



MATLAB: A Tool for Algorithms Development and System Analysis

Dr. Bradley J. Bazuin
Assistant Professor
Western Michigan University
Dept. of Electrical and Computer Engineering
College of Engineering and Applied Sciences



Engineering Tools

- **Throughout your engineering careers, you will find tricks, techniques, and tools that are useful for dealing with tedious, challenge or just plain difficult engineering.**
- **Build your own personal “Toolbox” of things that help you understand, explain, or prove your ideas.**
- **MATLAB can be one of your tools**
 - So can Mathematica, Maple, etc. can too ... pick the one you work with best!



Why Me ?

Summary of Qualifications :

- **Multi-Rate Digital Signal Processing, including Uniform Filter Bank Analysis and Synthesis, Quadrature Mirror Filtering, Filter-Decimation and Filter Interpolation processing. The derivation and implementation of algorithms, architectures, and ASICs for various signal and communications receivers.**
- **Adaptive Beamforming and Filter Processing. Develop and implement blind adaptive spatial beamforming for interference or jamming mitigation and cochannel signal extraction and recovery. Support to major signal and communications intelligence receiver systems, principal system engineer for anti-jam GPS processor technology, and architect of an ASIC based implementation of a custom spatial-code adaptive processor for a custom discrete multitone retrodirective communications system.**
- **Spread Spectrum Communications. System engineering, design, and program support for custom receivers for frequency hopped and direction sequence spread spectrum signals.**
- **ASIC Design and Development. Design engineer and design center manager for ASIC development. Components include: complex multiplier and special purpose accumulators, a residue number system (RNS) based polyphase filter processor, a quadrature residue number system (QRNS) based Radix -16 and -256 point FFT processors, a custom line-of-bearing processor for multiple antenna phase interferometry applications, and a dual complex multiply-accumulate digital signal processor development with a custom VLIW instruction set.**
- **System architectures for real-time signal processing. Analyze, partition, reconfigure, select alternate implementations, and map complex real-time processing algorithms and systems into ASICs and real-time hardware, real-time DSPs, commercial off the shelf (COTS) boards and modules, and embedded controllers. Providing the system vision and direction for the implementation of advanced concepts.**



