

# **Signal Processing Toolbox™**

## **Release Notes**

---

## How to Contact MathWorks



[www.mathworks.com](http://www.mathworks.com) Web  
[comp.soft-sys.matlab](mailto:comp.soft-sys.matlab) Newsgroup  
[www.mathworks.com/contact\\_TS.html](http://www.mathworks.com/contact_TS.html) Technical Support



[suggest@mathworks.com](mailto:suggest@mathworks.com) Product enhancement suggestions  
[bugs@mathworks.com](mailto:bugs@mathworks.com) Bug reports  
[doc@mathworks.com](mailto:doc@mathworks.com) Documentation error reports  
[service@mathworks.com](mailto:service@mathworks.com) Order status, license renewals, passcodes  
[info@mathworks.com](mailto:info@mathworks.com) Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.  
3 Apple Hill Drive  
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

*Signal Processing Toolbox™ Release Notes*

© COPYRIGHT 2004–2011 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

### Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

### Patents

MathWorks products are protected by one or more U.S. patents. Please see [www.mathworks.com/patents](http://www.mathworks.com/patents) for more information.

<b>Summary by Version</b> .....	<b>1</b>
<b>Version 6.16 (R2011b) Signal Processing Toolbox Software</b> .....	<b>4</b>
<b>Version 6.15 (R2011a) Signal Processing Toolbox Software</b> .....	<b>7</b>
<b>Version 6.14 (R2010b) Signal Processing Toolbox Software</b> .....	<b>10</b>
<b>Version 6.13 (R2010a) Signal Processing Toolbox Software</b> .....	<b>11</b>
<b>Version 6.12 (R2009b) Signal Processing Toolbox Software</b> .....	<b>16</b>
<b>Version 6.11 (R2009a) Signal Processing Toolbox Software</b> .....	<b>17</b>
<b>Version 6.10 (R2008b) Signal Processing Toolbox Software</b> .....	<b>18</b>
<b>Version 6.9 (R2008a) Signal Processing Toolbox Software</b> .....	<b>19</b>
<b>Version 6.8 (R2007b) Signal Processing Toolbox Software</b> .....	<b>20</b>
<b>Version 6.7 (R2007a) Signal Processing Toolbox Software</b> .....	<b>22</b>
<b>Version 6.6 (R2006b) Signal Processing Toolbox Software</b> .....	<b>24</b>

<b>Version 6.5 (R2006a) Signal Processing Toolbox</b>	
Software .....	26
<b>Version 6.4 (R14SP3) Signal Processing Toolbox</b>	
Software .....	28
<b>Version 6.3 (R14SP2) Signal Processing Toolbox</b>	
Software .....	29
<b>Version 6.2.1 (R14SP1) Signal Processing Toolbox</b>	
Software .....	34
<b>Version 6.2 (R14) Signal Processing Toolbox</b>	
Software .....	37
<b>Compatibility Summary for Signal Processing Toolbox</b>	
Software .....	43

## Summary by Version

This table provides quick access to what is new in each version. For clarification, see “Using Release Notes” on page 2.

<b>Version (Release)</b>	<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>
<b>Latest Version V6.16 (R2011b)</b>	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.15 (R2011a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.14 (R2010b)	Yes Details	No	Bug Reports Includes fixes
V6.13 (R2010a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.12 (R2009b)	Yes Details	No	Bug Reports Includes fixes
V6.11 (R2009a)	Yes Details	No	Bug Reports Includes fixes
V6.10 (R2008b)	Yes Details	No	Bug Reports Includes fixes
V6.9 (R2008a)	Yes Details	No	Bug Reports Includes fixes
V6.8 (R2007b)	Yes Details	No	Bug Reports Includes fixes
V6.7 (R2007a)	Yes Details	No	Bug Reports Includes fixes

<b>Version (Release)</b>	<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>
V6.6 (R2006b)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.5 (R2006a)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.4 (R14SP3)	Yes Details	No	Bug Reports Includes fixes
V6.3 (R14SP2)	Yes Details	Yes Summary	Bug Reports Includes fixes
V6.2.1 (R14SP1)	Yes Details	Yes Summary	NA
V6.2 (R14)	Yes Details	Yes Summary	NA

## Using Release Notes

Use release notes when upgrading to a newer version to learn about:

- New features
- Changes
- Potential impact on your existing files and practices

Review the release notes for other MathWorks® products required for this product (for example, MATLAB® or Simulink®). Determine if enhancements, bugs, or compatibility considerations in other products impact you.

If you are upgrading from a software version other than the most recent one, review the current release notes and all interim versions. For example, when you upgrade from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

---

## What Is in the Release Notes

### New Features and Changes

- New functionality
- Changes to existing functionality

### Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, the **Compatibility Considerations** subsection explains the impact.

Compatibility issues reported after the product release appear under Bug Reports at the MathWorks Web site. Bug fixes can sometimes result in incompatibilities, so review the fixed bugs in Bug Reports for any compatibility impact.

