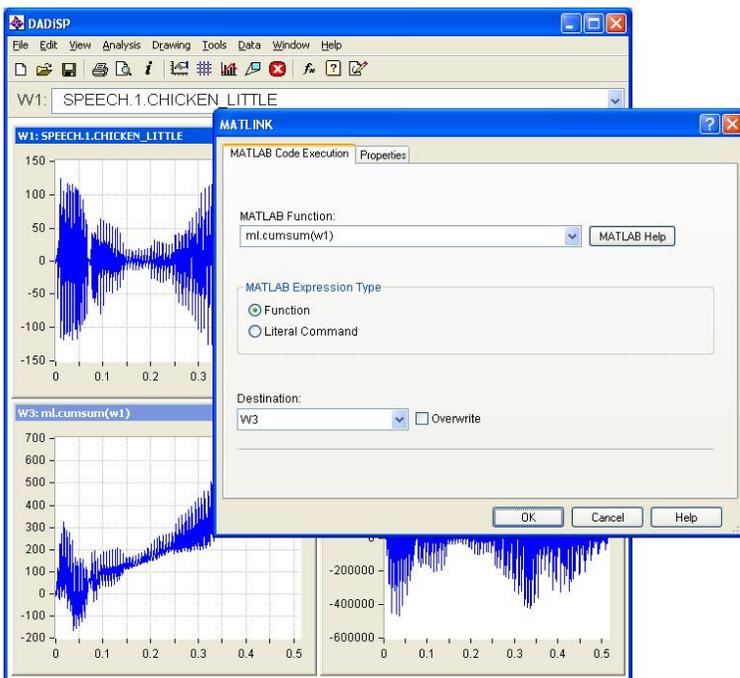# DADiSP / MATLINK

MATLAB® Code Execution Module

MATLINK provides a simple interface for executing MATLAB (1) code directly from DADiSP. Any MATLAB function or script can be processed just as if it were a native DADiSP function.

No need to rewrite code or manipulate temporary data files. MATLINK combines the functionality of existing MATLAB programs with the ease and power of the DADiSP Worksheet environment.

Series, arrays, scalars and strings are exchanged seamlessly. Series and array results returned from MATLAB plot automatically in DADiSP. MATLAB code embedded into a Worksheet Window formula is hot linked and automatically re-executes when the Window updates.

KEY FEATURES

- Seamless Execution of MATLAB Code from DADiSP

- Runs Built-in and Custom MATLAB Functions or Scripts

- Avoid code Rewrites, use any MATLAB Program as is

- Supports Expression Evaluation in Normal MATLAB Syntax

- Freely mix DADiSP Functions and MATLAB Code

- Automatic Visualization of Series and Array Results

- Bi-Directional Exchange of Program Variables

- Handles both Real and Complex Data

- Supports Scalar and String Data Types

- Supports MATLAB Verion 4.0 and Higher

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

MATLAB Code Execution Module

MATLINK runs MATLAB programs directly from DADiSP and exchanges data between DADiSP and MATLAB. Both built-in and custom MATLAB functions or scripts are executed just as if they were native DADiSP functions.

Arrays, scalar and string data are exchanged transparently in fast binary format using the ActiveX Automation interfaces of DADiSP and MATLAB.

How Does It Work?

The key to the simplicity and power of MATLINK resides in the MATLAB Object created by DADiSP. The MATLAB Object supports the execution of any MATLAB expression from the MATLAB environment and returns the results to DADiSP via ActiveX. Because the code is still run in MATLAB, the results are exactly the same as would normally be produced by MATLAB.

ML Dot Syntax

The default name of the MATLAB Object is *ml*. Simply preface the original MATLAB function name with *ml.* to run any MATLAB code from DADiSP. For example, the following DADiSP command displays the current version of the available MATLAB release:

```
ml.version
```

The returned version text is displayed by DADiSP. Because the connection to MATLAB is automatically established when the MATLAB Object is first invoked, explicit initialization is not required.
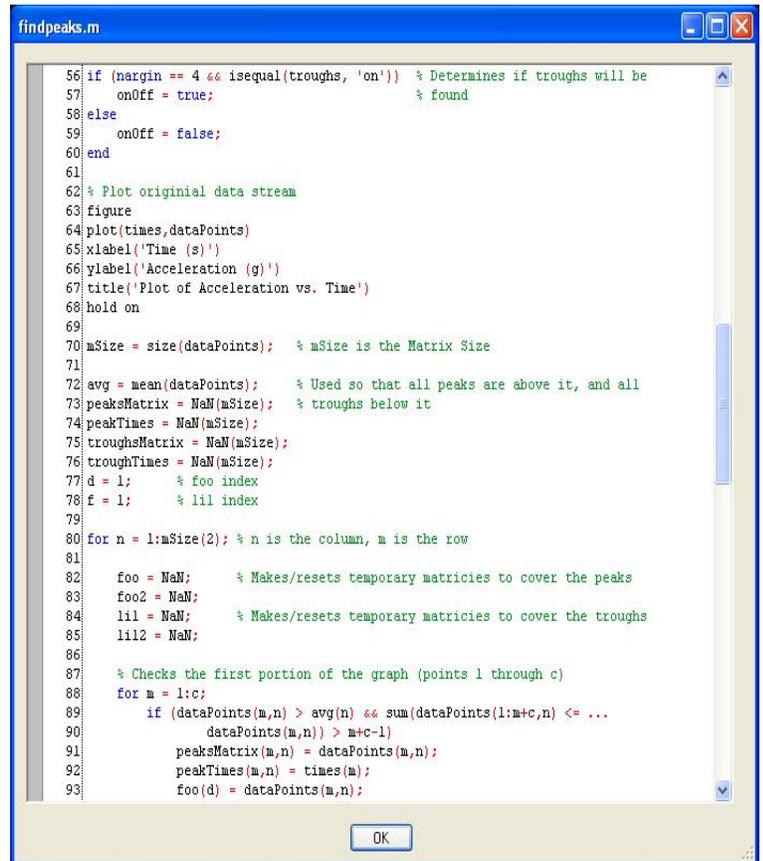

```
findpeaks.m
56 if (nargin == 4 && isequal(troughs, 'on'))  % Determines if troughs will be
57      onOff = true;                            % found
58 else
59      onOff = false;
60 end
61
62 % Plot originial data stream
63 figure
64 plot(times,dataPoints)
65 xlabel('Time (s)')
66 ylabel('Acceleration (g)')
67 title('Plot of Acceleration vs. Time')
68 hold on
69
70 mSize = size(dataPoints);   % mSize is the Matrix Size
71
72 avg = mean(dataPoints);     % Used so that all peaks are above it, and all
73 peaksMatrix = NaN(mSize);   % troughs below it
74 peakTimes = NaN(mSize);
75 troughsMatrix = NaN(mSize);
76 troughTimes = NaN(mSize);
77 d = 1;       % foo index
78 f = 1;       % lil index
79
80 for n = 1:mSize(2); % n is the column, m is the row
81
82      foo = NaN;      % Makes/resets temporary matricies to cover the peaks
83      foo2 = NaN;
84      lil = NaN;      % Makes/resets temporary matricies to cover the troughs
85      lil2 = NaN;
86
87      % Checks the first portion of the graph (points 1 through c)
88      for m = 1:c;
89          if (dataPoints(m,n) > avg(n) && sum(dataPoints(1:m+c,n) <= ...
90                  dataPoints(m,n)) > m+c-1)
91              peaksMatrix(m,n) = dataPoints(m,n);
92              peakTimes(m,n) = times(m);
93              foo(d) = dataPoints(m,n);
```

