simple delay in a continuous system
delay2	Model a delay in a continuous system with a higher order approximation
c2disc	Produce discrete model: take Z-transform with zero order hold of the continuous system
c2dbil	Produce the bilinear discrete equivalent of a continuous system
c2dback	Calculate discrete equivalent via the backward integration method
c2delayY	Produce discrete model: take Z-transform with zero order hold with processing delay
dcgain	Calculate DC gain of a continuous system
cresolv	Produce the resolvant matrix of a continuous system
d2cont	Perform inverse Z-transform with zero order hold to produce the continuous model
d2cbil	Produce inverse of the bilinear transform to convert discrete model to continuous equivalent
d2cback	Transform discrete transfer function to continuous equivalent via the inverse of the backward integration method

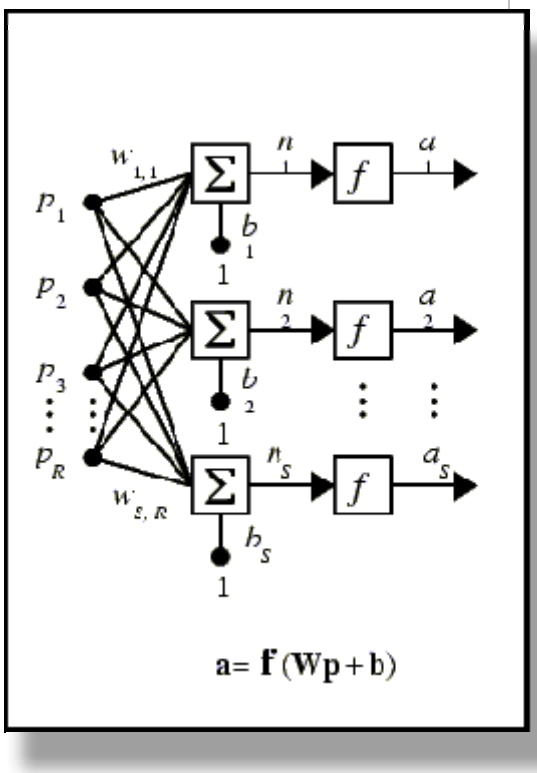
Analysis and Simulation

bode	Produce Bode magnitude and phase plots
nyquist	Generate Nyquist Plot
fstats	Calculate frequency response characteristics from Bode plot
dbode	Generate Bode plots for a discrete system
dnyquist	Produce Nyquist plot for a discrete system
dfstats	Calculate frequency response characteristics from discrete Bode plot
setfunit	Set units to be used by frequency response macros
pzmap	Plot pole and zero locations in the complex plane
rtlocus	Generate Root Locus Plot
pzgrid	Overlay a grid of constant natural frequencies and damping ratios
dpzmap	Plot location of the poles and zeros of a discrete system
drtlocus	Generate Root Locus Plot for a discrete system
dpzgrid	Overlay a grid of constant discrete natural frequencies and damping ratios
cimpulse	Calculate impulse response of a continuous system
cstep	Evaluate step response of a continuous system
cramp	Calculate response of a continuous system to ramp input
csim	Calculate response of a continuous system to specified input
csiminit	Calculate response of a continuous system to specified input and initial conditions
dimpulse	Calculate impulse response of a discrete system
dstep	Evaluate step response of a discrete system
dramp	Calculate response of a discrete system to ramp input
dsim	Calculate response of a discrete system to specified input
dsiminit	Calculate response of a discrete system to specified input and initial conditions
tstats	Calculate performance characteristics from continuous or discrete step response plot

DADiSP / NeuralNet

Neural Network Module

DADiSP/NeuralNet is an add-on module to DADiSP that provides direct and easy access to the demonstrated predictive power and pattern recognition capability of neural networking technology. With DADiSP/NeuralNet, users can build their own artificial neural networks (ANNs) and apply them to achieve more accurate predictions and pattern classifications.