Topics

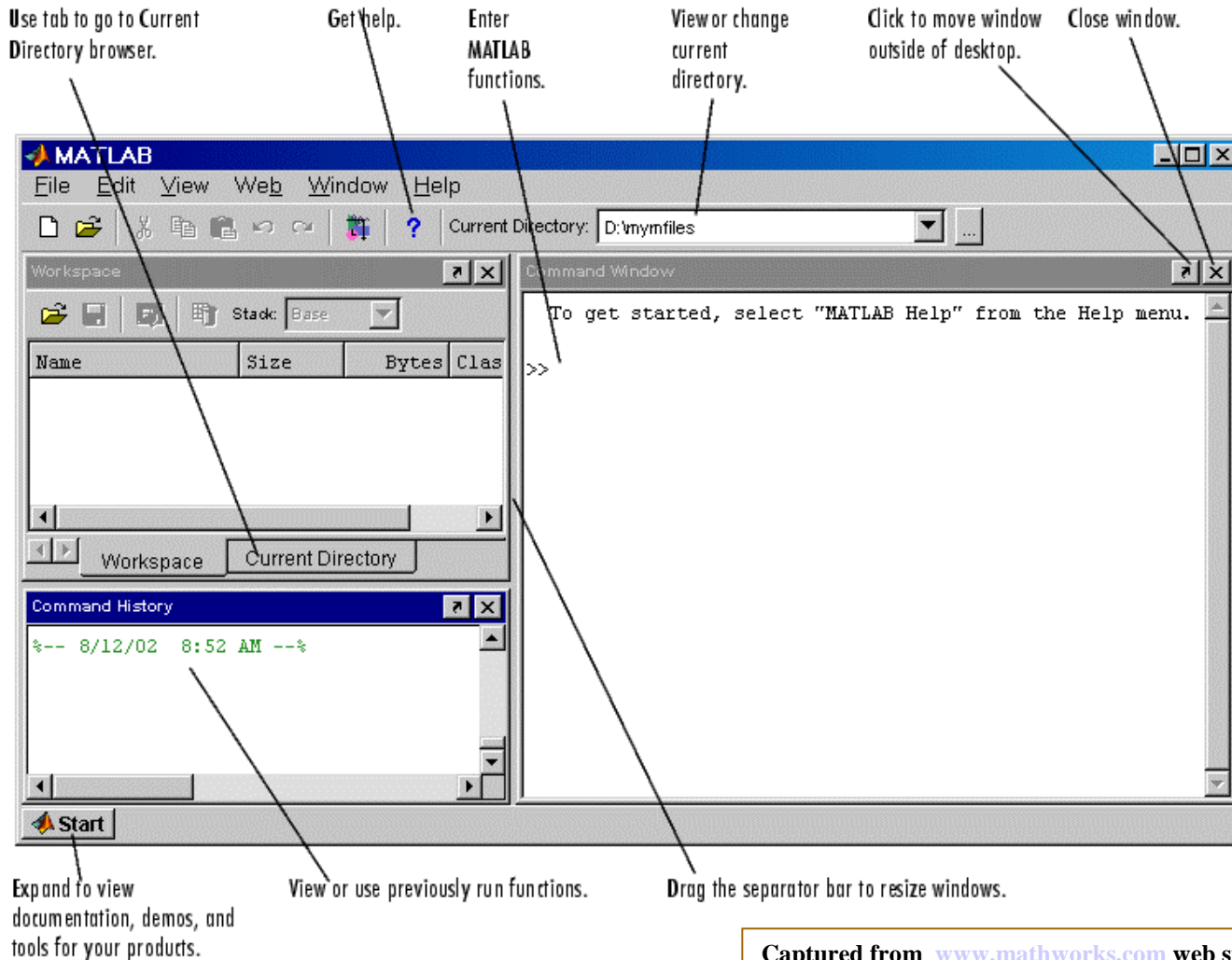
- **MATLAB Basics**
- **Generating Signals**
- **Time Domain and Frequency Domain**
- **Digital Filters (FIR)**
- **Advanced Demonstrations**
 - Digital Filter Bank Analysis
 - Creating a missile flight path
 - GPS
 - Spatial Beamforming
 - Etc.



MATLAB Usefulness

- **Mathematical Verification of Concepts**
 - Linear Systems, Signal Processing, Probability and Statistics, etc.
 - Rapid ability to tweak and perform “what if” trials
 - Visualize what an equation does or says
- **Engineering**
 - System Modeling
 - Algorithm Development and Verification
 - Implementation Validation
 - Test Signal Generation and Results Analysis

MATLAB Basics



Use tab to go to Current Directory browser.

Get help.

Enter MATLAB functions.

View or change current directory.

Click to move window outside of desktop.

Close window.

Expand to view documentation, demos, and tools for your products.

View or use previously run functions.

Drag the separator bar to resize windows.

Start

File Edit View Web Window Help

Current Directory: D:\mymfiles

Workspace

Name	Size	Bytes	Class

Stack: Base

Command Window

To get started, select "MATLAB Help" from the Help menu.

