

Technical Report

FiberSim User Manual

Minnan Fei, Chidambaram Pavanasam and
Kenneth Demarest

ITTC-FY2004-TR-22737-01

January 2004

Project Sponsor:

Sprint Corporation

Contents

Chapter 1 FiberSim Overview

1.1 Background.....	1
1.2 FiberSim Source Code Organization.....	1
1.3 FiberSim Component Library.....	3

Chapter 2 Installation Guide

2.1 FiberSim for Windows.....	4
2.1.1 Installation Procedures.....	4
2.1.2 Uninstalling FiberSim.....	4

Chapter 3 Using FiberSim

3.1 FiberSim Overview.....	7
3.2 FiberSim Graphical User Interface.....	9
3.2.1 Menubar.....	9
3.2.2 Toolbar.....	12
3.2.3 Component Icons.....	12
3.2.4 Navigator Panel.....	14
3.2.5 Multiple Windows.....	15
3.3 Getting Started.....	16
3.3.1 Constructing Fiber Optic Network.....	18
3.3.2 Simulation Parameters.....	20
3.3.2.1 Input Bits.....	21
3.3.2.2 Padding Bits.....	21
3.3.2.3 Simulation Bandwidth.....	21
3.3.2.4 Simulation Options.....	21
3.3.3 Transmitter Section.....	22
3.3.3.1 Pulse Generator.....	22
3.3.3.2 Laser.....	24
3.3.3.3 Modulator.....	25
3.3.3.4 Multiplexer.....	25
3.3.4 Optical Fiber.....	26
3.3.4.1 Single-Mode Fibers.....	27
3.3.4.1.1 Basic Parameters.....	27
3.3.4.1.2 Dispersion Parameters.....	27
3.3.4.1.3 Nonlinearity Parameters.....	28
3.3.4.1.4 Simulation Methods.....	29
3.3.4.2 Raman Fiber.....	30
3.3.5 Optical Amplifier.....	31
3.3.6 Receiver Section.....	32
3.3.6.1 De-multiplexer.....	32

3.3.6.2 Photodiode.....	33
3.3.7 Filters.....	34
3.3.7.1 Electrical Filter	34
3.3.7.2 Optical Filter.....	34
3.3.8 Measurement Plots.....	35
3.3.8.1 Spectral Plots.....	35
3.3.8.2 Pulse Plot.....	38
3.3.8.3 Eye Diagram.....	39
3.3.9 Multiple Spans.....	40
3.3.10 User Specified Components.....	40
3.3.11 Running and Stopping Simulation.....	44
3.4 Q-value and Eye Penalty Calculation.....	46
3.5 Output File.....	47
3.6 Error Messages.....	48

Appendix

A Parameter Tables for Single-mode Fiber and Leaf Fiber.....	50
B Maintaining and Updating FiberSim Source Code.....	51

Chapter 1 FiberSim Overview

1.1 Background

FiberSim is a Wavelength Division Multiplexing Fiber Optic Digital Transmission System Simulator. Its purpose is to simulate the performance of WDM fiberoptic networks at the physical layer. Fibersim is capable of modeling a wide range of modulation formats, laser sources, and fiber types, along with a number of associated fiber and electrical system elements. Fibersim allows a modeler to specify systems using a user-friendly graphical interface.

The interface to the application is a graphical interface that allows the user to assemble a system by clicking component icons. Each component can then be individually configured through their individual dialog boxes. At any point in the system, whether in the electrical or optical domain, the user may insert a plotting component that allows them to view the state of the system at that point in various ways. The performance output views available include Eye Diagrams, Pulse Intensity plots, and Spectral Intensity plots. At the same time, Eye-Penalty and Bit-Error rates are calculated.

FiberSim was produced by the Lightwave Communication Lab, part of the Information & Telecommunication Technology Center (ITTC) at the University of Kansas. The original source code of FiberSim resides in the ITTC computer network and is maintained and updated through WinCVS (Control Version System).

1.2 FiberSim Source Code Organization

The source code is organized in WinCVS as follows:

fs	Main Directory, has the configure script for the make files.
--docs	Documents on various things: CVS, FiberSim background...etc.
--fibersim	Old version of the C++ code.
--fibersimRFF	Old version of the C++ code.
-- fibersimRFF.nff	Old version of the C++ code.
--fibersimSCM	Current working version of the C++ code.
--gui	Old version of the Java code.
--gui.nff	Old version of the Java code.
--gui.Horizon	Current working version of the Java GUI code.
--manual	Simple guide of how to use the GUI.
--icons	Contains the GIF images used in the GUI.

--install	Code used for the installation program.
--Makefiles	Contains commands for compiling.
--working	Contains executable files, created through compiling.

All of the above will be created when the code is checked out for the first time, except working directory, which will be created automatically when compiling of the source code is done.

Another directory that is of importance is the */projects/Fiberspec* directory. This directory is located on devnull.ittc.ukans.edu and must be mounted. The Fiberspec directory contains numerous runs in various directories, and probably need to be cleaned up a little bit once a while. This chart will make sense of the important directories and their contents.

<i>/projects/Fiberspec</i>	Backed up projects directory.
--CVS	Version control directories.
--demo83	Sprint Runs - These are important because sprint give us money
--fsimage	CD Contents. When creating a deliverable replace the executables and the Jar files present in these directories.
--arch	Directories for various architectures' executable.
--gui	Where the FiberSim GUI jar file is.
--install	Where the Install jar file is.
--jre	Directories for various architectures' JRE so that the installer will run
--jredist	Compressed JRE's for various architectures.
--rcs*	Older version control directories.
--wdm_pure	Original Matlab code that this project is based on
--*	Probably a run or something that needs to moved

More information about the class hierarchy/function structure/data flow diagram of the source code is saved in P:\Fiberspec\docs\html in ITTC network.