KEY FEATURES

- Simple User Interface
- Automatic Normalization of Data
- Choice of the Number of Hidden Layers
- Unlimited Input and Output Variables
- Unlimited Number of Runs
- Cross-validation Training to Verify Output Results Simultaneously
- Built-in Protection Against Local Minima Distorting Output Results
- User Selectable Desired Mean Square Error, Minimum Gradient Norm and Desired Absolute Error
- Digital Error, Analog Error, Maximum Error and Gradient Values Post-Training Error Graph Types
- Extract Random Seeds Gives the Values Used to Start a Network and Enables Building another Network with the Same Weights
- Extract Network Weights Returns the Weights and Biases that Define the Network

Neural Network Module

Neural networks are able to recognize underlying patterns and predict outcomes based on incomplete or inconsistent information. In contrast to expert systems, which use a static set of rules, neural networks are able to learn and adapt to changes in the environment. With DADiSP/NeuralNet, users can build their own artificial neural networks (ANNs) and apply them to achieve more accurate predictions and pattern classifications.

Neural Network Training

Neural networks resemble the human brain because they can learn. A back-propagation neural network develops its predictive capabilities by being trained on a set of historical inputs and known resulting outputs. The neural net applies random weights to each designated input variable. It then adjusts the weights depending on how closely the actual output values match the desired output values in the training set of historical data. Once the appropriate variable weights have been set that minimize the difference between expected and actual output from the neural net, the neural net can then be applied to new data for classification.

Back-propagation Learning Algorithm

DADiSP/NeuralNet employs the back-propagation learning algorithm. Back-propagation has become the most widely used neural network paradigm for modeling, forecasting, and classification. To minimize the error in the network, DADiSP/NeuralNet uses a rapid-descent algorithm derived from the Vogl method of locating the global minimum. Since results depend on the initial conditions, the neural net module allows you to train a lot of neural networks on the same data with different initial configurations and pick the best one.

Powerful Preprocessing Functions

Preprocessing of the data is one of the largest problems in using neural network tools. DADiSP/NeuralNet is fully integrated with DADiSP, so hundreds of analysis functions are available to pre- and post-process neural network data. DADiSP has mathematical and statistical functions to scale, filter, and process the data to identify features to be learned by the neural network. A typical Worksheet will contain the preprocessing steps, the neural network and the output results. Simply change the input data or initial conditions and each dependent Window is automatically recalculated. You immediately see the effects of your changes on the neural network.

NeuralNet Functions

DADiSP/NeuralNet includes several functions to create, apply and analyze neural networks.

NeuralNet Functions

applynet	Apply a neural network to data
getrundata	Extract a particular set of data for a run
getseeds	Extract neural network random seed values
getweights	Extract neural network weights
makenet	Create a neural network
normalize	Normalize target data to +-1 range