### Fixed Bugs and Known Problems

MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time and as more information becomes available. Bug Reports include provisions for any known workarounds or file replacements. Information is available for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

### Documentation on the MathWorks Web Site

Related documentation is available on [mathworks.com](http://mathworks.com) for the latest release and for previous releases:

- Latest product documentation
- Archived documentation

## Version 6.16 (R2011b) Signal Processing Toolbox Software

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> below. See also Summary	Bug Reports Includes fixes

New features and changes introduced in this version are:

- “Passband and Stopband Weights for Fixed-Order Unconstrained Partial Band Differentiator Filters” on page 4
- “Numerator and Denominator Order Specifications Added to filterbuilder for Lowpass and Highpass Butterworth Designs” on page 5
- “Conversion of Error and Warning Message Identifiers” on page 5

### Passband and Stopband Weights for Fixed-Order Unconstrained Partial Band Differentiator Filters

In R2011b, you can specify passband and stopband weights for a fixed-order unconstrained partial band differentiator filter design. You can access this capability through `filterbuilder` and `fdesign.differentiator`. With `fdesign.differentiator`, use the specification string `'N,Fp,Fst'`, and set the design method to `'equiripple'`. The following example shows you how to see the passband weight, `Wpass`, and stopband weight, `Wstop`, design options.

```
d = fdesign.differentiator('N,Fp,Fst',30,0.25,0.5);
designopts(d,'equiripple')
```

Specify the passband or stopband weight values when you design your equiripple filter. For example:

```
Hd = design(d,'equiripple','Wstop',4);
```



## Numerator and Denominator Order Specifications Added to filterbuilder for Lowpass and Highpass Butterworth Designs

In R2011b, you can specify different numerator and denominator orders in filterbuilder for lowpass and highpass Butterworth (maxflat) designs.

Access this option in filterbuilder by setting **Impulse response** under **Filter specifications** to IIR and **Order mode** to Specify. The default is equal order for both the numerator and denominator. Check **Denominator order** to specify a different denominator order.

## Conversion of Error and Warning Message Identifiers

For R2011b, Signal Processing Toolbox™ error and warning message identifiers have changed.

### Compatibility Considerations

If you have scripts or functions that use these changed message identifiers, you must update the code to use the new identifiers. Typically, message identifiers are used to turn off specific warning messages. You can also use them in code that uses a try/catch statement and performs an action based on a specific error identifier.

If your code checks for a message identifier in a warning or error, you must update it to check for the new warning or error instead. To determine the identifier for a warning, run the following command just after you see the warning:

```
[MSG,MSGID] = lastwarn;
```

This command saves the message identifier to the variable MSGID.

To determine the identifier for an error, run the following command just after you see the error:

```
exception = MException.last;  
MSGID = exception.identifier;
```

---

**Note** Warning messages indicate a potential issue with your code. While you can turn off a warning, a suggested alternative is to change your code so that your code does not generate warnings.

---

## Version 6.15 (R2011a) Signal Processing Toolbox Software

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes	Bug Reports Includes fixes

- “Enhancements to `filtfilt`” on page 7
- “Symmetric Window Option for Blackman-Harris Windows ” on page 7
- “`rectpuls` Returns Double-Precision Vector” on page 8
- “Code Generation from MATLAB and Fixed-Point MEX-File Generation” on page 8

### Enhancements to `filtfilt`

In R2011a, there are two major enhancements to `filtfilt`:

- 1** `filtfilt` has been completely rewritten to improve performance. Actual performance improvement depends on your hardware, filter length, signal length, and number of channels.
- 2** `filtfilt` now accepts IIR filters in second-order section (biquad) form.

### Symmetric Window Option for Blackman-Harris Windows

In R2011a, `blackmanharris` and `nuttallwin` have a symmetric window design option. The input argument `SFLAG` controls the window option and defaults to `'symmetric'`. `'periodic'` returns a N-periodic window. The default symmetric option is preferred in FIR filter design because it results in linear phase. In spectral analysis applications, the periodic option is preferred.

## Compatibility Considerations

In releases previous to R2011a, `blackmanharris` and `nuttallwin` only return N-periodic windows. To reproduce behavior in R2011a consistent with behavior in previous releases, use:

```
win = blackman(N,'periodic'); % N is the window length
```

or

```
win = nuttallwin(N,'periodic'); % N is the window length
```

## rectpuls Returns Double-Precision Vector

In R2011a, `rectpuls` returns a double-precision vector instead of a logical vector.

## Compatibility Consideration

In previous releases, `rectpuls` returns a logical vector. To produce behavior in R2011a consistent with previous releases, cast the output of `rectpuls` to a logical vector.

```
t = linspace(0,1,0.01);  
y = logical(rectpuls(t));
```

## Code Generation from MATLAB and Fixed-Point MEX-File Generation

In R2011a, MathWorks is no longer using the term *Embedded MATLAB* to refer to the language subset that supports code generation from MATLAB algorithms. This nomenclature incorrectly implies that the generated code is used in embedded systems only.

The new term is *code generation from MATLAB*. This terminology better reflects the full extent of the capability for translating MATLAB algorithms into readable, efficient, and compact MEX and C/C++ code for deployment to both desktop and embedded systems.

Signal Processing Toolbox users who have the DSP System Toolbox™ and MATLAB® Coder™ software can generate deployable C/C++ code and MEX files using supported functions in the Signal Processing Toolbox. See the

“Code Generation from MATLAB” and “MATLAB Coder” documentation for details.

You can find material on using Code Generation from MATLAB with the Signal Processing Toolbox software in “Code Generation from MATLAB Support in Signal Processing Toolbox”.

Users who have the DSP System Toolbox and Fixed-Point Toolbox™ can accelerate MEX-files for fixed-point applications using `fiaccel`.

### **Compatibility Considerations**

The functionality associated with C/C++ and MEX code generation from MATLAB has changed in R2011a. These changes include:

- The compiler flag `#codegen` replaces `#eml`.
- `codegen` replaces `emlc` and `emlmex` for generating deployable C/C++ code and MEX-files from MATLAB algorithms. You must have the DSP System Toolbox and the MATLAB Coder software to use `codegen`.
- `fiaccel` replaces `emlmex` for generating fixed-point MEX code from MATLAB algorithms. To use `fiaccel`, you must have the DSP System Toolbox and Fixed-Point Toolbox software and your MATLAB code must satisfy the conditions described on the `fiaccel` reference page.

