

APPLICATION NOTE

Creating Simple Filter Shapes

Scope and Overview

This application note provides a comprehensive overview of the creation of filter files for use with the WaveShaper.

This application note covers the filter function, a function describing the complex change in wave amplitude; and attenuation and phase, describing the change in wave intensity and phase. These two different representations of optical filters are described and compared.

The User Configurable Filter (*.ucf) and WaveShaper Preset (*.wsp) file formats are introduced and contrasted. A set of easy-to-follow instructions are given on creating these filter files in Microsoft Excel, LabVIEW 8.5 and MATLAB. They can also be freely modified to accommodate your own filter designs.

Applicable Documents

- WaveShaper Manual
- Latest LabVIEW Driver Set

1. Filter Representation

An optical filter can be represented in two main forms:

1.1 The Filter Function

The filter function is an intuitive representation of a filter and describes how the wave *amplitude* changes after filtering. Consider a filter described by the filter function $F(f)$. When an input wave with frequency dependent amplitude $E_{in}(f)$ is filtered, its output amplitude is given by $F(f)E_{in}(f)$. In general, $F(f)$ is a complex number with a modulus describing the change in amplitude and a complex angle describing the change in phase.

1.2 Attenuation and Phase

Alternatively, a filter can be described by its attenuation and phase change. Unlike the filter function mentioned previously, attenuation describes the change in optical *intensity*, noting that wave intensity is proportional to the square of wave amplitude. Attenuation can be represented on a log scale, in dB instead of a ratio. Phase information must also be present in order to fully describe the filter behaviour.

WaveShaper filter profiles are represented in terms of attenuation and phase. Eq. 1 and Eq. 2 can be used convert a filter function into attenuation and phase:

$$A(f) = -20\log_{10}(|F(f)|) \quad \dots (1)$$

$$\phi(f) = \arg(F(f)) \quad \dots (2)$$

for attenuation $A(f)$, and phase change $\phi(f)$. Eq.1 contrasts from the standard dB equation, i.e. $10\log_{10}(F)$ for the following reasons: The factor of 2 arises from squaring F to convert amplitude coefficients to intensity coefficients. The negative sign appears because attenuation is a measure of loss rather than gain e.g. a positive attenuation represents a reduction in optical intensity.

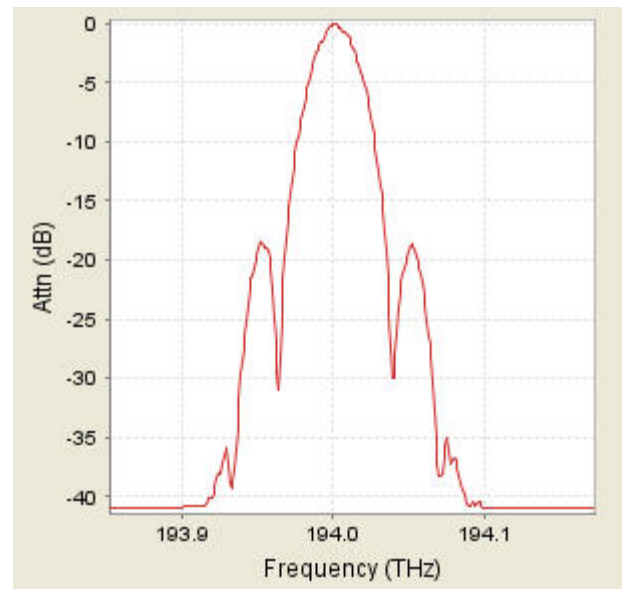


Figure 1: An example UCF filter of a DPSK 40Gbps modulation spectrum set with a center frequency of 194.000 THz User Configurable Filter (*.ucf)

1.3 Filter File Formats

The two main filter file formats supported by the WaveShaper are the User Configurable Filter (*.ucf) and WaveShaper Preset (*.wsp) files. WaveShaper filters can be

APPLICATION NOTE: Simple Filter Shapes

stored using both files, however choosing which file format to use will depend on its application.

1.3.1 User Configurable Filter (*.ucf) Files

The ability to load and then easily control user configured filters is key to the operation of the WaveShaper. This is enabled through the User Configurable Filter or UCF, designed specifically for use with WaveManager. A user's custom filter shape can experience the same level of configurability found with the band-pass, band-stop and Gaussian filter shapes without the need to regenerate new *.wsp files. The UCF file format provides ease of control over frequency centre, output port, attenuation, and phase.

The *.ucf file is a tab delimited text file with three columns: Frequency **Offset** (THz), Attenuation (dB) and Phase (Radians). The frequency offset data must contain exactly one entry for 0 and it is also recommended that the frequency offset data be resolved down to 1 GHz. An example *.ucf file looks like the following table:

-3.000	318.832	3.142
-2.999	69.545	3.142
...
0.000	0.000	0.000
...
2.999	69.545	3.142
3.000	253.153	3.142

An example UCF filter as seen in WaveManager is shown in Figure 1.

1.3.2 WaveShaper Preset (*.wsp) Files

The WaveShaper Preset (*.wsp) file is designed to store a full representation of the WaveShaper profile across all frequencies and ports. For this reason, WSP files are best used for complicated filters spanning multiple ports.