DADiSP / COMTRADE

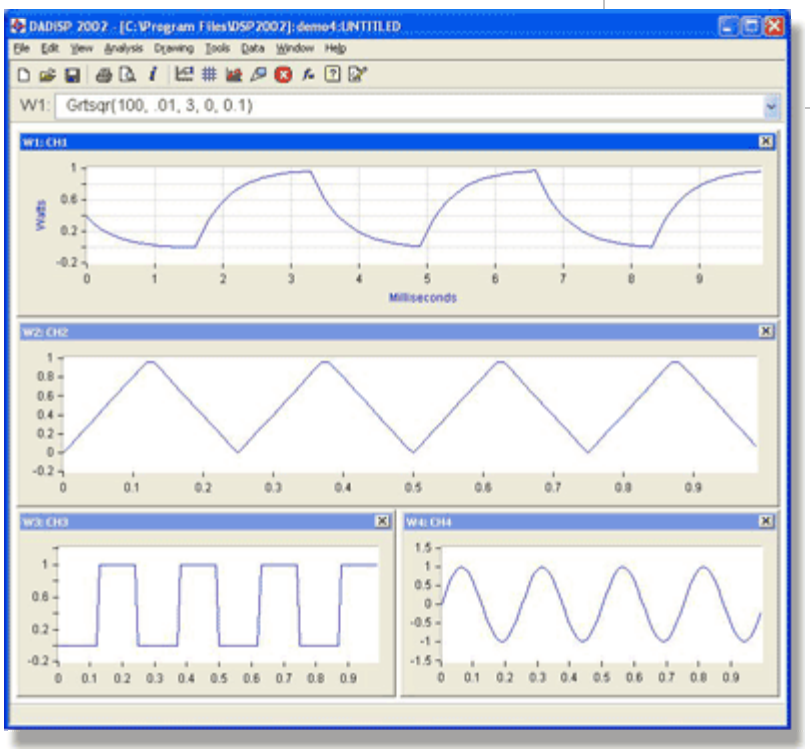
COMTRADE Data File Import Module

DADiSP/COMTRADE is a menu-driven module that extends DADiSP's importing functionality, allowing users to import COMTRADE (IEEE Standard Common Format for Transient Data Exchange) files.

Designed specifically for the power utility indiustry, DADiSP/COMTRADE imports COMTRADE data files by converting information in the COMTRADE configuration file into a fully qualified DADiSP header file. This header is automatically processed by DADiSP's Import Facility, providing a complete import of data into a DADiSP format.

KEY FEATURES

- Simple User Interface
- Fast and Direct COMTRADE File Import
- Automatic Header Conversion Preserves Important Data Properties



COMTRADE Data File Import Module

COMTRADE (Common Format for Transient Data) is an important IEEE standard (C37.111) developed explicitly for the power industry. The standard defines a common format for data files used for the interchange of various types of fault, test or simulation data for electrical power systems. The IEEE standard also describes the sources of transient data such as digital protective relays, digital fault recorders and transient simulation programs and discusses the sampling rates, filters, and sample rate conversions for the transient data being exchanged. The C37.111-1999 standard establishes that the following files are necessary for the data exchange: header file (.hdr), configuration file (.cfg) and data file (.dat).

COMTRADE Header Conversion

DADiSP/COMTRADE imports COMTRADE data files by automatically converting information in the COMTRADE configuration file (*.cfg) into a DADiSP header file. The header is processed by DADiSP's Import Facility to fully import COMTRADE data into a DADiSP series.

Simple Menu Interface

DADiSP/COMTRADE runs from the DADiSP worksheet and is accessed through the toolbar menu. The COMTRADE toolbar menu option utilizes DADiSP's easy-to-use pull-down menus which allow the user to select the COMTRADE Filename; PTR (Potential Transformer Ratio), CTR (Current Transformer Ratio), and the default Series Prefix. Once the COMTRADE parameters have been defined via the module's menu system, the COMTRADE data is imported into DADiSP using the specific data acquisition settings listed in the configuration file.

Full Anlaysis System

DADiSP provides a complete analysis, display, and processing environment using COMTRADE data. The integration of DADiSP/COMTRADE into DADiSP makes it easy to automate data import and analysis applications completely through SPL (Series Processing Language), macros, and command files.

COMTRADE Functions

DADiSP/COMTRADE is a fully menu driven module. However, the following functions can be used on a standalone basis to read COMTRADE files.

COMTRADE Functions

impcomtrade	Import a COMTRADE file into a series.
cnvrtctf	Convert a COMTRADE header into a DADiSP header