See the release notes and documentation for “Code Generation from MATLAB”, “MATLAB Coder”, and “Fixed-Point Toolbox” for details on how to make your code compliant with the new functionality.

## Version 6.14 (R2010b) Signal Processing Toolbox Software

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports Includes fixes

### Embedded MATLAB Support for Additional Signal Processing Toolbox Functions

In R2010b, Embedded MATLAB® supports `upsample` and `downsample` in the Signal Processing Toolbox.

## Version 6.13 (R2010a) Signal Processing Toolbox Software

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> below. See also Summary.	Bug Reports Includes fixes

New features or changes with compatibility impact in this version are

- “Single-Precision Support Added for `dfilt` Objects” on page 11
- “Embedded MATLAB Support for Additional Signal Processing Toolbox Functions” on page 11
- “Functions, Objects, Object Methods, and Object Properties Being Removed” on page 12
- “Warning for Filter Designer in SPTool” on page 14

### Single-Precision Support Added for `dfilt` Objects

In R2010a, users can construct `dfilt` objects with single-precision floating point arithmetic. Set the `Arithmetic` property to `'single'` to obtain a single-precision floating point representation of the filter coefficients.

### Embedded MATLAB Support for Additional Signal Processing Toolbox Functions

In R2010a, Embedded MATLAB supports additional functions in the Signal Processing Toolbox. You can find a comprehensive list of supported functions at [. You can find examples of using supported functions with the Signal Processing Toolbox software at “Code Generation from MATLAB Support in Signal Processing Toolbox”.](#)

## Functions, Objects, Object Methods, and Object Properties Being Removed

Name	What Happens When you use the Function, Object, Object Method, or Object Property	Use Instead	Compatibility Considerations
addsection (Method for dfilter.cascade and dfilter.parallel objects)	Errors	addstage method	Replace all instances of the addsection method with the addstage method for dfilter.cascade and dfilter.parallel objects.
FFTLength (Property of spectrum objects)	Errors	'NFFT' parameter in psd or msspectrum methods	Replace all instances of the FFTLength property in spectrum objects by the 'NFFT' parameter in the psd or msspectrum methods.
freqzplot	Warns	fvtool	Replace all instances of freqzplot with fvtool.



<b>Name</b>	<b>What Happens When you use the Function, Object, Object Method, or Object Property</b>	<b>Use Instead</b>	<b>Compatibility Considerations</b>
removesection (Method for <code>dfilt.cascade</code> and <code>dfilt.parallel</code> objects)	Errors	removestage method	Replace all instances of the <code>removesection</code> method with the <code>removestage</code> method for <code>dfilt.cascade</code> and <code>dfilt.parallel</code> objects.
Section (Property of <code>dfilt.cascade</code> and <code>dfilt.parallel</code> objects)	Errors	Stage property	Replace all instances of the <code>Section</code> property with the <code>Stage</code> property for <code>dfilt.cascade</code> and <code>dfilt.parallel</code> objects.

Name	What Happens When you use the Function, Object, Object Method, or Object Property	Use Instead	Compatibility Considerations
Sidelobe_atten (Property of sigwin.chebwin object)	Errors	SidelobeAtten property	Replace all instances of the Sidelobe_atten property with the SidelobeAtten property for sigwin.chebwin objects.
specplot	Warns	plot method with DSPDATA object	Replace all instances of specplot with the plot method for DSPDATA objects.

## Warning for Filter Designer in SPTool

In R2010a, the use of Filter Designer in `sptool` is not recommended. Use `fdatool` instead. Under `File > Preferences-> Filter Designer` in SPTool, you may still select to use Filter Designer, but you will receive a warning that Filter Designer will be removed in a future release. If you select Filter Designer, you will be prompted to change your preferences in the `sigprefs.mat` and `startup.spt` files when you exit SPTool. Changing your preferences to use Filter Designer results in a warning each time SPTool starts. See “Setting Preferences” and “Saving and Loading Sessions” for details.

## Compatibility Considerations

Because a future release will remove Filter Designer, use `fdatool` instead. Filters created in Filter Designer are not compatible with `FDATool`. Under `File > Preferences-> Filter Designer` in `sptool`, you can select to use

FDATool. You receive a prompt to convert filters created in Filter Designer to a format compatible with FDATool. When you exit `sptool` after changing your preferences, you receive another prompt, instructing you to update your preferences to use FDATool.

## Version 6.12 (R2009b) Signal Processing Toolbox Software

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports Includes fixes

New features or changes in this version are

- “Embedded MATLAB Support Added to Signal Processing Toolbox Functions” on page 16
- “Ability to Export Filter Coefficients Added to `realizemdl`” on page 16

### Embedded MATLAB Support Added to Signal Processing Toolbox Functions

In R2009b, Embedded MATLAB supports the generation of embeddable C code for a subset of Signal Processing Toolbox filter design and window generation functions. You must install both the Signal Processing Toolbox and Signal Processing Blockset™ software to use this feature. Depending on which Embedded MATLAB feature you wish to use, additional products are required. Please see “Which Code Generation Feature to Use” for a comprehensive list. The generated C code meets the strict memory and data type requirements of embedded target environments. See “Code Generation from MATLAB Support in Signal Processing Toolbox” for a list of supported Signal Processing Toolbox functions and examples.

