Filter Design Wizard (FIWIZ)
User Manual

\[ x_n \rightarrow + \rightarrow z^{-1} \rightarrow -b_1 \rightarrow a_1 \rightarrow + \rightarrow y_n \]

\[ x_n \rightarrow + \rightarrow z^{-1} \rightarrow -b_2 \rightarrow a_2 \rightarrow + \rightarrow y_n \]

\[ x_n \rightarrow + \rightarrow z^{-1} \rightarrow -b_M \rightarrow a_M \rightarrow + \rightarrow y_n \]
1. License Agreement

You should carefully read the following terms and conditions before using this software.

Registered Version

One registered copy of Fiwiz may either be used by a single person who uses the software personally on one or more computers, or installed on a single workstation used nonsimultaneously by multiple people, but not both.

You may access the registered version of Fiwiz through a network, provided that you have obtained individual licenses for the software covering all workstations that will access the software through the network. For instance, if 8 different workstations will access Fiwiz on the network, each workstation must have its own Fiwiz license, regardless of whether they use Fiwiz at different times or concurrently.

Disclaimer of Warranty

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OR MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program.

ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.
2. Glossary

dB Decibel
DE Differential Evolution
FIR Finite Impulse Response
GUI Graphical User Interface
IIR Infinite Impulse Response
JRE Java Runtime Environment

JAVA is a registered trademark of SUN Microsystems Inc.
MATLAB is a registered trademark of The Math Works Inc.
WINDOWS is a registered trademark of Microsoft Corporation

3. Installation Instructions

These installation instructions are given in terms of WINDOWS ® based computers. Since Fiwiz is written in Java it may run on any platform that supports the Java virtual machine, and the installation has to occur accordingly.

1) Fiwiz comes packaged in compressed .zip format, so you have to unpack it first. Do so in a directory named, for example, C:\Fiwiz. In C:\Fiwiz\Fiwiz you should then find the following directory tree:

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td></td>
<td>File Folder</td>
</tr>
<tr>
<td>de</td>
<td></td>
<td>File Folder</td>
</tr>
<tr>
<td>panel</td>
<td></td>
<td>File Folder</td>
</tr>
<tr>
<td>plot</td>
<td></td>
<td>File Folder</td>
</tr>
<tr>
<td>problem</td>
<td></td>
<td>File Folder</td>
</tr>
<tr>
<td>ptplot</td>
<td></td>
<td>File Folder</td>
</tr>
<tr>
<td>screen</td>
<td></td>
<td>File Folder</td>
</tr>
<tr>
<td>FIWIZ.class</td>
<td>1KB</td>
<td>Java Class File</td>
</tr>
</tbody>
</table>

Figure 3.1-1: Directory tree for Fiwiz.

Fiwiz.class is the class containing the main() method, and hence Fiwiz.class has to be called by Java to let the application run.

2) In order to do this get a Java 1.1.x runtime environment (JRE), for example, from http://www.javasoft.com or from http://www.icsi.berkeley.edu/~storn/fiwiz.html and install it.
on your computer if you haven’t done so already. Note: If you want to run Fiwiz from a batch file later on, make sure that you install the JRE in a path that exhibits directory names with a maximum of 8 characters. This limitation is due to DOS.

3) Let’s assume the Java runtime environment resides in C:\jre_1_1\bin, then from a DOS command prompt you can run Fiwiz by typing

C:\jre_1_1\bin\jre -cp C:\Fiwiz Fiwiz
4. Introduction
FIWIZ is a constraint based design program for IIR as well as FIR digital filters which is geared towards features which are difficult if at all to find in other filter design programs. The highlights of FIWIZ’s capabilities are:

- **Arbitrary magnitude constraints**
  This feature allows for multiband filters, differentiators, hilbert filters, sinc compensated filters and others.