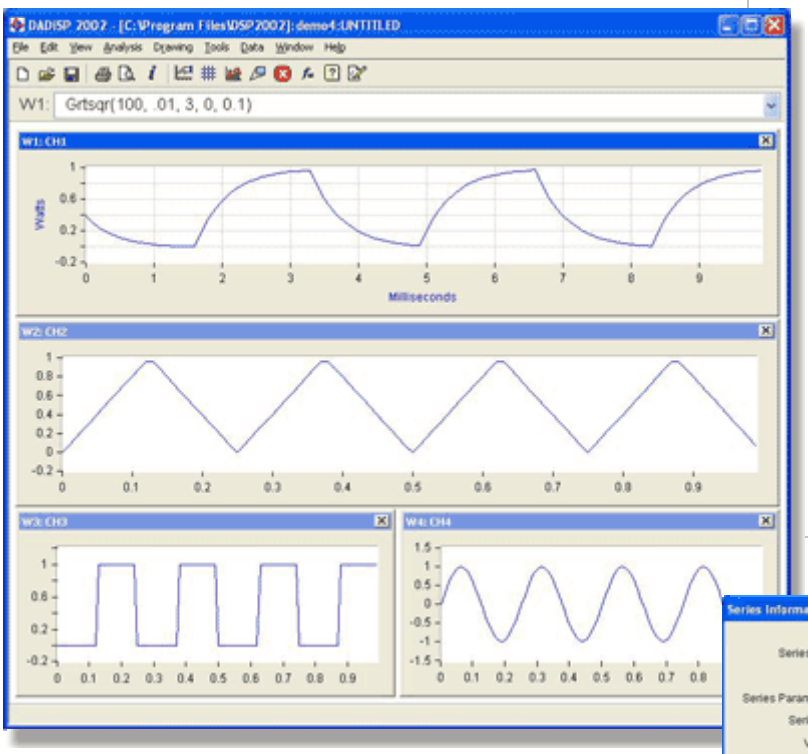
DADiSP / DADiMP

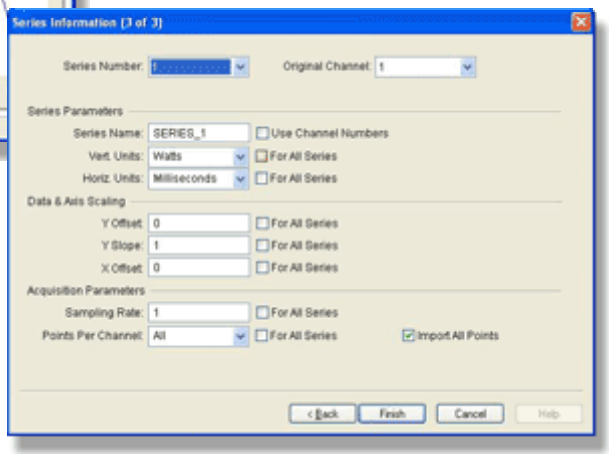
Stand Alone Data Import Module

DADiMP is a separate, standalone software module that extends the DADiSP importing functionality, allowing you to import data files with sophisticated formats. DADiMP imports your ASCII or Binary data files directly into DADiSP's Labbook data structure without actually running the DADiSP program.

KEY FEATURES

- Stand Alone Operation
- Small Executable Size
- Fast and Direct Data File Import
- Command Line or Header Based Import Parameters
- Imports Single Channel or Multi Channel Files
- ASCII and Binary Data Types Supported
- Imports Mixed Binary Files
- Import All Channels or a Selected Subset





Unlike DADiSP's standard internal IMPORT feature, DADiMP is run from the operating system command line or from a batch file or shell script. Because DADiMP and DADiSP are completely decoupled, DADiMP can run remotely, on a separate data collection computer. By providing direct, non-interactive access to your data files, DADiMP lets you automate, speed up, and simplify your standard data collection operations.

New Features

D Previous versions limited the total file size to 2GB. DADiMP 5.0 uses a 64 bit file index to enable handling of very large files. Although each individual series in a file is limited to 2GB, the total file size is now essentially *unlimited* (file sizes in the terabytes). Thus, a data file can now contain any number of series where each series can be up to 2GB in length.

In addition, previous versions of DADiMP import series one at a time, creating overhead in large interlaced files where DADiMP would take a separate pass through the data file for every series imported. DADiMP 5.0 keeps header info and buffers for as many series as possible and imports all of them in a single pass through the file, greatly improving performance on large interlaced files.