### Ability to Export Filter Coefficients Added to `realizemdl`

If you use Simulink, you can now use the new `MapCoeffstoPorts` property with `realizemdl` to map filter coefficients from `dfilt` objects to constant blocks. The coefficients also appear in the MATLAB workspace providing tunability to the realized Simulink model. See `dfilt` for a list of supported filter structures and any restrictions.

## Version 6.11 (R2009a) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.11 (R2009a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports Includes fixes

New features or changes introduced in this version are

- “New filter design approach using `fdesign` and `filterbuilder`” on page 17
- “New `dfilt` method to specify filter coefficients at block ports” on page 17

### New filter design approach using `fdesign` and `filterbuilder`

A new and more robust way to design filters has been added to the toolbox—`fdesign` objects and the `filterbuilder` GUI. The following filter responses are supported: lowpass, highpass, bandpass, bandstop, Hilbert, differentiator, pulse-shaping (including FIR Gaussian) and arbitrary magnitude. Advanced design methods and additional responses are available in Filter Design Toolbox™.

### New `dfilt` method to specify filter coefficients at block ports

A new option has been added to the `dfilt` block method to specify filter coefficients via Simulink block ports.

## Version 6.10 (R2008b) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.10 (R2008b):

<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>
Yes Details below	No	Bug Reports Includes fixes

New feature or change introduced in this version is

### **New Walsh–Hadamard Transform functions**

A new fast Walsh–Hadamard transform `fwht` function and an inverse fast Walsh–Hadamard transform `ifwht` have been added to the toolbox. An associated demo has also been added.

## Version 6.9 (R2008a) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.9 (R2008a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports Includes fixes

New features and changes introduced in this version are

- “New Marcum Q Function” on page 19
- “Conversion Between Magnitude and dB Added” on page 19
- “PMTM Function Enhanced with Ability to Keep or Drop Last Taper” on page 19

### New Marcum Q Function

A new `marcumq` function, which implements the generalized Marcum Q function, has been added to the toolbox.

### Conversion Between Magnitude and dB Added

The new utility functions `mag2db` and `db2mag` have been added for converting from magnitude to dB and dB to magnitude, respectively.

### PMTM Function Enhanced with Ability to Keep or Drop Last Taper

You can now specify whether to keep or drop the last taper for calculating the `pmtm`, which returns the PSD using the Thomson multitaper method. By default, the last taper is dropped. If you set the `DropLastTaper` property `false`, the last taper is included.

## Version 6.8 (R2007b) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.8 (R2007b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports Includes fixes

New features and changes introduced in this version are

- “Confidence Interval Estimation Added” on page 20
- “Spurious-Free Dynamic Range (SFDR) Measurement Added” on page 20
- “Local Maxima/Peak Finder Added” on page 20
- “Conversions Between Power and dB Added” on page 21

### Confidence Interval Estimation Added

The spectrum object has been enhanced with a new method for calculating confidence intervals for PSDs and mean-squared spectra.

### Spurious-Free Dynamic Range (SFDR) Measurement Added

The dspdata object has been enhanced with a new method to measure spurious-free dynamic range (SFDR) for mean-squared spectra.

### Local Maxima/Peak Finder Added

A new function `findpeaks` has been added to identify local maxima in a data vector. You can specify the minimum peak height and distance from its neighbors to limit the results. A `findpeaks` method has also been added to the `dspdata` object.



## **Conversions Between Power and dB Added**

The new utility functions `pow2db` and `db2pow` have been added for converting from power to dB and dB to power, respectively.

## Version 6.7 (R2007a) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.7 (R2007a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports Includes fixes

New features and changes introduced in this version are

- “lsf2poly and latcfilt Multi-Channel Input Support Added” on page 22
- “Circular Convolution (cconv) Function Added” on page 22
- “Spectrum Objects Partial Frequency Range Input Support Added” on page 23
- “cceps Factorize Algorithm Information Clarified” on page 23
- “dfilt.statespace Now Supports realizemdl Method” on page 23
- “ellip and ellipap Functions Enhanced” on page 23

### lsf2poly and latcfilt Multi-Channel Input Support Added

lsf2poly and latcfilt now support N-D array input where each column represents a separate input channel.

### Circular Convolution (cconv) Function Added

A new function (cconv) that computes circular convolution has been added to the toolbox.

## **Spectrum Objects Partial Frequency Range Input Support Added**

spectrum objects now support computing the spectrum and pseudospectrum on a user-specified vector of frequencies. This vector identifies the frequencies at which the spectrum or pseudospectrum is calculated.

## **cceps Factorize Algorithm Information Clarified**

A more detailed explanation of the factorize algorithm and an example have been added to the cceps reference page.

## **dfilt.statespace Now Supports realizemdl Method**

You can now create a Simulink block from dfilt.statespace objects with the dfilt realizemdl method.

## **ellip and ellipap Functions Enhanced**

Both ellip and ellipap have been enhanced so that they are able to handle filters with more stringent requirements.

## Version 6.6 (R2006b) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.6 (R2006b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are

- “Frequency Vector Input Added to Spectral Analysis Functions” on page 24
- “FFT Length in Spectral Analyses Changed” on page 25
- “sosfilt and dfilt filter Method Support Multidimensional Array Input” on page 25
- “dfilt block Method Supports Target Subsystem Destination and Link Between Command Line and Model” on page 25
- “gaussfir Algorithm Updated” on page 25

### Frequency Vector Input Added to Spectral Analysis Functions

The spectral analysis command line functions (`pburg`, `pcov`, `peig`, `periodogram`, `pmcov`, `pmtm`, `pmusic`, `pwelch`, and `pyulear`) now accept a frequency vector as an input parameter. This vector identifies the frequencies at which the spectral analysis function returns an estimate.