The format of a .wsp file is a tab delimited text file with four columns: **Absolute** Frequency (THz), Attenuation (dB), Phase (Radians) and Port. There is an important difference between the frequency column in the *.ucf file compared to the *.wsp file: the former represents frequency *offset*, while the latter represents *absolute* frequency. The frequency column must begin and finish with the start and stop frequencies specified for the WaveShaper and must also be specified down to a resolution of 1 GHz. A typical WSP profile can be seen in the table below:

191.250	2.0	3.14	2
191.251	2.1	3.15	2
191.252	2.2	3.16	2
...
...
196.273	6.2	3.16	4
...
196.274	7.2	3.15	4
196.275	8.2	3.14	4

An example WSP profile as seen in WaveManager is shown in Figure 2. It was constructed by combining a series of UCF filters set to different output ports.

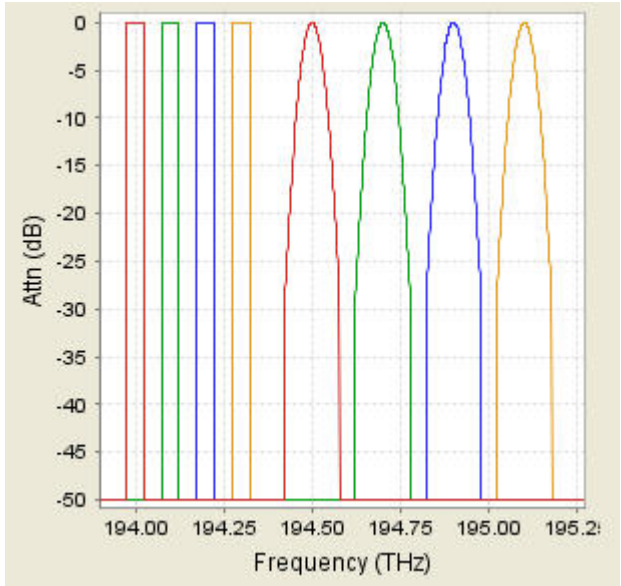


Figure 2: An example WSP filter. This filter was constructed from a series of separate UCF filters

1.3.3 Frequency Interpolation

The interpolation in the WaveShaper software chooses attenuation and phase values corresponding to the frequency value that is nearest to the respective frequency in the WaveShaper ("nearest neighbour interpolation"). This idea is shown graphically in Figure 3.

2. Creating Filters Using Microsoft Excel

2.1 Creating a User Configurable Filter (*.ucf)

Microsoft Excel is used to construct a User Configurable Filter (*.ucf) file in the following example. The filter constructed is a simple *sinc* ($=\sin(x)/x$) function with the

APPLICATION NOTE: Simple Filter Shapes

special feature that any delta-like input pulse is temporally shaped into a square-like output pulse.

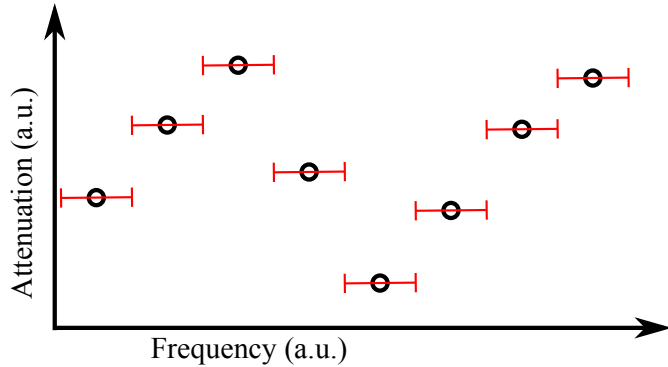


Figure 3: A graphical representation of the nearest neighbour interpolation used internally by the WaveShaper. The black circles show the filter data as represented by the UCF and WSP filter files. The red lines highlight how the WaveShaper interpolates for attenuation (and phase) for its own internal frequencies.

2.1.1 The Sinc Filter Function

With its control over attenuation and phase in the frequency domain, the WaveShaper can perform temporal pulse shaping. That is, an input pulse with a known temporal and spectral profile can be filtered in such a way to produce a given output temporal pulse. Consider an input pulse $E_{in}(t)$ entering the WaveShaper. The desired output pulse $E_{out}(t)$ is chosen to be the rectangular function modulated by a frequency f_0 .

$$\text{rect}(t) = \begin{cases} 1, & |t| < 1/2 \\ \frac{1}{2}, & |t| = 1/2 \\ 0, & |t| > 1/2 \end{cases} \quad \dots (3)$$

$$E_{out} = \text{rect}(t/\tau_w) e^{i2\pi f_0 t} \quad \dots (4)$$

where τ_w is the duration of the rectangular pulse. The WaveShaper performs filtering in the frequency domain, and can be described by the following:

$$\mathcal{F}\{E_{out}\} = F(f)\mathcal{F}\{E_{in}\} \quad \dots (5)$$

where $F(f)$ is the filter function. This function can be determined via a simple division:

$$F(f) = \frac{\mathcal{F}\{E_{out}\}}{\mathcal{F}\{E_{in}\}} \quad \dots (6)$$

Most input pulses are sufficiently small enough that they can be approximated ideally as a delta input pulse

$E_{in}(t) = \delta(t)$. The filter function is then simply the Fourier transform of the desired output pulse:

$$\begin{aligned} F(f) &= \mathcal{F}\{E_{out}\} && \dots (7) \\ &= \mathcal{F}\{\text{rect}(t/\tau_w) e^{i2\pi f_0 t}\} \\ &= \text{sinc}[\tau_w(f - f_0)] \end{aligned}$$

$$\text{sinc}(f) \equiv \frac{\sin(\pi f)}{\pi f} \quad \dots (8)$$

2.1.2 Microsoft Excel Procedure

The following instructions outline how to create a *.ucf file of the sinc function described in Section 2.1.1. The UCF file format is composed of 3 tab delimited columns: Frequency Offset (THz), Attenuation (dB) and Phase (Radians). More information about the UCF file format can be found in the WaveShaper User Manual.