1.3 FiberSim Component Library

- **Optical Components**

Single-Mode Fiber: single or dual-polarization nonlinear equation model, including linear and nonlinear effects at choice: dispersion, SPM, XPM, FWM, Raman scattering, parametric gain, birefringence, and PMD

Raman Fiber: signals and pumps can co-propagate in the fiber at the same time. Stimulated Raman Scattering effect is taken into account during the simulation. Linear and nonlinear effects are same as common fiber

Source: Laser (single or multiple-channel)

Modulator: Ideal (output power is directly proportional to input voltage)

Filters: Bessel, Butterworth, Ideal, Notch, and User-defined

Amplifier: EDFA model

Photodiode: PIN

Multiplexer\Demultiplexer

- **Electrical Components**

Data Sources: NRZ/RZ, and PSK

Filters: Bessel, Butterworth, Ideal, Notch, and User-defined

- **Measurement Tool**

Measurement Plot: includes pulse plot (electrical domain), spectrum plot (optical domain), and eye diagram (electrical domain), also serves as Q-value/BER/Eye-penalty estimator

- **User-defined Components**

User-specified Components: electrical/optical filters, demultiplexer, and photodiode

Custom Components: electrical/optical filters, and demultiplexer

Chapter 2 Installation Guide

FiberSim is a software written by people working in the Lightwave Communication Lab of Information & Telecommunication Technology Center (ITTC) at the University of Kansas. The original source code of FiberSim is maintained and updated through WinCVS (control version system), a software allowing multiple people to view and modify the code together. The user may download CVS from the Internet.

Appendix B includes the information of installing & using CVS, and compiling & modifying the source code in great details.

2.1 FiberSim for Windows

2.1.1 System Requirements

This section defines the system minimum requirements to install FiberSim. Please check that your system matches all the requirements before installing FiberSim.

Hardware Requirements

- Pentium II
- 128 Mbytes RAM
- At least 5 Gbytes of free disk space
- CD-ROM drive
- Graphic display with a resolution of 1024x768 pixels or higher; color is optional but highly recommended
- Keyboard and two- or three-button mouse

Software Requirements

- Microsoft Windows NT/XP/2000/98/95 Server
- TCP/IP communication protocol

2.1.2 Installation Procedures

To run FiberSim successfully, the following files have to be included:

- Executable Files –generating the GUI and call the source code to perform simulation.




FiberSim.jar,




FiberSim.exe, and




fsloader.dll

- Channel Files  – taking care of the GUI dialog boxes

customdemux.cdf,	customelecfilter.cdf,	customoptfilter.cdf
demux.cdf,	elecfilter.cdf,	fiber.cdf
iterator_begin.cdf	iterator_end.cdf,	laser.cdf
modulator.cdf,	multiplexor.cdf,	nrz.cdf
optamplifier.cdf,	opticalfilter.cdf,	photodiode.cdf
plot.cdf,	psk.pdf,	ramanfiber.cdf
simparameters.cdf,	specificdemux.cdf,	specificelecfilter.cdf
specificopticalfilter.cdf,	specficphotodiode.cdf	

- Image Files  – displaying the component icons, saved in “icons” folder

BER.gif	chart1-16.gif	chart1-24.gif
Char2.gif	chart2-24.gif	chart3-16.gif
CustomDemux.gif	CustomElecFilter.gif	CustomOptFilter.gif
DCG.gif	delete.gif	Demux.gif
ElecFilter.gif	EyeDiagram.gif	EyeDiagram2.gif
Fiber.gif	FiberDialog.gif	FilterDialog.gif
fs.gif	fs.png	FSIcon.gif
full-screen.gif	IteratorEnd.gif	IteratorStart.gif
keys16.gif	keys24.gif	Laser.gif
LaserDialog.gif	left.gif	logo4.png
Modulator.gif	Mux.gif	MuxDialog.gif
myControl16.gif	myControl24.gif	myHelp16.gif
myHelp24.gif	MySave.gif	new.gif
new16.gif	note16.gif	note24.gif
note24-trans.gif	NRZ.gif	NRZDialog.gif
NRZDialog2.gif	NRZDialog3.gif	open.gif
open16.gif	open2.gif	OptAmp.gif
OptAmpDialog.gif	OptDetect.gif	OptFilter.gif
PhotoDiode.gif	plot.gif	PlotDialog.gif
PlotDialog2.gif	PlotDialog3.gif	PSK.gif
PulseIntensity.gif	RamanFiber.gif	right.gif
runCombo.gif	save.gif	save1.gif
save1-2.gif	save16.gif	SimParamDialog.gif
simulate16.gif	SimulateDialog.gif	SimulateDialog2.gif
SimulateDialog3.gif	SpecificDemux.gif	SpecificElecFilter.gif
SpecificFilterDialog.gif	SpecificPhotoDiode.gif	SpectralIntensity2.gif
TxFiber.gif	xbutton16.gif	

- FSP Files  – dealing with the Sweep function in source code

2variablesweep.fsp
init.fsp

example.fsp

example_spacing.fsp

All of the files above are saved in a disc – FiberSim disc, which will be used for installation of FiberSim.

To install FiberSim, please follow several simple procedures showing below:

- Insert the FiberSim disc into the floppy drive
- Open the disc from the floppy drive.
- Copy and paste all the files to a directory on the hard drive.
- Go to the directory where the files have been saved.



- Double-click on `FiberSim` (executable jar file).
- The FiberSim GUI window should pop up, which means the installation is successful.