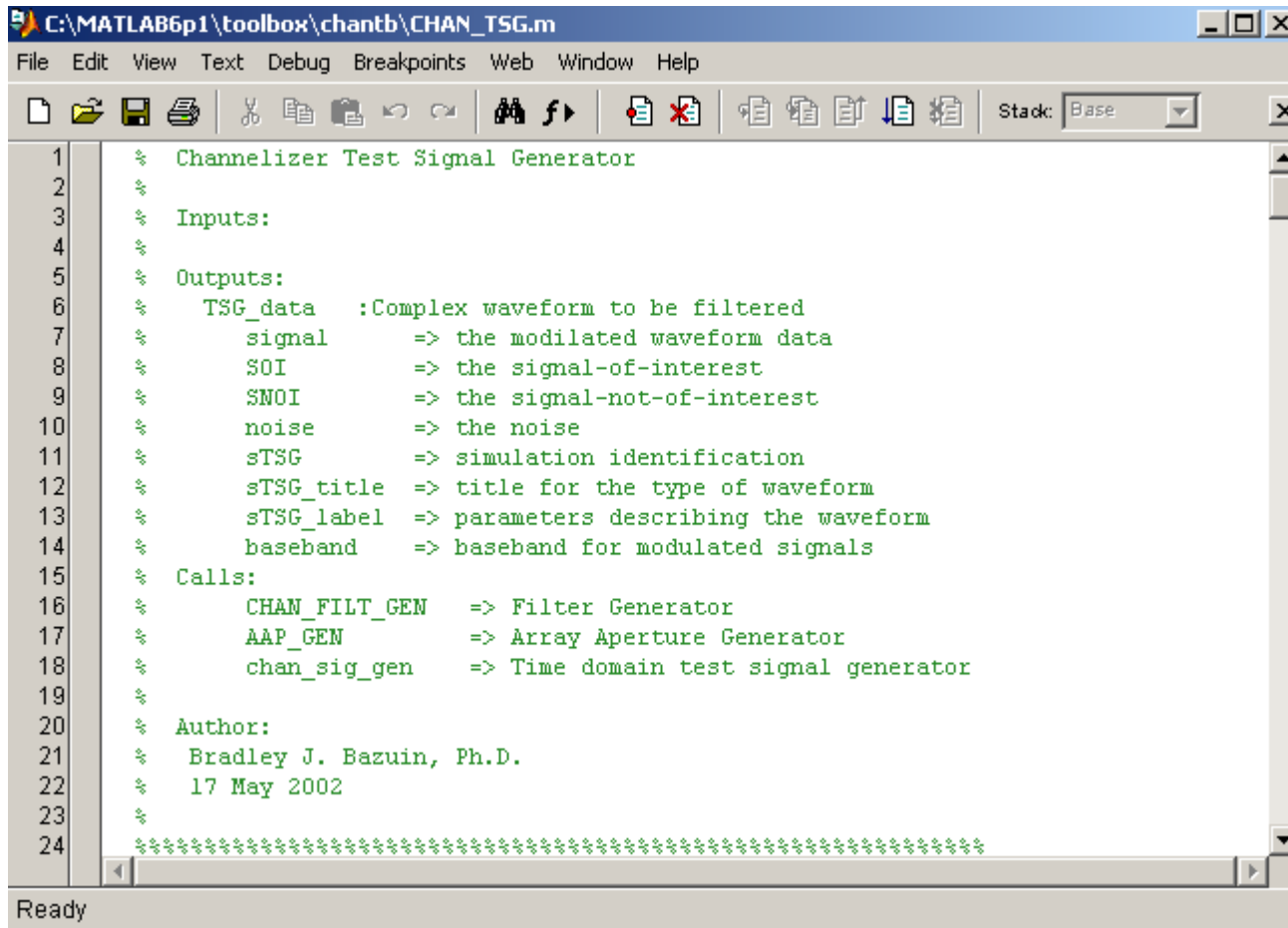
>>

Command History

8/12/02 8:52 AM

Captured from www.mathworks.com web site

MATLAB .m File Edit Window



C:\MATLAB6p1\toolbox\chantb\CHAN_TSG.m

File Edit View Text Debug Breakpoints Web Window Help

Stack: Base

```
1  % Channelizer Test Signal Generator
2  %
3  % Inputs:
4  %
5  % Outputs:
6  %   TSG_data   :Complex waveform to be filtered
7  %   signal    => the modulated waveform data
8  %   SOI       => the signal-of-interest
9  %   SNOI      => the signal-not-of-interest
10 %   noise     => the noise
11 %   sTSG      => simulation identification
12 %   sTSG_title => title for the type of waveform
13 %   sTSG_label => parameters describing the waveform
14 %   baseband  => baseband for modulated signals
15 %
16 % Calls:
17 %   CHAN_FILT_GEN => Filter Generator
18 %   AAP_GEN       => Array Aperture Generator
19 %   chan_sig_gen  => Time domain test signal generator
20 %
21 % Author:
22 %   Bradley J. Bazuin, Ph.D.
23 %   17 May 2002
24 %
*****
```