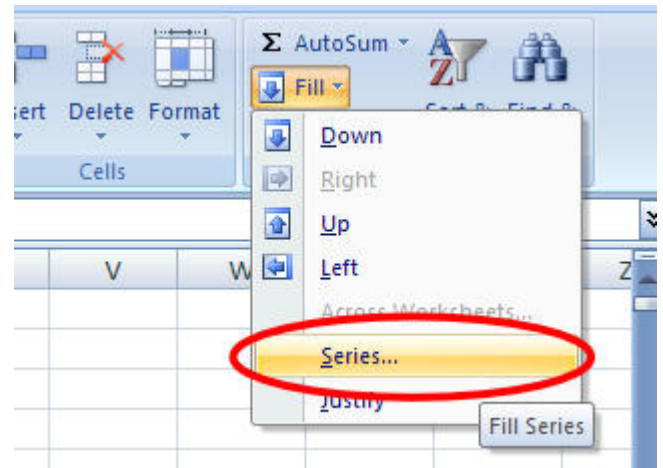


Figure 4: Navigating to the Series window

Step 1: Setting up and creating the Frequency Offset Column

Open up Microsoft Excel. When Excel has loaded, save a copy of a blank spreadsheet with the name 'Sinc Filter.xlsx'.

The first column to construct is the Frequency Offset, which is equivalent to $f - f_0$ in Eq. 7. In this example, the frequency offset is defined from -3 to 3 THz in 0.001 THz steps.

- Enter the "-3" into cell A1.
- Select the entire column by clicking the A column heading.
- Select *Fill* -> *Series...* from the Editing sub-tab of the Home Tab (Figure 4).

APPLICATION NOTE: Simple Filter Shapes

- Enter *Step value* and *Stop value* 0.001, and 3 respectively (Figure 5). Click OK to create the Frequency Offset column.

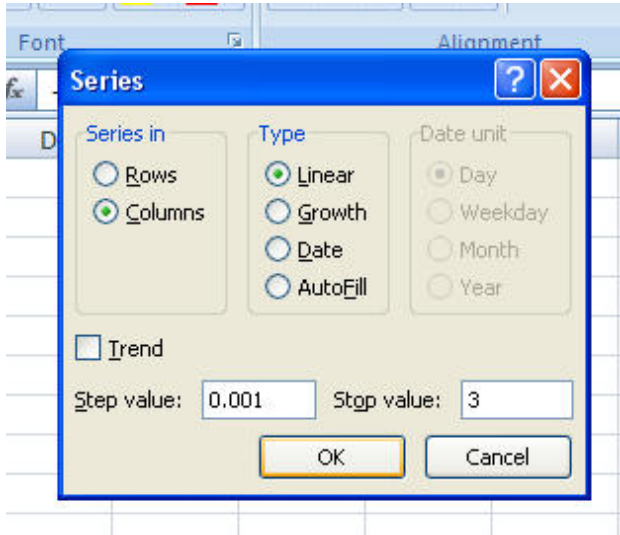


Figure 5: The Series window

Step 2: Creating the Filter Function Column

The filter function given by Eq. 7 will be entered into column B. Here Eq. 8 is used since the sinc function is not provided in Excel.

- Cell M1 will be used to store the rectangular width parameter τ_w . Enter 20 into the M1 Cell.
- Click B1, and enter (without quotes) the following into the formula bar:
- $=\sin(\$M\$1*PI()*A1)/(\$M\$1*PI()*A1)$
- Click the B1 cell and double click the bottom right black square. This will auto-fill the remaining rows.
- Excel cannot handle the singularity that occurs at $\text{sinc}(0)$. Navigate to cell B3001 and change its value to 1.

Step 3: Converting to Attenuation and Phase

The filter function must be converted into attenuation and phase values appropriate for the WaveShaper. This is achieved using Eq. 1 and Eq. 2.

- Click C1 and enter into the formula bar: $=-20*\log(\text{abs}(B1))$. Auto-fill the remaining rows by double clicking the bottom right corner of the C1 cell.
- Enter the following into the D1 cell: $=\text{IMARGUMENT}(B1)$ and auto-fill the remaining rows.

Step 4: Transferring to a new spreadsheet

Now that all columns have been constructed, they can now be copied into a second spreadsheet for export.

- Copy the entire A column.
- Create a new spreadsheet and paste the copied data into the first column. The data must be pasted in by value through the Paste Special... menu.
- Navigate to the first spreadsheet and copy both columns C and D. Paste them into the second and third columns of the new spreadsheet, again making sure to paste by value only.