2.1.3 Uninstalling FiberSim

Uninstalling FiberSim is very simple. First, choose “Start → Window Explorer”. When the window explorer window is open, go to the directory where the execution files are saved, right-click on it and choose “delete” from the pop-up menu. A message will appear and ask the user’s decision again, just click “Yes”. The software will be deleted from the computer.

Chapter 3 Using FiberSim

This chapter is meant as an introduction to the FiberSim simulation environment to a new user to learn the essentials and get started. After having reading this chapter, you will be able to build models on your own, run simulations, and see the effects of changing the various parameters.

3.1 FiberSim Overview

In order to setup a simulation, An optical network has to be constructed first and the parameters needs to be defined using the FiberSim’s Graphical User Interface, which is compliant to the Windows standard and easy to use. As shown in Figure 1, the menu bar and tool bar are showing at the left-higher corner. The left column is the navigator panel, which guides the user though simulation files, text outputs and different plots.

The blank area on the right is the place to build circuits. Users can pick from the drawing palette the component they wish to use, which is represented by a standard yellow symbol often used by other fiber optic softwares. The component icon will be displayed automatically in the blank area following a left-click. By double-clicking on each component, the user can define the values of its parameters. Once the optical system is designed, a typical window looks like:

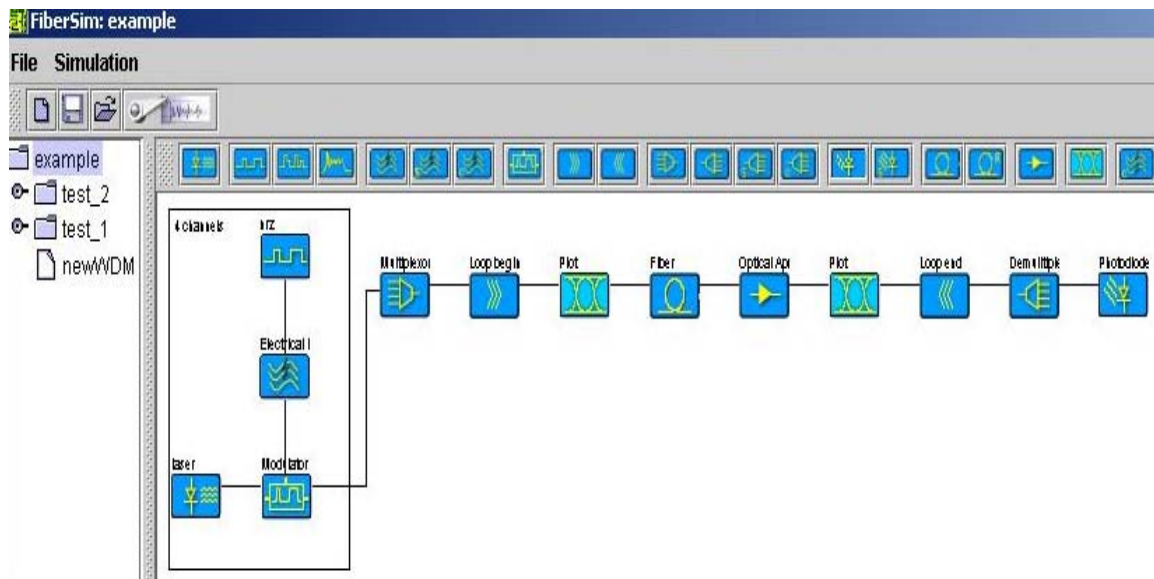


Figure 1 General View of FiberSim Environment

As soon as the numerical simulation is done, the results are available and they can be viewed by selecting the corresponding command in the navigator panel. The graphical results (such as eye-diagrams, spectra etc.) will be plotted and displayed by FiberSim, while all numerical results (such as Q calculation and eye penalty) are displayed as text.