For functions that use the Goertzel algorithm (`periodogram`, `pmtm`, `pwelch`, and `spectrogram`), the frequency inputs are rounded to match the nearest bin value used by the algorithm.

## FFT Length in Spectral Analyses Changed

The `FFTLength` parameter has been removed from all spectrum objects and you now specify the number of FFT points (NFFT) via the `psd`, `msspectrum`, or `pseudospectrum` estimation method. The NFFT value can be an integer or a string (either `'Nextpow2'` or `'Auto'`). `'Nextpow2'` is the default and sets the number of FFT points to the next power of 2 greater than the input signal length (or the segment length for `spectrum.welch` objects). `'Auto'` sets the number of FFT points to be equal to the input signal or segment length.

## Compatibility Considerations

You should update any existing code that specifies the `FFTLength` parameter and instead use the NFFT parameter associated with an estimation method.

You should also verify that any `spectrum.welch` objects are using the desired FFT length, since the FFT length is now based on the segment length instead of the input signal length.

## `sosfilt` and `dfilt` filter Method Support Multidimensional Array Input

You can now input a multidimensional array to `sosfilt` and to the `dfilt` filter method.

## `dfilt` block Method Supports Target Subsystem Destination and Link Between Command Line and Model

The `dfilt` block method now allows you to specify a target subsystem in your Simulink model where you want to place the block. Two new parameters implement this: `'Destination'` and `'Link2Obj'`. The `'Destination'` specifies where to place the block and `'Link2Obj'` creates a link between the block in your model and inputs from the command line.

## `gaussfir` Algorithm Updated

The `alpha` parameter in the formula used in `gaussfir` has been updated to match the formula in Rappaport T.S., *Wireless Communications Principles and Practice*, 2nd Edition, Prentice Hall, 2001.

## Version 6.5 (R2006a) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.5 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.	Bug Reports Includes fixes

New features and changes introduced in this version are

- “Taylor Window Function Added” on page 26
- “SPTool Filter Designer Replaced by FDATool” on page 26
- “sgolay Example Improved” on page 27
- “zp2sos zeroflag Parameter Added” on page 27
- “Help for Objects Changed” on page 27

### Taylor Window Function Added

A new function `taylorwin` for generating Taylor windows has been added.

### SPTool Filter Designer Replaced by FDATool

FDATool has replaced the SPTool Filter Designer as the preferred method for designing filters for use in SPTool. For details, see “FDATool” in the Signal Processing Toolbox documentation.

### Compatibility Considerations

The format in which filters are saved differs between SPTool Filter Designer and FDATool. When you load an SPTool session with saved filters, you are prompted to upgrade your filters to use FDATool format.

## **sgolay Example Improved**

The example for `sgolay` has been improved and expanded.

## **zp2sos zeroflag Parameter Added**

A new parameter `zeroflag` has been added to `zp2sos`. This parameter is used for real zeros that are the negatives of each other. For these zeros it specifies whether to keep them together instead of ordering them according to proximity to poles.

## **Help for Objects Changed**

To obtain help for objects, use `help object.constructor` instead of the old `help object/constructor`. Note that to obtain help for methods, you still use `help object/method`.

## Version 6.4 (R14SP3) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.4 (R14SP3):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports Includes fixes

New features and changes introduced in this version are described here

### **dfilt (Discrete-Time Filters) Delay Structure Added**

A new delay structure (`dfilt.delay`) has been added to `dfilt` objects. This structure adds latency to any signal filtered with it.

### **WinTool/WVTool Normalize Magnitude Added**

A `Normalize magnitude` option has been added to the **Analysis Parameters** of WinTool and WVTool magnitude plots.

### **FDATool/FVTool Plot Displays Improved**

The default plots for filter responses have been improved. The *y*-axis autoscaling includes buffer regions around the data and shows only the significant data. To see all of the data without any buffer regions, select **Full view** from the **View** menu.

### **FVTool Passband Zoom Added**

If you have a filter in FVTool that was created in FDATool or from a Filter Design Toolbox `fdesign` object, you can use **Passband** on the **View** menu to zoom the passband region.



## Version 6.3 (R14SP2) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.3 (R14SP2):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary	Bug Reports Includes fixes

New features and changes introduced in this version are organized by these topics:

- “FDATool and FVTool Changes” on page 29
- “dfilt Changes” on page 30
- “Spectral Analysis Changes” on page 31
- “Other Changes” on page 32

### FDATool and FVTool Changes

#### FDATool Spectral Rejection Masks Added

You can draw lines on your filter response in FDATool to indicate rejection areas.

#### FDATool Generated C Header File Complex Filter Support

FDATool now supports generating C header files for complex filters.

#### FDATool Tip of the Day Added

A new Tip of the Day dialog displays when you start FDATool. It contains tips and hints for using FDATool.

## FDATool State Space Filters Support Removed

FDATool no longer supports state space filters.

**Compatibility Consideration.** If you load a saved FDATool session that contains a state space filter, it is converted to a direct-form II transposed filter.

## FDATool/FVTool New Analysis Parameters Magnitude Response Options

Three new options have been added to the Analysis Parameters for magnitude response displays.

- **Normalize Magnitude to 1 (0 dB)** — displays the magnitude so that the maximum magnitude value occurs at 0 dB
- **Autoscale axes** — automatically scales the response data  $y$ -axis
- **dB Display Range** — If you are not using autoscale and the magnitude display is in dB, this allows you to specify the  $y$ -axis limits, .