Step 5: Adjusting number format

It is recommended to change the number format to show values accurate to 3 decimal places.

- Select the entire data by clicking the top left corner of the worksheet.
- Right click the selected cells and choose Format Cells...
- Select 'Number' in Category, and change decimal places to 3. Click OK.

Step 6: Exporting

Now the data is ready to be exported.

- Make sure the second spreadsheet is open, and select Save As -> Other Formats from the Office button.
- Choose text (Tab delimited) (*.txt) from the Save as type dropdown. Name your file, and click save. Click OK and then Yes to two popup messages to correctly export.
- This newly created file must be renamed to have the .ucf file extension. Closing the Excel Spreadsheet allows the file to be renamed.

Now that the .ucf file is created, it is ready to be loaded into the WaveManager. Note that the modulation frequency f_0 is equivalent to the Center frequency in WaveManager.

2.2 Creating a WaveShaper Preset Filter (*.wsp)

In this example, Microsoft Excel is used to create a *.wsp file featuring four 50 GHz bandpass channels sent to ports 1 through to 4.

Step 1: Setting up and creating the Absolute Frequency Column.

Open up a new Microsoft Excel spreadsheet and save it with the name 'WSPChannels.xlsx'. The first column to construct is

APPLICATION NOTE: Simple Filter Shapes

the frequency column. It is constructed by following these steps:

- Click the A1 Cell and enter in 'Absolute Frequency (THz)'.
- In the A2 Cell, enter in the start frequency of your WaveShaper in THz. In this example, the start frequency was 191.25.
- Select the entire A column by clicking the A column heading. Hold the Ctrl key and click the A2 Cell.
- Select Fill -> Series... from the Editing sub-tab of the Home Tab (Figure 4).
- Enter 0.001 for the Step value and enter in the stop frequency for your WaveShaper for the Stop value. In this example, this was 196.275. Click OK to create the Frequency column.

Step 2: Creating the Attenuation and Phase Columns

The next columns to create will store the attenuation and phase data. In this example, the channels will be controlled by port number, where a port number of 0 corresponds to blocking a particular frequency. With this in mind, the attenuation and phase columns may be set to zero for the entire frequency range.

- Enter 'Attenuation (dB)' and 'Phase (Radians)' into the B1 and C1 cells respectively.
- Enter '0' into the B2 and C2 cells.
- Highlight the B2 and C2 cells and double-click the bottom-right corner of the selection, marked by the black box. This will auto-fill the rest of the columns.

Step 3: Creating the channel filters

The last column to create is the port column. In this example, four 50 GHz channel filters will be created and set to ports 1 through to 4 respectively. Any frequencies outside of a channel are set to port 0. These parameters are entered into Excel to make this example more versatile:

- Enter the following into the J1-N4 Cell range:

	Ch. 1	Ch.2	Ch.3	Ch.4
Center Frequency	194	194.1	194.2	194.3
Bandwidth	0.05	0.05	0.05	0.05
Port	1	2	3	4

Some logic needs to be performed in order to generate the channels.

- Enter 'Channel 1' into the D1 Cell.
- Enter ' $=K\$4*(ABS(\$A2-K\$2)<=K\$3/2)$ ' into the D2 Cell.
- Select both the D1 and D2 cell. Click and drag the bottom right black box of the selection to the G column.
- Select cells D2 to G2 and auto-fill the rest of the columns.

Finally, the port column needs to be created. It is simply the sum of channels 1 to 4.

- Enter 'Port' into the H1 Cell.
- Enter ' $=SUM(D2:G2)$ ' into the H2 Cell.
- Auto-fill the rest of the column.

Step 4: Exporting to a *.wsp file

The four required columns for a *.wsp file have been created and it's now time to export the data. However it must first be put into the correct format.

- Highlight and copy columns A, B, C and H. Paste this into a new spreadsheet.
- Right click on the A1 Cell and select Delete... Select Entire row and click OK. This will delete the heading row, which is not a part of the *.wsp file.

It is recommended to change the number format to show values accurate to 3 decimal places.

- Select the frequency column by clicking the A column heading.
- Right click the selected cells and choose Format Cells...
- Select 'Number' in Category, and change decimal places to 3. Click OK.

The data is now ready to be exported:

- Make sure the second spreadsheet is open, and select Save As -> Other Formats from the Office button.
- Choose text (Tab delimited) (*.txt) from the Save as type dropdown. Name your file, and click save. Click OK and then Yes to the two popup messages to correctly export.
- This newly created file must be renamed to have the .wsp file extension. Closing the Excel Spreadsheet allows the file to be renamed.

APPLICATION NOTE: Simple Filter Shapes

3. Creating a UCF Filter in LabVIEW 8.5

In this example, LabVIEW 8.5 is used to construct a Gaussian shaped User Configurable Filter (.ucf) for use in the WaveManager software.