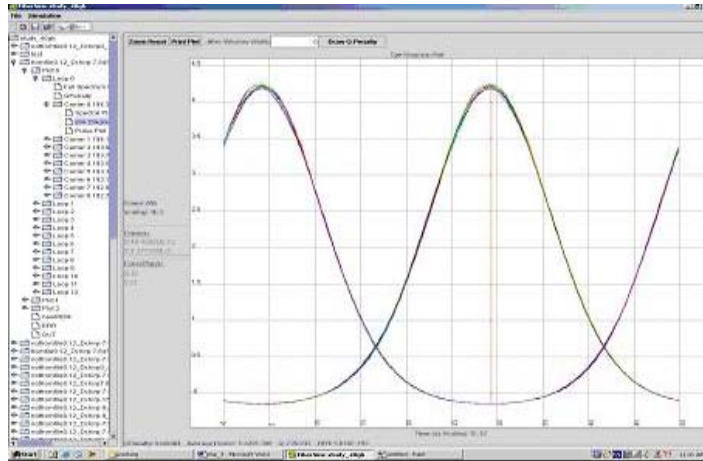


Figure 2 Eye Diagram

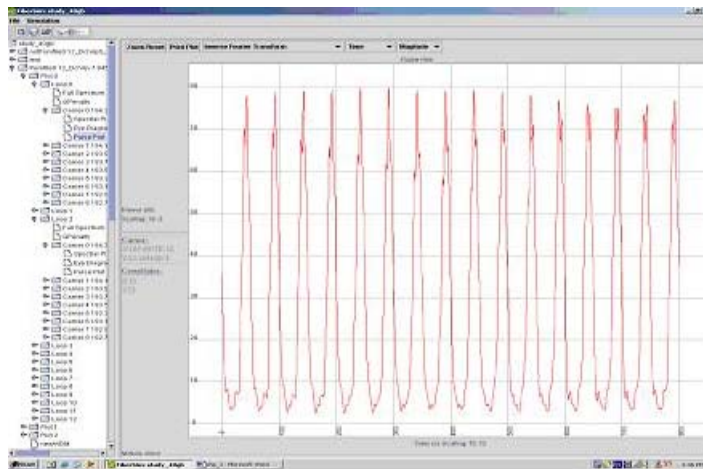


Figure 3 Pulse Plot

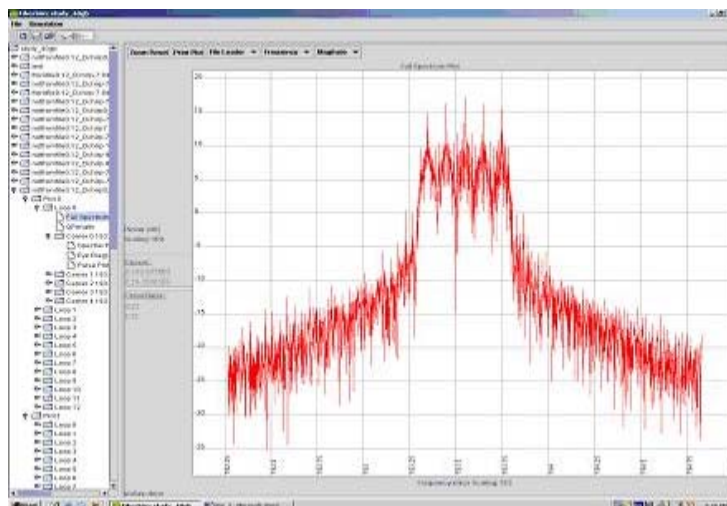


Figure 4 Spectral Plot

FiberSim has the capability of simulating long-haul WDM system with all types of nonlinear effects (such as dispersion, SPM, XPM, PMD etc.). All simulation components are based on a time-domain computation. The most complicated and important component—optical fiber, is simulated by using the standard Split-Step Fourier method to solve the nonlinear Schrodinger equation. Extremely long fiber links may be simulated on a large time window with excellent accuracy.

3.2 FiberSim Graphical User Interface

This section is a detailed guide for the input-out interface. Users familiar with PC software should have no difficulty in understanding how to work with FiberSim.

3.2.1 Menubar

FiberSim’s menubar provides two options: “File” and “Simulation”. File menu is used to control and edit the simulation projects. Simulation menu takes care the simulation issues.

3.2.1.1 File Menu

File menu has seven commands, which can be chosen by left-click on it.

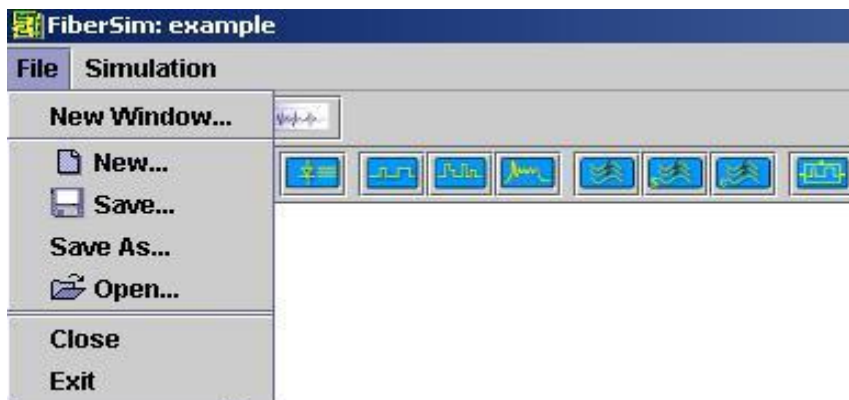




Figure 5 File Menu


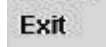
New Window... - It lets users work on several simulation projects at the same time. Every time click on it, one more window will pop up, then, click **Open...** to open the project wanted in each window.

Save As... - Whenever creating a new project and working on a test at the first time, the user may use “Save as” to save this test after constructing the circuit, then,

choose a directory and assign a name to it. Or, it saves the current project into a new file with a new name.

 - When modifying the same test, just hit it to save the newest version.

 - It allows the user to build a different project.

  - “Close” closes the present window users are working on. “Exit” means quitting the whole program—all of the FiberSim’s windows will be closed.

3.2.1.2 Simulation Menu

Simulation menu has three functions as shown in Figure 6:

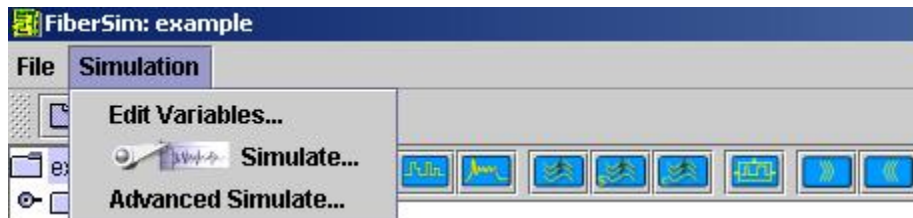



Figure 6 Simulation Menu

 - It defines global variables for dialog boxes used in FiberSim. Click on it, a window will pop up.