Because the DADiMP code base is fully integrated with DADiSP, all new header keywords supported by DADiSP are automatically supported by DADiMP 5.0, including byte swapping, integer byte size, binary data types, date/time formats and dozens of others to simplify the importing of 3rd party ASCII and binary file formats.

DADiMP 5.0 NEW FEATURES SUMMARY

- 64 Bit File Index supports Terabyte File Sizes
- Optimized ASCII and Binary Interlaced Data Importing
- Full DADiSP Header Format and Keyword Support

Stand Alone Import Module

DADiMP is a small, fast running, stand alone executable program that automates and simplifies the task of data file import into a DADiSP Labbook and Dataset. DADiMP can import data files created by your own programs or by commercial database, spreadsheet and word-processing software. Because it is a self contained program, DADiMP can run separately and independently from DADiSP, allowing automated, unattended import jobs.

Data Importing Parameters

Data importing parameters can be specified at run-time, via the DADiMP command line, or through an ASCII Header. In addition, DADiMP can bypass foreign headers and other extraneous information embedded between channels.

Single and Multi-Channel Support

DADiMP can import files with a single channel of data. For example, import a Binary file of 100 points of 2-byte signed integer data sampled at 100 Hertz. DADiMP also understands two basic ways of organizing multiple channels: sequential and interlaced. In sequential files, the data from each channel is a contiguous block -- Channel 1 followed by Channel 2 followed by Channel 3, and so on -- while the channels in interlaced files are interwoven -- sample 1 of Channels 1, 2, and 3 followed by sample 2 of Channels 1, 2, and 3.

Multiple File/Data Types

DADiSP supports a variety of data types, including:

Data Type	Description
ASCII	ASCII data consisting of decimal numbers separated by a space, tab, carriage-return, comma or semicolon
BYTE	unsigned one-byte integer
SBYTE	signed one-byte integer
UINTEGER	unsigned two-byte integer
SINTEGER	signed two-byte integer
LONG	signed four-byte integer
ULONG	unsigned four-byte integer
FLOAT	IEEE four-byte floating point
DOUBLE	IEEE eight-byte double precision floating point

DADiMP can also handle data records with mixed Binary data types.

Selective Importing

All of the fields (channels) in a file can be imported or choose a subset by selecting the column number. Data can be read from the middle of the file or any location by specifying an offset. Using the OFFSET parameter, non-DADiSP header information can be preserved in the original file, but safely bypassed. Bytes or characters between channels can be skipped, eliminating bad or unwanted data.

Time Displacement and Data Scaling

The X_OFFSET, RATE, parameters allow specification of the time base of your data and the Y_OFFSET and SLOPE parameters support independent scaling of each data series. Because uniformally sampled data can be specified with just the X_OFFSET and RATE parameters, the entire time channel can be eliminated, resulting in faster, more robust importing and data processing. Non-uniformally sampled data can be displayed and processed as an XY series.

DADiMP Example Header

DADiSP maintains summary information that describes each series or channel contained in the DADiSP database. To import data with DADiMP, you must supply an ASCII header describing each channel that you plan to import. The header information can be stored in the same file as the data or in a separate file.

The following example shows a standard DADiMP input file containing a data header and four series of ASCII data:

```
LABBOOK      PROCESS1
DATASET      Lot1
SERIES       Pressure, Temperature, Level, Lot
NUM_SERIES   4
STORAGE_MODE INTERLACED
FILE_TYPE    ASCII
RATE         10
VERT_UNITS   PSI, Degrees C, Centimeters, No Units
HORZ_UNITS   Sec
COMMENT      Interlaced multi-channel Process 1, Lot 1
DATA
14.000000,       27.147235,       0.842013,        1
14.187381,       27.032460,       0.901001,        1
14.368125,       27.784237,       2.237624,        1
14.535827,       27.176321,       1.768234,        1


etc...
```

If the filename of this data is `lot1.dat`, the following command line imports the data:

```
dadimp lot1.dat
```

The file is then imported into the Dataset Lot1 located in the Labbook PROCESS1.