3.1 The Gaussian Filter Function

The Gaussian filter function is given by the following form:

$$F(f) = e^{-a(f-f_0)^2} \quad \dots (9)$$

where $f - f_0$ is the frequency offset. As described in Section 1.2, the filter function must first be converted to attenuation (dB) and phase (radians) using Eq. 1 and Eq. 2:

$$A(f) = \frac{20a}{\ln 10} (f - f_0)^2 \quad \dots (10)$$

$$\phi(f) = 0 \quad \dots (11)$$

This shows that a Gaussian filter function results in a parabolic attenuation function. A more useful parameter can be provided by expressing the Gaussian coefficient a in terms of the full width half maximum (FWHM) of the attenuation profile:

$$a = \frac{2 \ln 2}{w^2}, \quad \dots (12)$$

where w is the FWHM of the attenuation profile. The Gaussian filter function can then be expressed by the following:

$$F(f) = e^{-2 \ln 2 \left(\frac{f-f_0}{w}\right)^2} \quad \dots (13)$$

3.2 LabVIEW Procedure

The following describes how to construct a User Configurable Filter (.ucf) file using LabVIEW 8.5. The UCF file format is composed of 3 tab delimited columns: Frequency Offset (THz), Attenuation (dB) and Phase (Radians). More information about the UCF file format can be found in the WaveShaper User Manual. These instructions will guide you through creating the UCF data as a numerical array, and then exporting to the *.ucf file. This instruction set makes use of the LabVIEW driver set. Before following these instructions, make sure that the driver set has been downloaded and extracted. The driver set can be downloaded from the Finisar website (<http://www.finisar.com/waveshaper>).

Step 1: Setting up LabVIEW

Open LabVIEW and create a new blank Virtual Instrument (VI). Save your VI into a location of your choice. Also open up the LabVIEW driver set library file *wsapi.lvlib*. This library contains a set of VIs that provide access to all the functions available in the C API. While creating the Gaussian UCF filter does not require using the API directly, it does make use of some additional VIs in the library used for writing the UCF data to disk.

Step 2: Creating the Frequency Offset Vector

The frequency offset vector, equivalent to $f - f_0$ in Eq.13, is created in this step. In this example, the frequency offset vector is defined to be from -3 THz to +3 THz in 0.001THz steps.

Navigate to the Block Diagram. The easiest way to create the frequency offset vector is through use of a while loop from Programming -> Structures -> While Loop. The Block Diagram schematic shown in Figure 6 shows a possible way of constructing the vector:

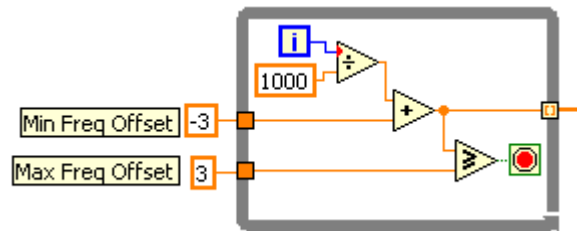


Figure 6: An example LabVIEW 'While Loop' used to create the Frequency Offset vector.

Step 3: Creating the Gaussian Filter Function

In this step, the Gaussian filter function from Eq. 13 is constructed using the *Formula VI*. Note that while this function is entirely real, the *Formula VI* cannot produce complex numbers. This means that producing complex filter functions must use another approach, e.g. using complex function VIs. The following describes how to create the filter function:

- Drag in the Formula VI, found in Mathematics -> Scripts & Formulas -> Formula. The Formula window should appear.
- Rename the X1 input label to 'freq', and the X2 input label to 'w'.
- Enter "exp(-2*ln(2)*(freq/w)**2)" into the Formula box and click OK.

APPLICATION NOTE: Simple Filter Shapes

- Wire the frequency offset output to the first Formula input labeled freq.
- Create a new DBL numeric control labeled 'FWHM (THz)' and wire it to the second Formula input labeled w. This can be done by right clicking the input and selecting Create -> Control.

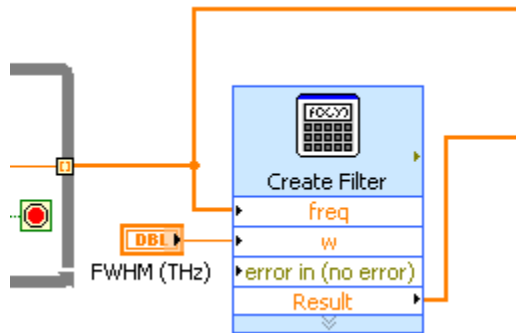


Figure 7: Using the Formula VI to create the filter function.

The result output from the *Formula VI* should now output the filter function. The Block Diagram should now look similar to Figure 7.

Step 4: Converting to Attenuation and Phase

The filter function was created in Step 3. In Step 4, this function is converted into attenuation (dB) and phase (radians). This is done using the labeled circuit shown in Figure 8, where the Filter Function wire is the result from the *Formula VI*.

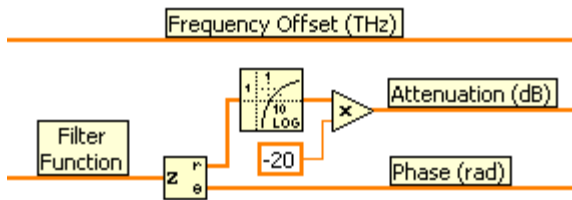


Figure 8: Converting the filter function to attenuation (dB) and phase (radians).

The contained VIs are found in the following locations:

Complex To Polar:

Mathematics -> Numeric -> Complex

Logarithm Base 10:

Mathematics -> Elementary -> Exponential

Multiply:

Mathematics -> Numeric

Note that the attenuation and phase profiles have been calculated in two separate steps: firstly creating the filter function, and then secondly converting to attenuation and phase. Alternatively, the first step can be skipped entirely by evaluating direct expressions for attenuation and phase, e.g. by using Eq. 10 and Eq. 11. However, by generating the filter function separately, the VI is more versatile, because it can be easily modified to evaluate different filters.