## FVTool SOS Filter Coefficients Display Enhancement

The coefficient view in FVTool now displays each section of a second-order section filter as a separate filter with its own numerator, denominator, and gain.

## FVTool Default Phase Units Changed

The default units for the phase response in FVTool have been changed to radians. This is consistent with the `phasez` function.

**Compatibility Consideration.** You should verify that the FVTool phase plots generated by existing code display the desired units.

## dfilt Changes

### dfilt Coefficients Method Changed

The `dfilt` coefficients method has been changed to the `coeffs` method, which returns a structure. See the Methods section of `dfilt` for information.

**Compatibility Consideration.** You should update any code that uses the `coefficients` method to use the new `coeffs` method and its returned structure.

### **dfilt Filter States Changed to Use States Property**

You cannot pass filter states (initial and final conditions) via the `dfilt filter` method. You must use the `states` property. See `dfilt` for more information.

**Compatibility Consideration.** You should update any code that passed filter states via the `dfilt filter` method to use the new `states` property.

## **Spectral Analysis Changes**

### **spectrogram Function Replaces specgram**

`spectrogram` has been added to replace the grandfathered `specgram` function. If you use this function with no outputs, a surface plot is displayed, instead of an image.

**Compatibility Consideration.** You should update any code that references `specgram` to use the new `spectrogram` function. `spectrogram` uses different default values than `specgram` and the order of the inputs has changed.

### **Spectral Analysis Functions Inputs Changed**

`pwelch` (and the other spectrum analysis functions) no longer accept `'half'` or `'whole'`. You must use `'onesided'` or `'twosided'` to indicate the type of analysis you want.

**Compatibility Consideration.** You should update any code that uses `'half'` or `'whole'` with spectral analysis functions and instead use `'onesided'` or `'twosided'`, respectively.

### **PSD Objects and Function Output Plots Changed**

The following functions and methods now generate standard MATLAB plots, instead of launching an interactive plot. Refer to the MATLAB documentation for information on plots.

- `dspdata` plot method
- `spectrum` `psd`, `pseudospectrum`, and `msspectrum` methods
- `pburg`
- `pcov`
- `periodogram`
- `pmcov`
- `pmtm`
- `pwelch`
- `pyulear`

## Other Changes

### **gaussfir Function Replaces firgauss**

`gaussfir` has been added to replace the grandfathered `firgauss` function. `gaussfir` uses parameters that are common to communications systems.

**Compatibility Consideration.** You should update any code that references `firgauss` to use the new `gaussfir`.

### **firpm and cfirpm Inputs Changed**

The `firpm` and `cfirpm` functions now take function handles as inputs instead of strings.

**Compatibility Consideration.** You should update any code that uses `firpm` or `cfirpm` so that it will work correctly with function handle inputs instead of string input.

### **New Demos**

Signal Processing Toolbox demos have been reorganized and a new demo on the analysis of a numerically controlled oscillator (NCO) has been added.

**Filter Wizard Product Dependency Removed**

The Filter Wizard no longer requires Filter Design Toolbox software. You can use the Filter Wizard if you have Signal Processing Toolbox software and Simulink installed. If you have the Filter Design Toolbox software installed, more options are available. See `dspfwiz` for more information.

## Version 6.2.1 (R14SP1) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.2.1 (R14SP1):

New Features and Changes	Version Compatibility Considerations
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.

New features and changes introduced in this version are

- “FDATool Changing Order of Filters in Cascaded Filter” on page 34
- “FDATool Exporting Filter Coefficient Files Added” on page 34
- “FDATool Exporting Lattice ARMA filters to C Header Files Added” on page 35
- “FDATool Code Generation Dialogs Buttons Changed” on page 35
- “FDATool Undo/Redo Buttons Removed” on page 35
- “FVTool Numeric Format Selection in Coefficient Viewer” on page 35
- “dfilt ResetBeforeFiltering Property Renamed” on page 35
- “dfilt NumSamplesProcessed Property Display Changed” on page 35

### FDATool Changing Order of Filters in Cascaded Filter

The FDATool Filter Manager now has Up and Down buttons that allow you to rearrange the order in which filters are cascaded.

### FDATool Exporting Filter Coefficient Files Added

You can now export filter coefficient files in either decimal or hexadecimal format from FDATool.

## **FDATool Exporting Lattice ARMA filters to C Header Files Added**

You can now export Lattice ARMA filters to C header files from FDATool.

## **FDATool Code Generation Dialogs Buttons Changed**

The code generation dialogs in FDATool now include Generate and Close buttons, instead of OK, Cancel, and Apply buttons.

## **FDATool Undo/Redo Buttons Removed**

The Undo and Redo buttons have been removed from FDATool.

## **FVTool Numeric Format Selection in Coefficient Viewer**

You can now select decimal or hexadecimal numeric format in the FVTool Coefficient Viewer.

## **dfilt ResetBeforeFiltering Property Renamed**

The `ResetBeforeFiltering` property of the `dfilt` object has been renamed to `PersistentMemory`. Refer to the `dfilt` reference page for information.

## **Compatibility Consideration**

You should update any code that uses the `ResetBeforeFiltering` property to use the `PersistentMemory` property.

## **dfilt NumSamplesProcessed Property Display Changed**

The `NumSamplesProcessed` property no longer displays automatically for every `dfilt`. You can view this property with the following command, where `Hd` is a created `dfilt`:

```
get(Hd, 'NumSamplesProcessed')
```

### **Compatibility Consideration**

You should update any code that assumes the `NumSamplesProcessed` is displayed to use the code shown above to display that property.