To specifically import MATLAB MAT files, see the DADiSP/MAT File Module.

DADiSP / GPIBLab

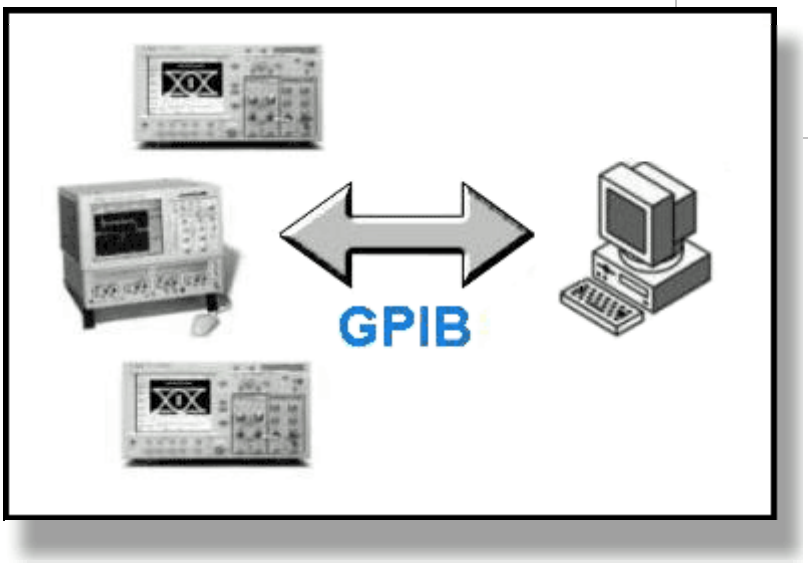
IEEE-488 Instrument Control Module

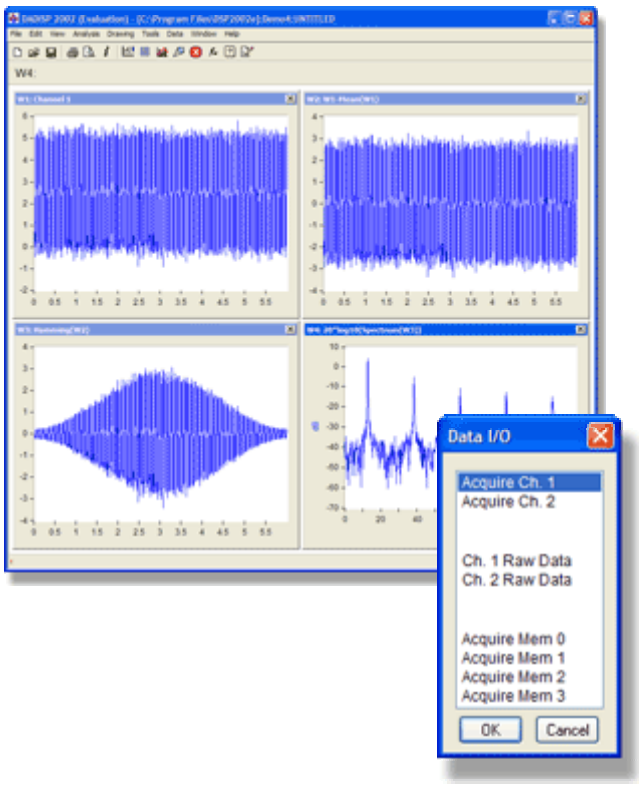
GPIBLab is a menu-driven software module for DADiSP that collects data from IEEE-488 instruments. GPIBLab requires no programming or messy configuration. GPIBLab and DADiSP provide a full range of instrument control and data collection options through the easy-to-understand menus.