DSP Development Corporation

Mix and Match

DADiSP and MATLAB functions can be combined and used interchangeably within a Worksheet or SPL functions. For example:

```
W1: gnorm(10000, 1)
W2: ml.diff(W1)
```

W1 contains 1000 samples of Gaussian distributed random noise. The data is then processed by the MATLAB diff function and displayed in W2. When W1 changes, W2 automatically updates and the new data is processed by MATLAB.

The MATLAB diff function executes just like any other DADiSP function except:

    1. DADiSP sends the required data to MATLAB.
    2. MATLAB executes the function.
    3. The results are returned to DADiSP.

The entire process is perfomed automatically, transparently and efficiently.

DADiSP and MATLAB functions can be freely mixed. For example:

```
W1: gnorm(10000, 1)
W2: movavg(ml.diff(w1), 10)
```

The MATLAB diff function is applied to the data in W1 and the result is smoothed by means of a 10 point moving average computed by the DADiSP movavg function. Any combination of DADiSP and MATLAB functions can be combined in this manner.

Data Exchange

Although generally not required, MATLINK supports explicit data exchange between DADiSP and MATLAB variables. For example:

```
// assign a DADiSP series to a MATLAB variable
ml.MatVar = 1..100;

// return a MATLAB variable to DADiSP and process
x = 10 * ml.MatVar;
```

The MATLAB variable MatVar is created and assigned a series of values from 1 to 100. The DADiSP variable x contains the value of the MATLAB variable multiplied by 10. Though required data transfers are automatically handled by the MATLAB Object, the "dot syntax" provides a natural method for exchanging program variables when desired.

MATLAB Command Prompt

MATLAB expressions in original MATLAB syntax are evaluated directly by using the > prompt.

```
W1: >1:100
W2: integ(W1)
```

W1 contains the values 1 through 100 as produced by the MATLAB expression 1:100. W2 integrates the data. Again, a change to W1 automatically updates W2.

Although the equivalent data in W1 can be generated natively by DADiSP with the statement 1..100, the expression 1:100 is standard MATLAB syntax. Any expression in standard MATLAB syntax is supported and evaluated when the expression is prefaced by the > prompt.

Additional Features

The mlhelp function displays the original help text for MATLAB functions. For example:

```
mlhelp eig
```

The standard help information for the MATLAB eig function is displayed in DADiSP.

The mldoc function displays the online documentation for MATLAB functions. For example:

```
mldoc fft
```

The online documentation for the MATLAB fft function is displayed in DADiSP.

Multiple return values from MATLAB functions is supported.

```
W1: rand(10)
W2: (v, d) = ml.eig(W1);d
```

The MATLAB eig function returns two arrays. The second array, a diagonal matrix of eigenvalues, is placed and dislayed in W2.

Error messages or warnings produced by MATLAB are displayed in DADiSP. Functions that produce textual results are presented in a pop-up dialog box. For example, to display a list of currently available MATLAB variables with corresponding data types:

```
ml.whos
```

DSP Development Corporation

Misc Functions

MATLINK functionality is most easily accessed from the MATLAB Object. However, the following routines are available to evaluate MATLAB expressions and exchange data in functional form.

MATLINK Functions

mldoc	Display MATLAB online help
mleval	Evaluate a MATLAB function with optional arguments
mlexic	Execute an expression in MATLAB syntax
mlget	Get data from a MATLAB variable
mlhelp	Display help text on a MATLAB function
mlinit	Initializes the connection with MATLAB and creates a MATLAB Object
mlput	Put data to a MATLAB variable
mlshow	Shows or hides the current MATLAB Automation server

MATLINK requires a licensed copy of MATLAB Version 4.0 or higher to be available from the machine running DADiSP.

DADiSP 6.5 B01 or higher is also required. Contact us for information about updating your current version of DADiSP.