Step 5: Creating the *.ucf File

The *.ucf file is ready to be created now that all the needed data has been generated. In this step, two VIs which are part of the WaveShaper LabVIEW driver set are used: *BundleProfile.vi* and *CreateProfileString.vi*. The data, in its present form consists of three separate 1D arrays: frequency offset, attenuation and phase. Before the *.ucf file can be created, these arrays must be concatenated together. This can be done conveniently by using *BundleProfile.vi*.

- Drag in *BundleProfile.vi* from the *wsapi.lvlib* library.
- Wire the Frequency Offset, Attenuation and Phase arrays to their associated inputs of the *BundleProfile VI*.
- Wire a constant false Boolean to the 'Is WSP Profile' input of *BundleProfile*. This instructs *BundleProfile* that it is concatenating a 3-column UCF profile rather than the 4-column WSP profile. The port input can be left unwired.

The 2D numerical array outputted from *BundleProfile* can now be exported as the *.ucf file. This is done by using *CreateProfileString.vi*.

- Drag in *CreateProfileString.vi* from the *wsapi.lvlib* library.
- Wire the 2D data array into its associated input of *CreateProfileString*.
- Wire a constant true Boolean to the 'Save File' input of *CreateProfileString*. This tells *CreateProfileString* to pop up a file save dialogue allowing the profile string to be saved to disk.
- (Optional) Wire a string indicator to the output of the *CreateProfileString*.

The Block Diagram should now appear similar to Figure 9.

APPLICATION NOTE: Simple Filter Shapes

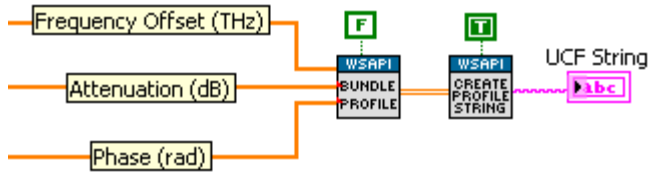


Figure 9: Saving the UCF data into a *.ucf file.

Step 6: Running the VI

The VI can now be executed.

- Navigate to the Front Panel and enter 0.1 into the FWHM control.
- Click the Run button.
- A Save File dialogue should appear. Make sure you save your *.ucf file with the .ucf file extension. Once this file exists inside the WaveManager ucf folder, it is ready to be used within WaveManager.

The full version of this VI is saved as *Example4_CreateGaussian.vi* in the *WaveShaperExamples.lvproj* LabVIEW Project. In addition to the functionality described above, *Example4_CreateGaussian.vi* also plots the resulting filter attenuation and phase.

4. Creating a UCF Profile in MATLAB

In this example, MATLAB is used to create a UCF profile of a hyperbolic secant filter for use with WaveManager. Some ultrashort pulses can have a spectral distribution envelope proportional to the hyperbolic secant function. E.g. some mode locked lasers, especially soliton lasers. The WaveShaper can then be used to emulate the spectral profile of these laser types.

4.1 The Hyperbolic Secant Filter Function

The hyperbolic secant filter function is given by Eq. 14

$$F(f) = \text{sech}[a(f - f_0)] \quad \dots (14)$$

$$\text{sech}(x) \equiv \frac{2}{e^x + e^{-x}}$$

where $f - f_0$ is frequency offset. Eq. 15 shows how the secant coefficient a can be expressed in terms of the FWHM of the attenuation function i.e. the width of the profile at 3dB.

$$a = \frac{2 \ln(\sqrt{2} + 1)}{w} \quad \dots (15)$$

$$\therefore F(f) = \text{sech} \left[2 \ln(\sqrt{2} + 1) \frac{f - f_0}{w} \right] \quad \dots (16)$$

The final form for the filter function is given by Eq. 16 where w is the FWHM of its associated attenuation profile.

4.2 MATLAB Procedure

The following describes how to construct a *.ucf file using MATLAB. The UCF file format is composed of 3 tab delimited columns: Frequency Offset (THz), Attenuation (dB) and Phase (Radians). More information about the UCF file format can be found in the WaveShaper User Manual.

Step 1: Setting up MATLAB

Open up MATLAB and create a new script file. Save this script file in your current MATLAB work directory and name it 'SechFilter.m'.

Step 2: Creating the Frequency Offset Vector

The frequency vector, equivalent to $f - f_0$ in Eq. 16 is first created. In this example it is defined from -3 THz to 3 THz in 0.001 THz steps. To create this vector, enter in the following script code:

```
wsFreqOffset = (-3:0.001:3)';
```

Step 3: Creating the Filter Function

In this step, the filter hyperbolic secant filter function vector from Eq. 16 is created with the following code:

```
w = 100e-3;
Filter = sech(2*log(sqrt(2)+1)*(wsFreqOffset/w));
```

The bandwidth of the profile is set to 100 GHz in this example. This is done by the first line above. Note that the units of wsFreqOffset are in THz, and so the units of w must also be in THz.