- **Arbitrary group delay constraints**
  Applications are mainly classical lowpass, highpass, bandpass, and bandstop filters which should exhibit approximately linear phase in the passband(s) but not necessarily in the stopband(s). This way the filter degree can often be reduced considerably compared to exactly linear phase FIR filters.

- **Minimum phase filters**
  Some applications don’t require any specific phase response, and hence the filter degree can be minimized by using minimum phase filters.

- **Filter design with quantized coefficients**
  FIWIZ allows to include coefficient quantization in the filter design, i.e. quantization is incorporated into the design as opposed to quantizing the coefficients after the filter has been designed with high precision coefficients. The filter structures currently supported are the direct forms 1 and 2 [Mit93].

Some other convenient features of FIWIZ are:

- **Linear phase filters**
  FIWIZ is able to design linear phase filters of up to 200 taps.

- **Output of poles and zeroes**
  The results file of FIWIZ contains not only the filter coefficients of the direct form 1 (or 2) [Mit93] but also the pole and zero radii as well as angles. The poles and zeroes refer to the non-quantized filter coefficients.

- **MATLAB® friendly output format**
  FIWIZ’s output can be directly posted on to MATLAB’s ® command line interface for further analysis.

- **Storage and retrieval of configuration files**
  The settings of constraints and design parameters can be stored and retrieved so that there remains only little retyping if a previous filter design shall be altered.

- **Platform independence through JAVA® technology**
  FIWIZ has been written completely in JAVA® and hence it runs on any platform which supports the JAVA® virtual machine.

- **Wizard based approach**
  FIWIZ’s wizard based approach makes using FIWIZ almost self-explanatory. The sequence of operations is evident.
5. Usage of FIWIZ

5.1 Startup Screen

Figure 5.1-1 shows the startup screen of FIWIZ. Provides the possibility to load a properly formatted configuration file by which all important design parameters of a previous filter design can be loaded. These design parameters show up in the following screens. Whenever you want to save a certain parameter setting during a filter design session you have to go (back) to the startup screen and save the configuration using the Save As button.

The results of your filter design will be poles and zeroes as well as the filter coefficients for the direct form 1 (or 2) of a digital filter. In the results file textfield you can type in the name of the file which you want your results to be stored in. The default filename is Fiwiz.out.

With the next button you can proceed to the next screen, in this case the magnitude screen. The back button is deactivated in the startup screen since there exists no previous screen. The exit button serves to exit FIWIZ immediately without doing any save operation. The
“About” check box provides information about FIWIZ’s origin and the current version number.

### 5.2 Magnitude Screen

The magnitude screen shown in Figure 5.2-1 serves to define magnitude constraints in dB for the filter under design. The constraints list shows the currently active values for the upper and lower magnitude constraints. Whether the list for the upper or the one for the lower constraints is shown depends on the state of the radio buttons.

![Magnitude Screen](image)

**Figure 5.2-1: Magnitude Screen.**

Thanks to the graphics capabilities provided by PtPlot [PtPlot00] the entire constraints can also be viewed graphically as shown in Figure 5.2-2. The graphics can be activated by using the checkbox and can be helpful when checking the list entries for correctness.

In order to add a new constraint to the list the add button must be pressed. To modify an existing entry the edit button must be used. For the deletion of an entry go to the appropriate line in the list and press the delete button.
Figure 5.2-2: The magnitude constraints can also be viewed graphically.

Whenever the add or edit button is used the magnitude screen will change appearance according to Figure 5.2-3.

Figure 5.2-3: Magnitude screen after pressing either the add or the edit button.

In the omega text field the normalized frequency
\[ \Omega = \frac{\omega}{\omega_s} = \frac{2\pi \cdot f}{2\pi \cdot f_s} \in [0, 0.5] \]

can be entered. In this context \( f \) denotes the natural frequency and \( f_s \) represents the sampling frequency. Text field \( 2.12 \) takes the constraint value in dB. If you decide to move your constraint to the constraints list simply hit the apply button \( 2.13 \), otherwise hit the close button \( 2.14 \). Note that no frequency value may appear twice in the constraints list, i.e. strictly vertical lines in the constraints graph are not possible. Of course, the difference between two adjacent frequency values in the list can be made very small in order to approximate a vertical line in the constraints graph.