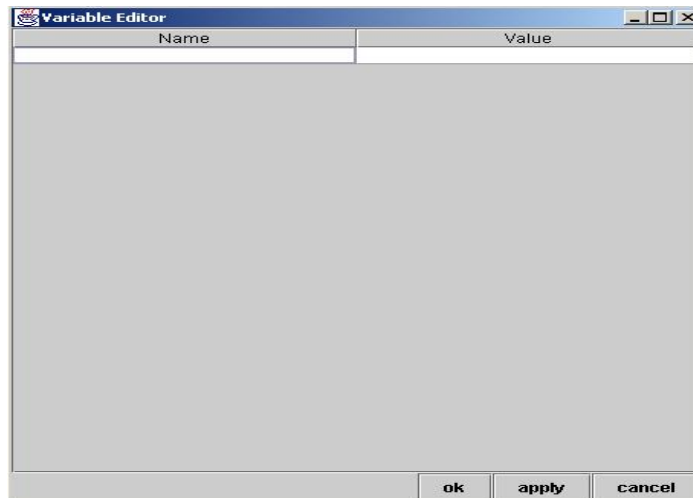


Figure 7 Edit Variable Window

If the user decides to define some global variable, they will need to input the global variable's name and value into the corresponding column. Wherever the global variable's value is needed, just fill its name into the parameter box. In this way, the user only need to edit the global variable in the "Variable Editor" window, instead of typing in the number in every place using the same value.

For example, if the laser power and the fiber length are 100mW and 100km, respectively. A global variable can be defined as:

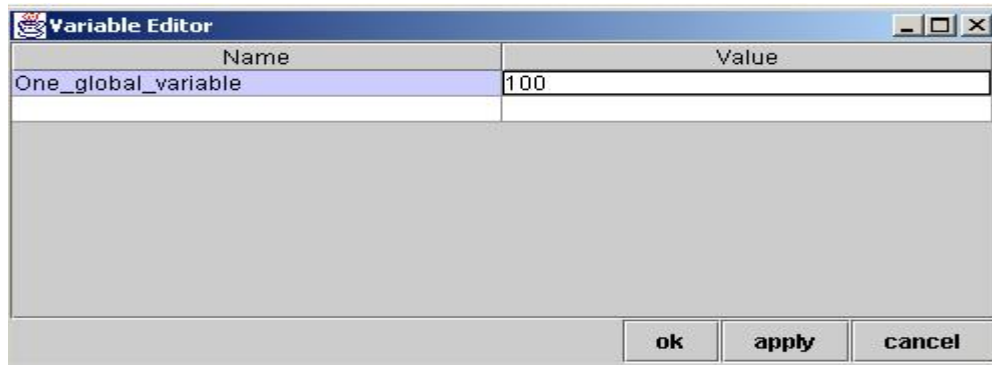
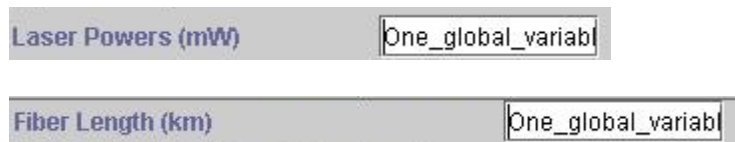



Figure 8 One Example of Defining A Global Variable

Then, in the corresponding laser and fiber parameter boxes, the user should input the global variable's name instead of "100":



If the laser power and the fiber length are changed to 50mW and 50km in next test, the user only needs to go to the Variable Editor window to replace "100" by "50". Otherwise, they have to click on each dialog box and do the same thing twice.

 - It displays a window. The user will be asked to assign a name to the test before pressing "Do Simulation". The details of running simulation will be discussed in section 3.3.11 later on.

3.2.2 Toolbar



Figure 9 FiberSim's Toolbar

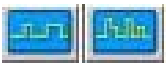
Actually, Toolbar is just a shortcut for Menubar. The user may already have some ideas from the symbols. Starting from the left, the first one is “New”, the second is “Save”, the third means “Open”, and the last one is “Simulate”. Their functions have been explained in 3.2.1.1.

3.2.3 Component Icons

There are total 23 optical components in FiberSim. Each of them is unique and has its associated parameters.



a laser – optical source, please see section 3.3.3.2



pulse generators – NRZ, RZ and PSK, please see section 3.3.3.1



an electrical filter – please see section 3.3.7.1



specified and custom electrical filters – user can specify the filter bandwidth or input filter data from file, please see section 3.3.10



an ideal modulator. The output power $P(t)$ is directly proportionally to the electrical signal input $v(t)$. Please see section 3.3.3.3



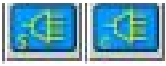
symbols of “loop stars” and “loop ends” – please see section 3.3.9



a multiplexer – please see section 3.3.3.4



a de-multiplexer – please see section 3.3.6.1



specified and custom de-multiplexers – user can specify the bandwidth or input data from file, please see section 3.3.10



a PIN photodiode – please see section 3.3.6.2



a specified photodiode – please see section 3.3.10



fiber – optical fiber, please see section 3.3.4



Raman fiber – pumps are include in the fiber to simulate Raman effect, please see section 3.3.4.5



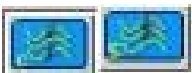
an optical amplifier – please see section 3.3.5



symbol of “output plots” – including eye diagram, pulse plot and spectra plot, please section 3.3.8



an optical filter – please see section 3.3.7.2



specified and custom optical filters – user can specify the filter bandwidth or input filter data from file, please section 3.3.10