Step 4: Converting to Attenuation and Phase

Eq. 1 and Eq. 2 are used in this step to convert the filter function into attenuation and phase vectors using the following code:

```
wsAttn = -20*log10(abs(Filter));
wsPhase = angle(Filter);
```


APPLICATION NOTE: Simple Filter Shapes

Step 5: Creating the *.ucf File

At this stage, the code has been used to create the three vectors of data which constitute the *.ucf file. Before this data can be saved, it must first be concatenated into a 2D array:

```
UCFProfile = [wsFreqOffset, wsAttn, wsPhase];
```

Now the data is ready to be written to file by the following code:

```
dlmwrite('SechFilter-100GHz.ucf', UCFProfile,  
'delimiter', '\t', 'precision', '%.3f');
```

Make sure all the above code is inside your script file. Save the script and then run it by pressing F5, or typing SechFilter into the command window. This will create a *.ucf file named 'SechFilter-100GHz.ucf' in your current work directory. Once this file exists within the WaveManager ucf folder, it is ready to be used as a custom filter in WaveManager.

5. Creating a WSP Profile in LabVIEW 8.5

In this example, LabVIEW 8.5 is used to construct a Gaussian shaped WaveShaper Preset File (.wsp) for use in the WaveManager software and WaveShaper API.

5.1 The Gaussian Filter Function

The Gaussian filter function is given by the following form:

$$F(f) = e^{-a(f-f_0)^2} \quad \dots (17)$$

where f is absolute frequency and f_0 is the centre frequency. The filter function must first be converted to attenuation (dB) and phase (radians):

$$A(f) = \frac{20a}{\ln 10} (f - f_0)^2 \quad \dots (18)$$

$$\phi(f) = 0 \quad \dots (19)$$

This shows that a Gaussian filter function results in a parabolic attenuation function. A more useful parameter can be provided by expressing the Gaussian coefficient a in terms of the width of the attenuation profile at 3 dB:

$$a = \frac{2 \ln 2}{w^2}, \quad \dots (20)$$

where w is the width of the attenuation profile at 3 dB. The Gaussian filter function can then be expressed by the following:

$$F(f) = e^{-2 \ln 2 \left(\frac{f-f_0}{w}\right)^2} \quad \dots (21)$$

5.2 LabVIEW Procedure

The following describes how to construct a WaveShaper Preset File (*.wsp) using LabVIEW 8.5. The WSP file format is composed of 4 tab delimited columns: Absolute Frequency (THz), Attenuation (dB), Phase (Radians) and Port. More information about the WSP file format can be found in the WaveShaper User Manual. These instructions will guide you through creating the WSP data as a 2D numerical array, and then exporting to the *.wsp file. This instruction set makes use of the LabVIEW driver set. Before following these instructions, make sure that the driver set has been downloaded and extracted. The driver set can be downloaded from the [Finisar website](http://www.finisar.com/waveshaper) (<http://www.finisar.com/waveshaper>).

Step 1: Setting up LabVIEW

Open LabVIEW and create a new blank Virtual Instrument (VI). Save your VI into a location of your choice. Also open up the LabVIEW driver set library file *wsapi.lvlib*. This library contains a set of VIs that provide access to all the functions available in the C API. While creating the Gaussian WSP filter does not require using the API directly, it does make use of some additional VIs in the library used for writing the WSP data to disk.

Step 2: Creating the Absolute Frequency Vector

The absolute frequency vector, equivalent to f in Eq. 18, is created in this step. In this example, the absolute frequency vector is defined to be from 191.25 THz to 196.275 THz in 0.001 THz steps. This range is standard across all C-Band WaveShapers, and can be looked up in the WaveShaper User Manual at Section 9.3 *Optical Specifications*.

Navigate to the Block Diagram. The easiest way to create the absolute frequency vector is through use of a while loop from Programming -> Structures -> While Loop. The Block Diagram schematic shown in Figure 10 shows a possible way of constructing the vector:

APPLICATION NOTE: Simple Filter Shapes

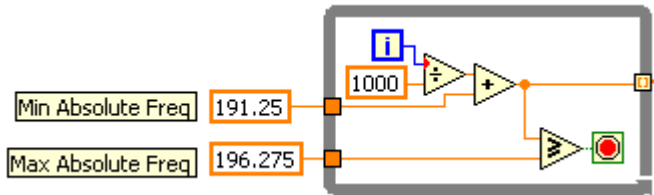


Figure 10: An example LabVIEW 'While Loop' used to create the Absolute Frequency vector.

Step 3: Creating the Gaussian Filter Function

In this step, the Gaussian filter function from Eq. 18 is constructed using the *Formula VI*. The centre frequency, f_0 , is given by a LabVIEW Control. The following describes how to create the filter function:

- Drag in the *Formula VI*, found in Mathematics -> Scripts & Formulas -> Formula. The Formula window should appear.
- Rename the X1 input label to 'freq', the X2 input label to 'w', and the X3 input label to 'centre'.
- Enter " $\exp(-2*\ln(2)*((\text{freq}-\text{centre})/\text{w})**2)$ " into the Formula box and click OK.
- Wire the absolute frequency output to the first Formula input labeled 'freq'.
- Create a new DBL numeric control labeled 'FWHM [THz]' and wire it to the second Formula input labeled 'w'. This can be done by right clicking the input and selecting Create -> Control.
- Create a second DBL numeric control labeled 'Central Frequency [THz]' and wire it to the third Formula input labeled 'centre'.