Since all points in a constraints list are connected by straight lines arbitrary constraint shapes can be generated. Note, however, that it is prudent to keep the constraints list as short as possible since a long list will lead to longer filter design times. The longer design time results from the increased number of points along the frequency axis that have to be checked by FIWIZ.

As a convenience a dB calculator using the radio buttons \( 2.8 \) and \( 2.9 \) has been added to the magnitude screen so linear magnitude values can easily be transformed into dB values and vice versa.

The next button will lead to the group delay screen.

### 5.3 Group Delay Screen

The group delay shown in Figure 5.3-1 allows you to specify group delay constraints. Using group delay constraints for IIR filter design can be an interesting alternative to linear-phase FIR filters especially if the filter degree shall be kept low and quasi-linearity of the phase is required only in the passband(s) of a filter.
Figure 5.3-1: Group delay screen.

The operation of the group delay screen works just the way it worked for the magnitude screen. The constraints are provided in normalized time

\[ T = \frac{t}{t_s} = t \cdot f_s \]

In contrast to magnitude constraints the group delay constraints will be floating, i.e. an arbitrary but constant group delay \( T_0 \) might be added to the given constraints by the filter design procedure as indicated in Figure 5.3-2.
Figure 5.3-2: The filter design procedure has the freedom to shift the group delay tolerance scheme along the ordinate.

5.4 Filter Realization Screen

The filter realization screen allows for the exact specification of the transfer function

\[ H(z) = A_0 \frac{\prod_{n=0}^{N-1}(z - z_{0n})}{\prod_{m=0}^{M-1}(z - z_{pm})}, \quad z = e^{j\Omega}, \quad j = \sqrt{-1} \]

FIWIZ takes care that all zeroes \( z_{0n} \) and all poles \( z_{pm} \) are either real or have a conjugate complex partner.

- 4.1 takes the number of zeroes which are not subject to any phase constraint,
- 4.2 takes the number of poles, and
- 4.3 sets the number of linear-phase zero pairs, i.e. for the latter the corresponding filter degree is always twice the number entered. The following design rules should be adhered to:
Figure 5.4-1: The filter realization screen defines the makeup of the filter transfer function.

1) When a minimum phase filter shall be designed the entry in \(4.3\) must be zero, and no group delay constraints shall be chosen. In this case an arbitrary phase filter and a minimum phase filter will be designed and their values printed out. Only the arbitrary phase filter will, however, be shown in the plots.

2) In case of a linear phase FIR filter design \(4.1\) and \(4.2\) should be set to zero.

3) In case of an IIR filter design with almost constant delay in the passband(s) it is advisable to not only use entries in \(4.1\) and \(4.2\) but also choose some linear phase zero pairs in \(4.3\). The linear phase zero pairs facilitate the design since some linearity is already introduced by design.

Note that the maximum entry in any of the text fields of the filter realization screen is currently limited to 100.

Choice control \(4.4\) takes the number of bits (including the sign bit) which all coefficients of the filter structure shall be restricted to. Note that this information is incorporated into the design procedure and is not just done after a design with high precision coefficients.

Choice control \(4.5\) allows to choose the filter structure defining the set of coefficients. Currently only the direct form structures (1 and 2) are supported.
5.5 Design Control Screen

The next screen is the design control screen which is depicted in Figure 5.5-1. Fiwiz employs a genetic algorithm called Differential Evolution or briefly DE [Sto00] for filter design. The global optimization capability and flexibility of DE allows for much freedom with respect to the constraints in magnitude and group delay. In quite a few cases Fiwiz allows you to design filters which are very difficult if not impossible to design even with expensive filter design tools. This flexibility and freedom have to be paid for by increased filter design times compared to traditional filter design methods. In addition, as with every genetic optimizer, convergence cannot be guaranteed. However, DE usually does a very good job when it comes to filter design.