Ready



The Language: Basic Math Structures

```
% Test Script

a=1
b=2;
a*b
a/b
c=a+b
c=a+b;
for i=1:10
    a=i*i
end
a=0
for i=1:2:25
    a=a+(1/i)*sin(2*pi*(i*b/100)*(1:400));
    plot(a);drawnow;pause
end
plot(a)
```



Row and Column Vectors

```
>> 1:10
```

```
ans =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
>> (1:10)'
```

```
ans =
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```



Vectors (Cont. 1)

```
>> (1:10)*(1:10)
??? Error using ==> *
Inner matrix dimensions must agree.

>> (1:10)*(1:10)'  

ans =  

    385

>> (1:10) .* (1:10)
ans =  

     1     4     9    16    25    36    49    64    81   100

>> (1:10) .^(1:10)
ans =  

1.0e+010 *  

Columns 1 through 7  

    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0001  

Columns 8 through 10  

    0.0017    0.0387    1.0000
```



Vectors (Cont. 2)

```
>> (1:4)'*(1:4)
```

```
ans =
```

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

```
>> sum((1:4)'*(1:4))
```

```
ans =
```

10	20	30	40
----	----	----	----

```
>> cumsum((1:4)'*(1:4))
```

```
ans =
```

1	2	3	4
3	6	9	12
6	12	18	24
10	20	30	40

```
>> prod((1:4)'*(1:4))
```

```
ans =
```

24	384	1944	6144
----	-----	------	------

Vectors (Cont. 3)

```
>> (1:4) *(2:5)
```

```
ans =
```

2	3	4	5
4	6	8	10
6	9	12	15
8	12	16	20

```
>> prod(((1:4) *(2:5)))
```

```
ans =
```

384	1944	6144	15000
-----	------	------	-------

```
>> prod(((1:4) *(2:5))')
```

```
ans =
```

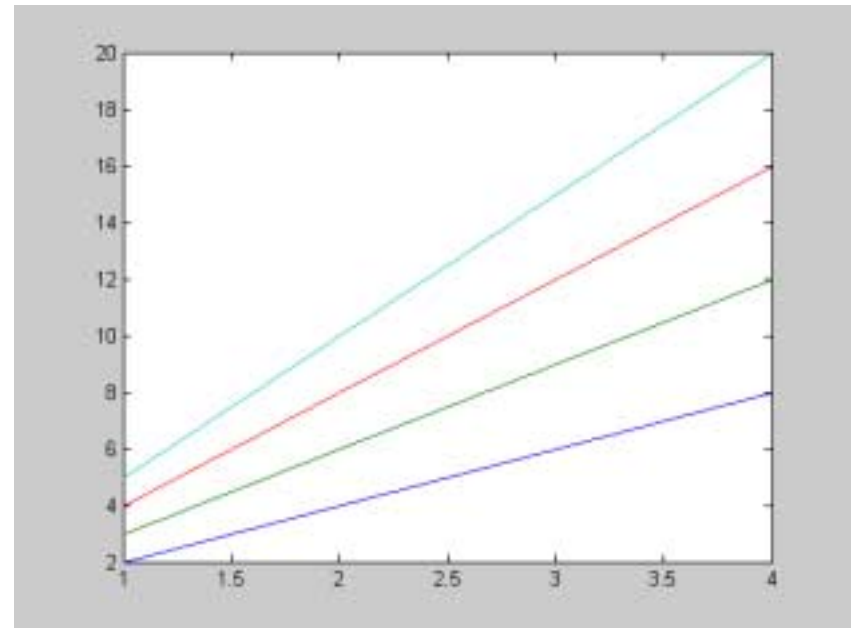
120	1920	9720	30720
-----	------	------	-------