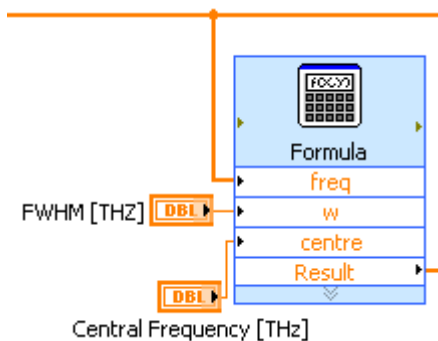


Figure 11: Using the *Formula VI* to create the filter function.

The result output from the *Formula VI* should now output the filter function. The Block Diagram should now look

similar to Figure 11: Using the *Formula VI* to create the filter function. Figure 11.

Step 4: Converting to Attenuation and Phase

The filter function was created in Step 3. In Step 4, this function is converted into attenuation (dB) and phase (radians). This is done using the labeled circuit shown in Figure 12, where the Filter Function wire is the result from the *Formula VI*.

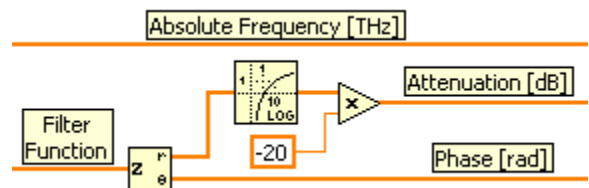


Figure 12: Converting the filter function to attenuation (dB) and phase (radians).

The contained VIs are found in the following locations:

Complex To Polar:

Mathematics -> Numeric -> Complex

Logarithm Base 10:

Mathematics -> Elementary -> Exponential

Multiply:

Mathematics -> Numeric

Note that the attenuation and phase profiles have been calculated in two separate steps: firstly creating the filter function, and then secondly converting to attenuation and phase. Alternatively, the first step can be skipped entirely by evaluating direct expressions for attenuation and phase, e.g. by using Eq. 18 and Eq. 19. However, by generating the filter function separately, the VI is more versatile, because it can be easily modified to evaluate different filters.

Step 5: Creating the Port Vector

Finally, this step creates the Port vector. The filter in this example directs its entire frequency range to a single port. This is achieved by creating a vector containing a constant value given by Port User Control.

- Create a For Loop.

- Wire the phase vector into the For Loop as an auto-indexed tunnel.

APPLICATION NOTE: Simple Filter Shapes

- Create an unsigned 8-bit integer control labeled 'Port' inside the For loop.
- Connect the Port control to the outside of the For Loop, making sure that it creates an auto-indexed tunnel.

These steps will create the port vector. The Block Diagram should now look similar to Figure 13.

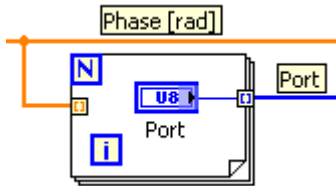


Figure 13: Creating the Port Vector.

Step 6: Creating the *.wsp File

The *.wsp file is ready to be created now that all the needed data has been generated. In this step, two VIs which are part of the WaveShaper LabVIEW driver set are used: *BundleProfile.vi* and *CreateProfileString.vi*. The data, in its present form consists of four separate 1D arrays: absolute frequency, attenuation, phase and port. Before the *.wsp file can be created, these arrays must be concatenated together. This can be done conveniently by using *BundleProfile.vi*.

- Drag in *BundleProfile.vi* from the *wsapi.lvlib* library.
- Wire the Absolute Frequency, Attenuation, Phase, and Port arrays to their associated inputs of the *BundleProfile* VI.

The 2D numerical array outputted from *BundleProfile* can now be exported as the *.wsp file. This is done by using *CreateProfileString.vi*.

- Drag in *CreateProfileString.vi* from the *wsapi.lvlib* library.
- Wire the 2D data array into its associated input of *CreateProfileString*.

- Wire a constant true Boolean to the 'Save File' input of *CreateProfileString*. This tells *CreateProfileString* to pop up a file save dialogue allowing the profile string to be saved to disk.
- (Optional) Wire a string indicator to the output of the *CreateProfileString*.

The Block Diagram should now appear similar to Figure 14.

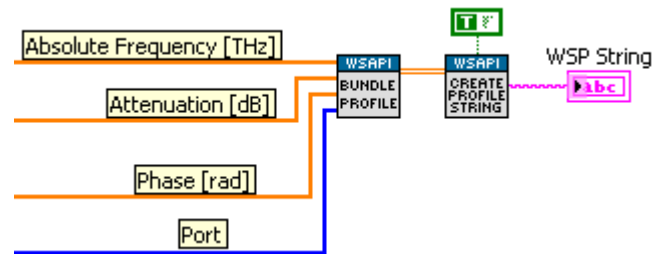


Figure 14: Saving the WSP data into a *.wsp file.

Step 6: Running the VI

The VI can now be executed.

- Navigate to the Front Panel and enter 0.1 into the FWHM control.
- Enter 1 into the Port Control.
- Click the Run button.
- A Save File dialogue should appear. Make sure you save your *.wsp file with the .wsp file extension.

Congratulations, you created a WSP filter using LabVIEW!