Data collected from your instruments via the General Purpose Interface Bus (GPIB) is displayed automatically in DADiSP's multi-window graphical analysis Worksheet. Because GPIBLab operates within DADiSP's graphical Worksheet environment, all of the DADiSP analysis and graphics functionality is available to display, reduce, analyze, and output your data

KEY FEATURES

- Specify Data Collection Parameters through Simple Dialogs
- Query and Control Hundreds of IEEE-488 Instruments
- Direct, High Speed Data Collection
- Easily Automate Instrument Procedures
- Standard Drop-In Macro Template for New Instruments
- Customize the GPIBLab User Interface





IEEE-488 Instrument Control Module

Originally developed by Hewlett-Packard, the General Purpose Interface Bus, GPIB, is a digital interface standard for connecting electronic test and measurement equipment to "controllers" such as personal computers. The bus was standardized by the Institute of Electrical and Electronics Engineers as the IEEE Standard Digital Interface for Programmable Instrumentation, IEEE-488.1.

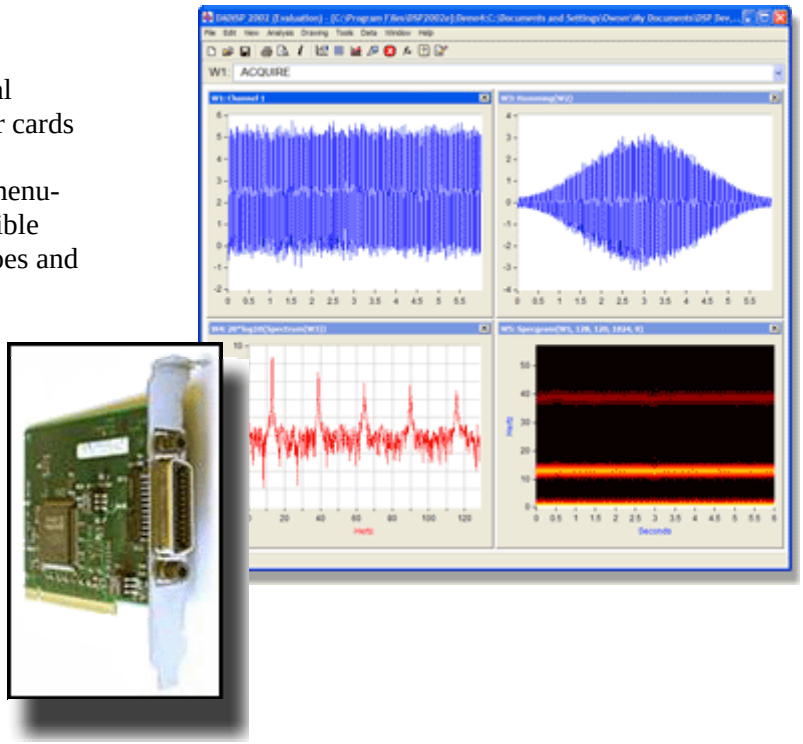
A typical GPIB setup requires a computer, a GPIB control card, driver software for the card, a GPIB cable and instrument. GPIBLab makes it easy to control and transfer data from these instruments without requiring in depth knowledge of the IEEE-488 protocol or the GPIB card driver software.

Fully Integrated

Data collected from your instruments via GPIB is displayed automatically in DADiSP's multi-window graphical analysis Worksheet. Because GPIBLab operates within DADiSP's graphical Worksheet environment, all of the DADiSP analysis and graphics functionality is available to reduce, analyze and output your data.

Standard Instrument Drivers

GPIBLab is available for Windows 9X/2000/NT/XP. It supports IOtech, National Instruments and compatible device controller cards that adhere to the IEEE-488 standard. The GPIBLab product includes a wide range of menu-driven drivers for popular IEEE-488 compatible laboratory instruments, including oscilloscopes and spectrum analyzers.



Let Your Instrument Do The Talking

Users can customize the DSP-supplied ASCII menu drivers to their particular applications, adding additional instrument query, control, and collection functions to the instrument menus. Because GPIBLab menus and command scripts can include standard GPIB functions, GPIBLab eliminates programming in low level languages. All acquisition and analysis operations are accomplished interactively via easy-to-follow menus or through automated DADiSP sessions.