```
>> x=(1:4) *(2:5)
```

```
x =
```

2	3	4	5
4	6	8	10
6	9	12	15
8	12	16	20

```
>> plot(1:4,x)
```





Test Signal Generation

```
function [waveform,stitle,sxlabel,baseband,freq_carrier,rphase,gdB,var1] = ...
    chan_sig_gen(wave_type,xform_size,data_blocks,r_or_c,c_freq)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FUNCTION [waveform,stitle,sxlabel,baseband,freq_carrier] =
%   chan_sig_gen(wave_type,xform_size,data_blocks,r_or_c,c_freq)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generate test signal waveforms
%
% Inputs:
% wave_type    = waveform type
%
%             0) Random Samples Test
%             1) Random Sine Wave Test
%             2) Random Two Tone Test
%             3) Random Block Spectrum with a Notch
%             4) Fixed Comb Generator
%             5) Random Comb Generator
%             6) Random 20 ksps CPFSK
%             7) Random AM PTT Signal
%             8) Random Frequency Ramp
%             9) Impulse Response
%            10) CPFSK with pulse shaping
%            11) CPFSK with pulse shaping & alternating data
%            12) DSB AM Radio (10 kHz BW)
%            13) TVI - DSB AM Radio
%            14) Random Two Tone Test w/ Gain Diff
%            15) Multi-Dwell CPFSK with pulse shaping
%            16) Pulse
%            17) BPSK
%            18) BPSK with pulse shaping
%            19) Spread Spectrum
```



Run This Code

```
num_pts=1024
fs=16
freq_1=0.5

rphase=rand(1,1);
m = exp(sqrt(-1)*2*pi*(freq_1*(1:num_pts)'/fs + rphase));
baseband=ones(num_pts,1);
freq_carrier = freq_1;
fprintf('Sine Wave Test:\n')
fprintf('Tone = %g Phase = %g deg\n',freq_1,360*rphase)
stitle = sprintf('Sine Wave ');
xlabel= sprintf('Tone = %g Phase = %g deg\n',freq_1,360*rphase);
plot(real(m))
title(stitle)
fprintf([stitle '\n' xlabel])
```



Test Signals

Signals

- Sine Wave
- Two Tone
- Phase Modulation
- Etc.

The Calling Routine

```
% Channelizer Test Signal Generator
% Inputs:
% Outputs:
%   TSG_data      :Complex waveform to be filtered
%   signal        => the modulated waveform data
%   SOI           => the signal-of-interest
%   SNOI          => the signal-not-of-interest
%   noise         => the noise
%   sTSG          => simulation identification
%   sTSG_title    => title for the type of waveform
%   sTSG_label    => parameters describing the waveform
%   baseband      => baseband for modulated signals
% Calls:
%       CHAN_FILT_GEN          => Filter Generator
%       AAP_GEN                => Array Aperture Generator
%       chan_sig_gen          => Time domain test signal generator
```

%%%



Parks-McClellan FIR Filters

```
% Parks McClellan Generator
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parks McClellan low pass filter generator                                BJB
% Inputs:
%   LPF_taps    => Number of filter taps
%   N_freq     => Nyquist frequency
%   Wp         => passband frequency
%   Ws         => stopband frequency
%
% Calls:
%   remez      => Parks McClellan Filter coefficients
%
% Outputs:
%   save PMCC_LPF LP_filter sLPF filter_taps filter_gain filter_power
%   LP_filter  => Low pass filter
%   sLPF       => Text filter description
%   filter     => The full set of QMF coefficients
%   LPF_taps   => Assumed number of filter taps
%   LPF_gain   => Coherent gain of the filter
%   LPF_power  => Power of the filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

MATLAB Overview

Questions?

Dr. Bradley J. Bazuin

Western Michigan University, CEAS

Dept. of Electrical and Computer Engineering

brad.bazuin@wmich.edu

<http://homepages.wmich.edu/~bazuinb>