## Version 6.2 (R14) Signal Processing Toolbox Software

This table summarizes what is new in Version 6.2 (R14):

New Features and Changes	Version Compatibility Considerations
Yes Details below	Yes — Details labeled as <b>Compatibility Considerations</b> , below. See also Summary.

New features and changes introduced in this version are

- “Spectral Estimation Objects Changes” on page 37
- “dfilt Enhancements” on page 38
- “Filter States Object Added” on page 39
- “sigwin Enhancements” on page 40
- “FDATool/FVTool Changes” on page 40
- “Function Name Changes” on page 41
- “New Demos” on page 42

### Spectral Estimation Objects Changes

#### Spectrum Object Added

Signal Processing Toolbox version 6.2 adds a new spectrum object. Spectrum objects contain parameter information for a particular spectral estimation method (e.g., `spectrum.welch`). This object provides an improved way to view and manipulate spectral estimation parameters. See the `spectrum` reference page and the associated estimation method reference pages for more information.

#### Dspdata Object Added

A new data object `dspdata` has been added. These data objects contain parameter information for spectral data. Available types of data objects are

`dspdata.msspectrum`, `dspdata.psd`, and `dspdata.pseudospectrum`. These objects contain the mean-square spectrum data, the power spectral density data, and the pseudospectrum data, respectively. With these objects, you can apply methods that calculate certain values or modify the data, such as computing the average power, centering the DC value, converting the spectrum between one-sided and two-sided, and normalizing the frequency. You can plot the data in a new **Spectral Analysis** plot. See `dspdata` for information.

### **Spectrum Options Object Added**

A new options object `dspspts` has been added. This object contains optional parameter information for spectrum objects. You create options objects via methods on other objects.

### **Spectral Estimation Functions Enhancement**

All of the spectral estimation functions (`pburg`, `pcov`, `peig`, `periodogram`, `pmcov`, `pmtm`, `pmusic`, `pwelch`, and `pyulear`) now use the new **Spectral Analysis** plot when no output arguments are specified. For an example of this plot, see the spectrum reference page.

### **psd and psdplot Functions Replaced by psd and plot methods**

The `psd` method on spectrum objects or `plot` method on `dspdata.psd` objects replaces the obsoleted `psd` function.

**Compatibility Consideration.** You should update any code that uses `psd` to use the `plot` or `dspspts.psd` method.

### **pmem Function Replaced by spectrum.yulear**

The `pmem` function has been replaced by the new `spectrum.yulear` object.

**Compatibility Consideration.** You should update any code that uses `pmem` to use `spectrum.yulear` instead.

### **dfilt Enhancements**

- The leading coefficient of the denominator (`a0`) can not be set to zero.

- New methods have been added to `dfilt` objects:
  - `fcfwrite` — writes an ASCII filter coefficient file.
  - `info` — displays filter structure, order, stability and linear phase information.
- The terminology used for multistage and second-order sections (SOS) filters has been clarified and some `dfilt` method names have been changed. SOS filters have sections, where sections compose a single filter. Cascade and parallel filters have stages, where each stage is a separate filter. See the "Methods" section of the `dfilt` reference page for information on methods related to sections and stages. The new or renamed methods are
  - `addstage` — adds a stage to a parallel or cascade filter object.
  - `nstages` — returns the number of stages in the filter.
  - `removestage` — removes a stage from a parallel or cascade filter object.
  - `setstage` — overwrites a stage of a parallel or cascade filter object.

## Filter States Object Added

A new filter states object `filtstates` with two properties—Numerator and Denominator, that contain the filter states for an IIR Direct-form I filter, has been added. The states property of a `dfilt.df1`, `dfilt.df1t`, `dfilt.df1sos`, or `dfilt.df1tsos` filter now contains a `filtstates.df1ir` object, instead of a double vector. See `filtstates` for more information.

Note that you cannot change the values in the Numerator or Denominator using this syntax:

```
Hd.states.numerator = [<data>]
```

## Compatibility Considerations

The sign of the Denominator states for `dfilt.df1t` and `dfilt.df1tsos` is negated from what it was in previous versions. Refer to the structure diagrams shown on the reference pages.

If you use the `double` command on a filter states object to get a vector of double-precision values of the numerator and denominator, the numerator

values are listed first, followed by the denominator values. In previous versions, the denominator values were listed first.

States saved for SOS objects in previous versions cannot be loaded into the current version.

## **sigwin Enhancements**

Two new methods have been added to `sigwin` objects:

- `winwrite` — writes an ASCII file containing window weights
- `info` — displays information about the window object

## **FDATool/FVTool Changes**

### **FDATool Pole/Zero Editor Added**

A new graphical **Pole/Zero Editor** panel has been added. You can graphically edit the filter (move, add, and delete poles/zeros) and see the resulting effect on your filter design.

### **FDATool File Export Enhancements**

**File->Export** changes include adding the ability to export the filter directly to SPTool, exporting an ASCII coefficient file instead of a text file, exporting cascades and parallel structures, and general improvements to the dialog box.

### **FDATool New Design Methods Added**

FIR Constrained Least Squares (`fircls`) and complex and nonlinear-phase equiripple FIR (`cfirpm`) filters have been added as design methods.

### **FDATool Filter Manager Added**

A Filter Manager has been added. You can use the Filter Manager to store and recall multiple filters in a single FDATool session. You can also use it to cascade filters easily.

## **FDATool/FVTool Analysis Parameters Improved**

The Analysis Parameters dialog box has been reorganized and simplified. The Frequency Units field has been removed and a **Normalized Frequency** checkbox has been added. (This also applies to FVTool.)