No Programming Required

GPIBLab requires no programming or messy configuration. GPIBLab and DADiSP provide a full range of instrument control and data collection options through the easy-to-understand menu interface. Once the IEEE-488 driver software is installed, GPIBLab users can begin immediately collecting and analyzing data within the DADiSP Worksheet. The size of the data transfer supported by DADiSP/GPIBLab is limited only by the memory on your instrument.

GPIB Instrument Control, Query and Collection Features

Control

- Set Time Range, Volt Range, Time Delay, Offset
- Measurement, Sampling Rate
- Send ASCII commands or binary data to the device
- Setup Service Requests
- Change instrument display screen
- Set interrupt detection on/off
- Operate bus locally or remotely
- Disable instrument front-panel control

Query

- Query Time Range, Volt Range, Time Delay, Offset, Measurement, Sampling Rate
- Query number of bytes in a buffered transfer
- Query status of bus interface

Data Collection

- Transfer ASCII data directly to a DADiSP window
- Transfer binary data directly to a DADiSP window (Size of data transfers limited only by instrument memory)
- Specify terminators between values & end-of-line
- Set Timeout length for data transfers
- Specify data collection triggers
- Specify optional data header
- Specify ASCII data value size

GPIBLab Functions

AlthAlthough most users access GPIBLab through the dialog based interface, GPIBLab includes over 30 standalone functions. Each function can be incorporated int custom SPL routines or macros to provide specific instrument control capability.

The following table is an alphabetical summary of each function.

GPIBLab Functions

abort488	Regains control of the IEEE-488 bus.
arm488	Allows IEEE-488 device driver interface to detect interrupts from specified sources.
buffered488	Displays the number of bytes sent to a buffer in a buffered transfer.
clear488	Returns specified device to a power-on state.
closeieee488	Closes the IEEE-488 device driver opened by the INIT488 command and places GPIBLab in the unintialized state.
config488	Sets up optional header and terminator bytes to skip for ENTER488 and ENTERB488 commands. Also specifies size of ASCII data values and optional terminator.
disarm488	Disables interrupt handling by the PC.
dma488	Enables/Disables DMA data transfer.
enter488	Enters a buffer of ASCII data from an IEEE-488 instrument to a DADiSP window or displays a text message from an instrument. Addresses the instrument for each data point transferred.
entera488	Enters a buffer of ASCII data from an IEEE-488 instrument to a DADiSP window or displays a text message from an instrument. Addresses the instrument only once for the entire buffer transfer.
enterb488	Enters a buffer of binary data from an IEEE-488 instrument to a DADiSP window.
eol488	Sets the end-of-line terminators for input, output, or both from IEEE-488 device driver.
hello488	Checks communication with the IEEE-488 device driver interface.
init488	Provides DADiSP command control over the IEEE-488 bus.
local488	Returns bus devices to manual operation.
lol488	Inhibits front panel operation of bus devices.
output488	Outputs text commands to IEEE-488 devices.
outputb488	Outputs binary data from a window to IEEE-488 devices.
passcontrol488	Allows IEEE-488 device driver interface to give control to another controller source and enter peripheral mode.
ppc488	Configures a device's service request to a particular data line.
ppd488	Disables the parallel poll response of the specified device.
ppoll488	Requests status information from many bus devices simultaneously in the event of a service request.
ppu488	Disables the parallel poll response of all bus devices.
preamble488	Defines an ASCII preamble string to be sent before the binary data in the OUTPUTB488 command.
remote488	Addresses the specified device to listen, placing it in remote state.
request488	Generates a service request from IEEE-488 device driver when in peripheral mode.

reset488	Provides a warm start of the IEEE bus interface.
resume488	Allows data transfers between two peripheral bus devices.
send488	Allows byte-to-byte control over data transfers and great flexibility in issuing commands.
spoll488	Returns 8-bit device response to a serial poll of a device.
status488	Displays the status of the IEEE-488 bus interface.
timeout488	Specifies length of time allowed for a data byte transfer to be completed.
trigger488	Issues a Group Execute Trigger to specified device.