Since genetic algorithms are population based there is a text field which takes the number NP of population members. In contrast to most other genetic algorithms DE allows this number to be fairly small. In general

\[ NP = 2^* (\text{number of all poles} + \text{number of all zeroes}) \]

is a reasonable value to start with.

![Design Control Screen](image)

Figure 5.5-1: Design Control Screen.

The text field takes the number NP2 of population members for the refinement phase of the design procedure. As a general rule NP2 should be the higher the smaller the number of coefficient-bits is chosen. A good estimate for NP2 is
\[ NP2 = NP + 1000 \cdot \left( \frac{32 - \text{bits}}{32} \right) \]

The last control variable in the design process is the sampling density along the frequency axis. It is possible to choose from coarse, medium and high density where the default is coarse. A coarse sampling density tests the least number of points for compliance with the constraints chosen and hence the filter design proceeds faster as if medium of high were chosen. However, it might be that a coarse sampling grid misses out on some frequency values where the constrains are violated.

### 5.6 Design Process Screen

The last screen of Fiwiz is the design process screen shown in Figure 5.6-1. Beginning and end of the filter design can be activated with the Start/Stop-Button. 

Once start is pressed the next press will stop the design. The filter design can also be paused with the Pause/Resume-Button which also is a toggle button. The monitor values show the elapsed number of DE-generations, elapsed number of filter function evaluations, current best cost function value, and state of the optimizer respectively. When the filter is designed successfully the cost function value of the refinement phase will be smaller than 1.e-12 and the state will return to “Ready”. As stated before even a cost value of zero does not guarantee that the filter meets all the constraints if the sampling grid along the frequency axis was not chosen high enough.

If the cost value arrives at zero or the design process is stopped with the Start/Stop-Button the filter parameters achieved are printed into the results file specified in the startup screen of Figure 5.1-1. The printout is in a MATLAB friendly format, i.e. the results can be copied directly into MATLAB’s command line interface for further analysis. An online plot of magnitude, zeroes and poles, and group delay can be invoked with the checkboxes respectively. This may be done during the design process or after its completion.
The plotters make use of PtPlot [PtPlot00] from the University of Berkeley, CA and allow for zooming in and out. Zooming in is done by mouse-drawing a rectangular box from left upper to right lower corner around the region of interest. Zooming out is done in a reverse manner by drawing a rectangular box from a lower right to an upper left corner. Figure 5.6-2 and Figure 5.6-3 give an example of the plotting capabilities provided by Fiwiz.

Figure 5.6-2: Example plot of magnitude in normal and a zoomed in version.
6. Trouble Shooting

6.1 Misconvergence
As has been stated before Fiwiz doesn’t guarantee to converge although convergence is likely if a solution exists. If the latter is assured, possible remedies are:

1. Simply rerun the design in the design process screen. Although DE can handle local optima fairly well it still might get trapped in one once in a while.
2. Increase the number of population members. Too low a number might prevent that the parameter space is sampled thoroughly enough.
3. Especially in IIR-filter design where quasi-linearity of phase (i.e. quasi constant group delay) in the passbands is desired, it is often advantageous for the optimizer if some of the zeroes are from the lin-phase zero pairs section in the filter realization screen.

If you are unsure about the feasibility of your design project try to get some estimate of the filter degree. If that is impossible work your way up from lower to higher degree filters. Fiwiz will yield a best fit in the least squares sense, so even when Fiwiz can’t find a solution in the first few runs you get valuable information about the problem at hand.