3.2.4 Navigator Panel

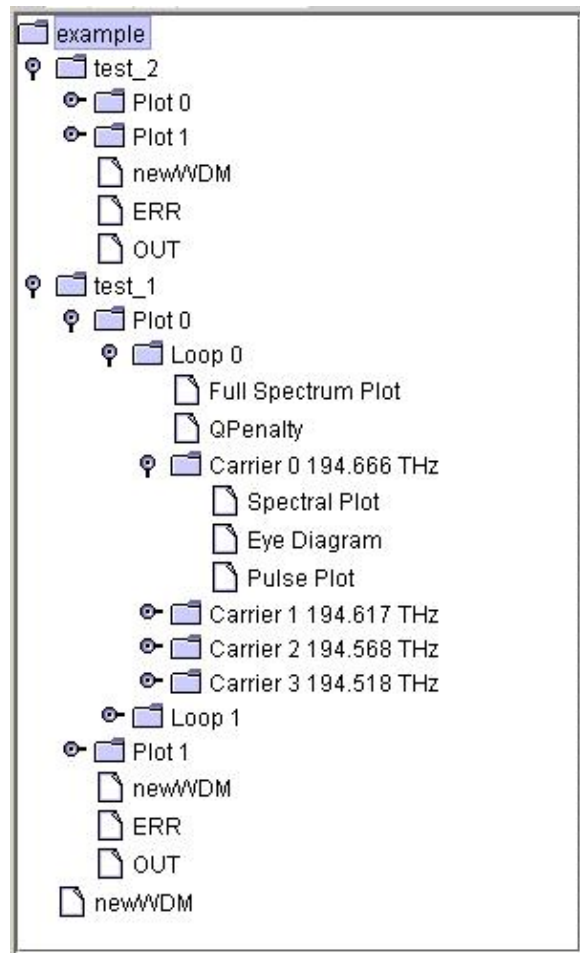

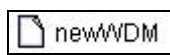


Figure 10 Navigator Panel

Navigator panel guides the user through simulated tests and output files. Refer to the example shown in Figure 10, “example” is the project name. The user only can open one project in each FiberSim window. “test_1” and “test_2” are two different tests under the same project. Every time the user starts a simulation, they need to give a name to the test. Once simulation is done, the test name will show up in the panel.

Click on any of the test, its output file list will pop up (see the list under “test_1”). If there


are N output plots  in the designed network, the user will see “Plot0”, “Plot1”, ... and “Plot(N-1)” in the list.



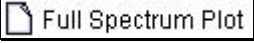
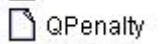
is an text file containing all the parameters defined in the test.

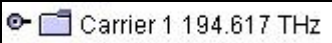


means error messages. If there is any error occurring during the simulation, it will be displayed here.

 is the output test file providing all the information associated with simulation, such as simulation time, the number of sample points, generated data pattern, fiber step size, etc. The information may become very helpful when the user needs to figure out what is wrong with the simulation results.

Under the “Plot N”, there are sub-lists: Loop0, Loop1, ...until Loop(M-1), if there are total M loops in the test.

Click on , it gives the complete spectrum showing all the carriers of the present LoopM. Similarly,  calculates and outputs the Q-values and eye penalties of every channel of the present LoopM.

 provides the user some information like how many carriers are in the system, which carrier they are looking at, and the carrier frequency. By clicking on each carrier, its graphical output list can be viewed. The list includes Spectral Plot, Eye Diagram and Pulse Plot. The corresponding plot will be displayed when the user click on its name. Sometimes the plot may take a while to show up depending on how complex the optical system is.

3.2.5 Multiple Windows

FiberSim supports multiple windows features. Every time the user opens a new project or loads a previous on, a new window appears. This feature is very useful when one needs to cut and paste part of a project from a window to another.

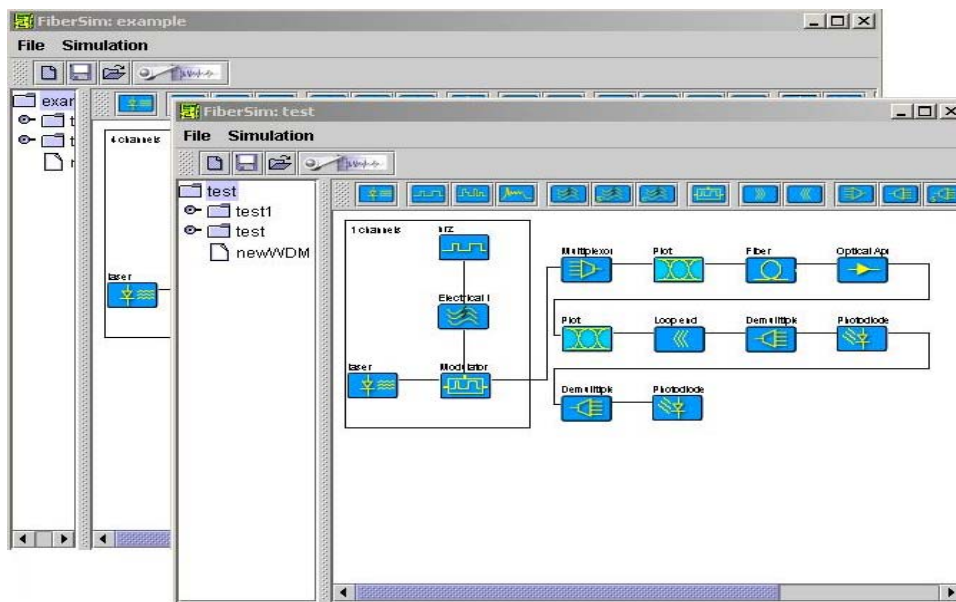


Figure 11 View of Multiple Windows

3.3 Getting Started

This section will teach the user step by step to design and set up a new simulation, explains how to use each component and define their parameters.

To start FiberSim, click on FiberSim.jar file, a window and the FiberSim GUI will appear (Figure 12) and ask the user to create a new project or open an old one.