## **FVTool Changing Sampling Frequency**

From the command line, you can change the sampling frequency ( $F_s$ ) for the filter(s) in FVTool. See the Controlling FVTool from the MATLAB Command Line section of `fvtool` for more information.

## **Function Name Changes**

### **remez Function Replaced by firpm**

The `remez` function has been replaced by the new `firpm` function.

**Compatibility Consideration.** You should update any code that uses `remez` to use `firpm` instead.

### **cremez Function Replaced by cfirpm**

The `cremez` function has been replaced by the new `cfirpm` function.

**Compatibility Consideration.** You should update any code that uses `cremez` to use `cfirpm` instead.

### **remezord Function Replaced by firpmord**

The `remezord` function has been replaced by the new `firpmord` function.

**Compatibility Consideration.** You should update any code that uses `remezord` to use `firpmord` instead.

### **tfe Function Replaced by tfestimate**

The `tfe` function has been replaced by the new `tfestimate` function.

**Compatibility Consideration.** You should update any code that uses `tfe` to use `tfestimate` instead. Note that `tfestimate` has different default values and may produce different results.

### **csd Function Replaced by cpsd**

The `csd` function has been replaced by the new `cpsd` function.

**Compatibility Consideration.** You should update any code that uses `csd` to use `cpsd` instead. Note that `cpsd` has different default values and may produce different results.

### **cohere Function Replaced by mscohere**

The `cohere` function has been replaced by the new `mscohere` function.

**Compatibility Consideration.** You should update any code that uses `cohere` to use `mscohere` instead. Note that `mscohere` has different default values and may produce different results.

## **New Demos**

Two new demos have been added:

- **Waveform Generation** — generates common waveforms, such as pulse trains.
- **DTMF** — generates telephone pad tones.

## Compatibility Summary for Signal Processing Toolbox Software

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided with the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
Latest Release V6.16 (R2011b)	See the <b>Compatibility Considerations</b> subheading for this new feature or change: <ul style="list-style-type: none"> <li>• “Conversion of Error and Warning Message Identifiers” on page 5</li> </ul>
V6.15 (R2011a)	See the <b>Compatibility Considerations</b> subheading for this new feature or change: <ul style="list-style-type: none"> <li>• “Symmetric Window Option for Blackman-Harris Windows” on page 7</li> <li>• “rectpuls Returns Double-Precision Vector” on page 8</li> <li>• “Code Generation from MATLAB and Fixed-Point MEX-File Generation” on page 8</li> </ul>
V6.14 (R2010b)	None
V6.13 (R2010a)	See the <b>Compatibility Considerations</b> subheading for this new feature or change: <ul style="list-style-type: none"> <li>• “Functions, Objects, Object Methods, and Object Properties Being Removed” on page 12</li> <li>• “Warning for Filter Designer in SPTool” on page 14</li> </ul>
V6.12 (R2009b)	None
V 6.11 (R2009a)	None
V6.10 (R2008b)	None
V6.9 (R2008a)	None

<b>Version (Release)</b>	<b>New Features and Changes with Version Compatibility Impact</b>
V6.8 (R2007b)	None
V6.7 (R2007a)	None
V6.6 (R2006b)	<p>See the <b>Compatibility Considerations</b> subheading for this new feature or change:</p> <ul style="list-style-type: none"> <li>• “FFT Length in Spectral Analyses Changed” on page 25</li> </ul>
V6.5 (R2006a)	<p>See the <b>Compatibility Considerations</b> subheading for this new feature or change:</p> <ul style="list-style-type: none"> <li>• “SPTool Filter Designer Replaced by FDATool” on page 26</li> </ul>
V6.4 (R14SP3)	None
V6.3 (R14SP2)	<p>See the <b>Compatibility Considerations</b> subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> <li>• “FDATool State Space Filters Support Removed” on page 30</li> <li>• “FVTool Default Phase Units Changed” on page 30</li> <li>• “dfilt Coefficients Method Changed” on page 30</li> <li>• “dfilt Filter States Changed to Use States Property” on page 31</li> <li>• “spectrogram Function Replaces specgram” on page 31</li> <li>• “Spectral Analysis Functions Inputs Changed” on page 31</li> <li>• “gaussfir Function Replaces firgauss” on page 32</li> <li>• “firpm and cfirpm Inputs Changed” on page 32</li> </ul>



<b>Version (Release)</b>	<b>New Features and Changes with Version Compatibility Impact</b>
V6.2.1 (R14SP1)	<p>See the <b>Compatibility Considerations</b> subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> <li>• “dfilt ResetBeforeFiltering Property Renamed” on page 35</li> <li>• “dfilt NumSamplesProcessed Property Display Changed” on page 35</li> </ul>
V6.2 (R14)	<p>See the <b>Compatibility Considerations</b> subheading for each of these new features or changes:</p> <ul style="list-style-type: none"> <li>• “psd and psdplot Functions Replaced by psd and plot methods” on page 38</li> <li>• “pmem Function Replaced by spectrum.yulear” on page 38</li> <li>• “Filter States Object Added” on page 39</li> <li>• “remez Function Replaced by firpm” on page 41</li> <li>• “cremez Function Replaced by cfirpm” on page 41</li> <li>• “remezord Function Replaced by firpmord” on page 41</li> <li>• “tfe Function Replaced by tfestimate” on page 41</li> <li>• “csd Function Replaced by cpsd” on page 42</li> <li>• “cohere Function Replaced by mscohere” on page 42</li> </ul>