6.2 Constraint-violation
If the filter design of Fiwiz converges but the resulting filter still violates the constraints set forth in the magnitude and/or group delay screens the sampling density in the design control screen has been chosen too low. If the violation still occurs with even the highest sampling density the
constraints have to be divided into more segments. Figure 6.2-1 provides an example where a constant magnitude constraint of x dB between the normalized frequencies $\Omega_1$ and $\Omega_2$ is altered as the sequence of two constraints of x dB between the frequencies $\Omega_1$ and $\Omega_3$ as well as $\Omega_3$ and $\Omega_2$ respectively. As Fiwiz applies its sampling density to each segment individually the procedure outlined increases the overall sampling density.

![Figure 6.2-1: Example for increasing the number of constraint segments.](image)

For the sake of shortening the design time it often is even advisable to resort to the technique shown in Figure 6.2-1 rather than switching to the next higher sampling density.

### 6.3 Instable IIR-filters
Fiwiz tries to prevent instable filter solutions, i.e. the poles should always lie within the unit circle of the Z-domain. In rare cases, however, it might happen that a few poles escape to the outside. Just rerun the design and most probably the next filter design is stable.

### 7. Examples
The following examples shall provide a better idea of Fiwiz’s capabilities and usage. The examples are by no means exhaustive.
7.1 Notch filter
7.2 IIR-Differentiator

Filter Realization
- Numerator degree: 2
- Denominator degree: 4
- Number of in-phase zero pairs: 0
- Coefficient Quantization (bit): 16
- Filter Structure: Direct Form

Design Control
- Design Strategy: Differentiation
- No. of population vectors: 30
- nlo (refinement phase): 60
- Sampling Granularity: coarse, medium, fine

Design Process
- Status update every: 10 Generations
- Generations: 239
- Evaluations: 7296
- Best cost: 6.9
- State: Ready

Magnitude Plot
- Frequency range: 0.0 to 0.5
- Magnitude scale: 10^2

Pole Zero Plot (no quant.)
- Pole locations

Group Delay Plot
- Group delay function

Copyright © by Rainer Storn
7.3 Multiband filter

Filter Realization
- Numerator degree: 12
- Denominator degree: 12
- Number of imaginary zeros pairs: 9
- Coefficient Quantization (bit): 12
- Filter Structure: Direct Form

Design Control
- Design Strategy: Differential Evolution
- No. of population vectors: 100
- Nio (refinement phase): 100
- Sampling Orthogonality: coarse, medium, fine

Design Process
- Start
- Status update every: 100 Generations
- Generation: 140
- Evaluations: 64150
- Best cost: 6.6
- State: Ready

Magnitude Plot
- Frequency range: 0.0 to 0.5

Pole Zero Plot (no quant.)
- Pole locations: -1.5, 0, 1.5
- Zero locations: -1.5, 0, 1.5

Group Delay Plot
- Delay range: 0.0 to 0.5
- Delay units:

Copyright © by Rainer Storn
7.4 Bandpass

Filter Realization
- Numerator degree: 11
- Denominator degree: 11
- Number of lfsr phase zero pairs: 0
- Coefficient Quantization (bit): 16
- Filter Structure: Direct Form

Design Control
- Design Strategy: Differential Evolution
- No. of population vectors: 30
- std. (refinement phase): 80

Design Process
- Status update every: 10 generations
- Generation: 5210
- Evaluations: 213708
- Best cost: 2.51623x10^-6
- State: Ready (refinement)

Magnitude Plot
- x10^2

Pole Zero Plot (no quant.)
- Group Delay Plot

Copyright © by Rainer Storn
7.5 Lowpass with quasi-constant group delay
7.6 Lowpass with constant group delay

Filter Realization

- Numerator degree: 0
- Denominator degree: 0
- Number of finite impulse response: 20
- Coefficient Quantization digit: 3
- Filter Structure: Direct Form

Design Control

- Design Strategy: Differential Evolution
- No. of population vectors: 30
- dto. refinement phase: 60
- Sampling Granularity: coarse, medium, fine

Magnitude Plot

- Generation: 10,440
- Evaluations: 339,360
- Best cost: 0.8
- Status: Ready

- Magnitude
- Zeroes and Poles
- Group Delay
8. References