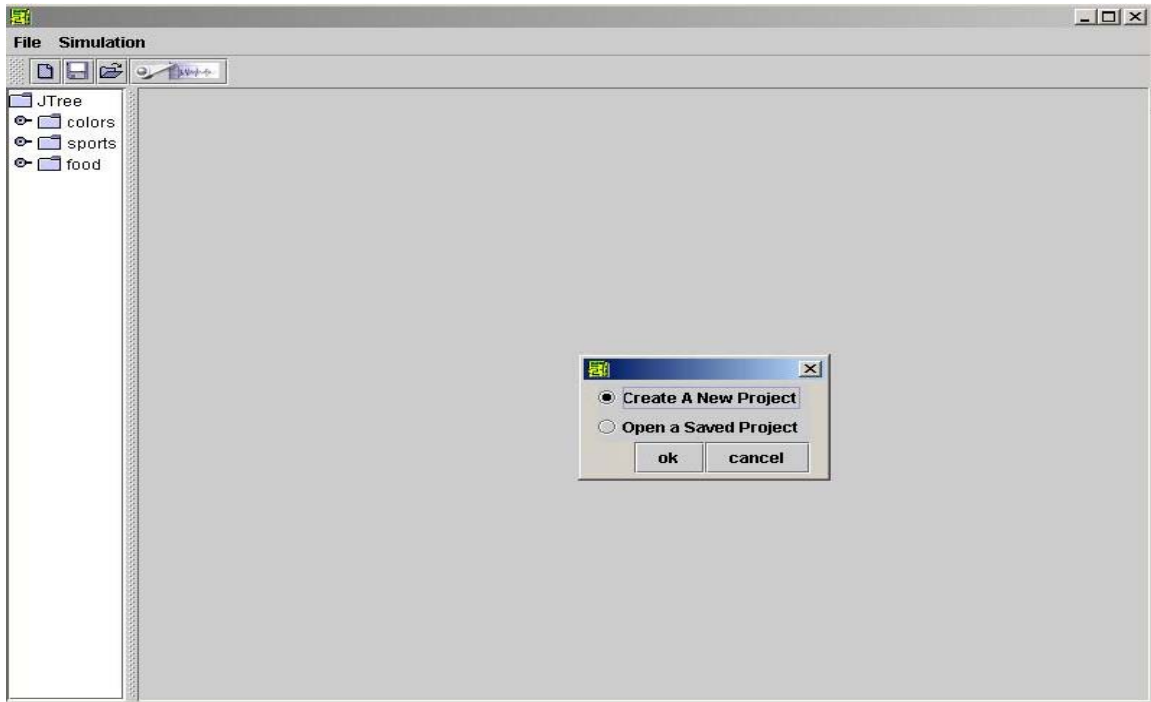


Figure 12 Starting Window of FiberSim

The default choice is to create a project. Click on "ok", the user will see something similar to what is shown in Figure 13:



Figure 13 Creating a Project in FiberSim

The user needs to select a directory to save the project. They may click on the downward arrow in “Look in: “ to choose the path. Then, the user has to assign a name to the project by typing it in the area after “File name: “. Finally, click on “Save”. Now the GUI looks like:

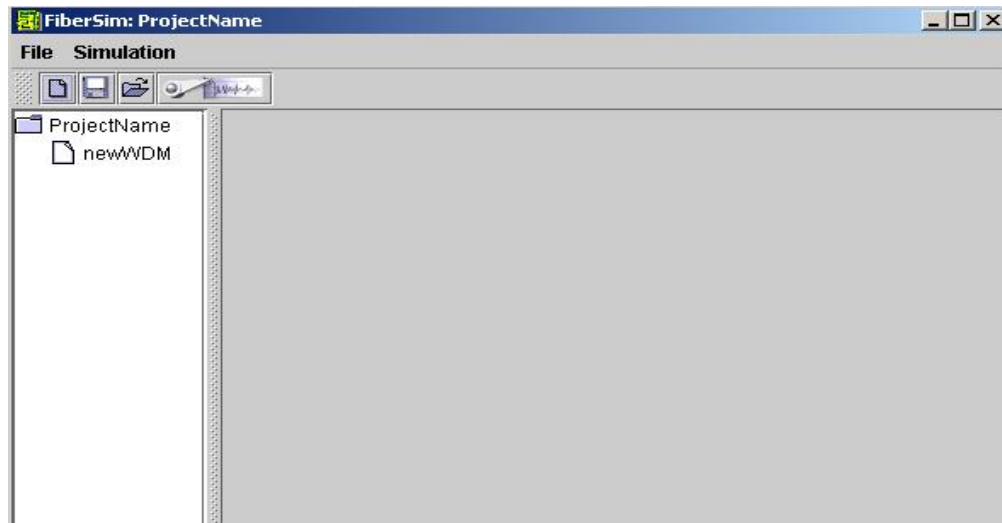


Figure 14 First View after Creating A Project

Notice that the project name will show up in the Navigator Panel. Click on the name, the component icons will appear.



Figure 15 Complete GUI View after Creating A Project

To open an old project, a very similar window as shown in Figure 13 will appear with an “Open” button instead of “Save” button. Just browse the directory to find the project, then click “Open”.

After creating the project, it is time to construct the network, which will be discussed next.

3.3.1 Constructing A Fiber Optic Network

Theoretically, to place a component in the drawing area, just pick one from the palette and left-click on it, then, the component icon will appear in the draw area automatically, connecting to the previous one. If the user likes to put the component in a specific place (e.g. between two icons) or move it to somewhere else, he or she needs to follow: select it with the left mouse button, drag it to the spot you want, then release the mouse.

However, to build a correct optical network, the user has to follow certain order: transmitter section is always the first, following is the transmission (fiber) section, last is the receiver section.

The first step is the transmitter section, which is at least composed of three components: data source, a laser bank, and a modulator. However, an electrical filter is usually added after the data source to improve the system performance. There are strict rules for building this section: the data source always goes first, then the filter (optional), then the laser, and the finally the modulator.

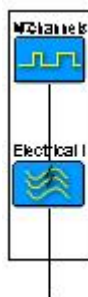


Figure 16 A Start of the Transmitter Section

As shown in Figure 17, the complete transmitter section is always bounded by a rectangle, which may remind the user about the section.

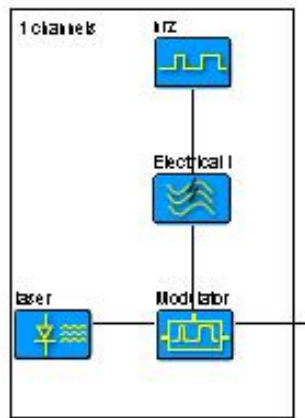


Figure 17. View of System Transmitter Section

A multiplexer is always placed after the transmitter section, which starts the transmission section.

The optical fiber usually follows the Mux. Optical amplifiers and measurement plots can be added before, after and in between fibers.

There is no rectangle outside the receiver section, which usually consists of de-multiplexer and photodiode. Filters and measurement plots can be inserted anywhere in the system.

A simple, complete optical system is shown in Fig. 18:

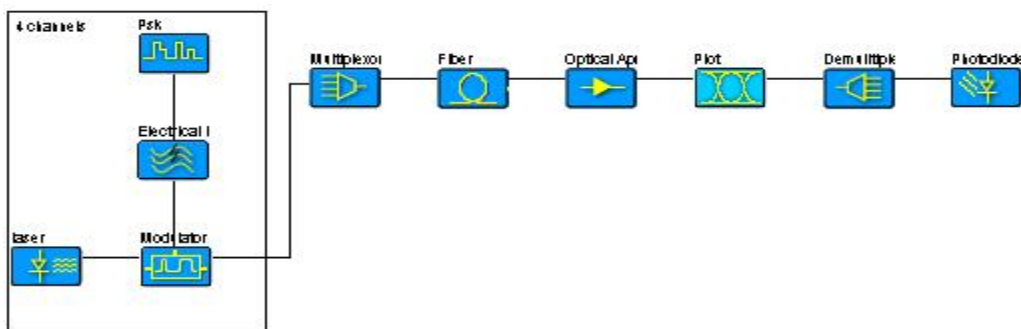


Figure 18 A Simple Example of Optical System in FiberSim

Deleting an item can be accomplished by the follow: left-click on the icon (the user will see the name on top of the icon is highlighted), right-click the mouse, a small menu will pop up,

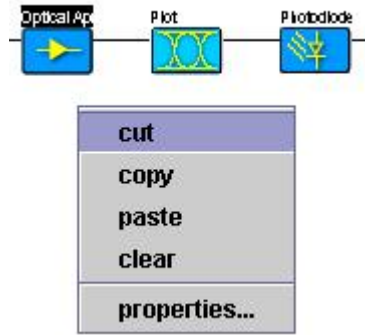


Figure 19 Mouse Pop-up Menu

then, choose cut or clear (they do the same job), the icon will disappear. Or, the user may press the button “Delete” from the keyboard to delete the component.

To copy and paste, just follows the same steps, except choosing command “copy” instead of “cut”. Then, move the mouse to another place or another window, right-click and select “paste”. If working on more than one items, first highlight the icons by drawing a rectangle over them, then copy and paste.

3.3.2 Simulation Parameters

Before assigning parameter values to different components, the Simulation Parameters need to be defined first, i.e. those parameters are essential to control the accuracy and duration of the simulation. Its window can be opened by double-click the blank area (not the icons). The user may choose to use global variables and input the variables’ names into the dialog box (see section 3.2.1.2).

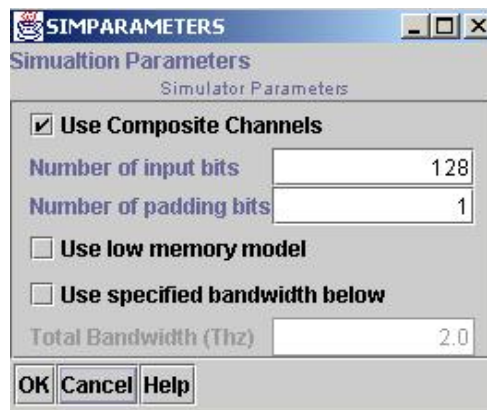


Figure 20 Simulation Parameters Window

3.3.2.1 Input Bits

Number of input bits

This parameter decides how long the input signal pattern will be at the reference data rate (see Pulse Generator in 3.3.3.1). Of course, longer pattern will give more accurate results. But keeping in mind that the CPU simulation time will increase with the total number of input bits. For example, in soliton systems, a very short pulse may be sufficient for studying propagation effects. On the other hand, if the user wishes to study the effects of noise of an optical system (a statistic process), a long pattern is required. Default value is 128.

3.3.2.2 Padding Bits

Number of padding bits

Padding bits means the additional “0”s added at the beginning and the end of the input signal pattern, which makes sure that there is no overlap between two patterns. For example, if the user wants the number of input bits to be 32, and the number of padding bits to be 2, then the actual number of data bits is $2+32+2 = 36$. There are two “0”s at each end of the real 32-bit-long signal. Default value is 1.

3.3.2.3 Simulation Bandwidth

Use specified bandwidth below
Total Bandwidth (Thz)

FiberSim allows the user define the simulation bandwidth in [THz] of each test. The default simulation bandwidth is 2 times the signal bandwidth. Users may also check the option “Use specified bandwidth below” first and input the bandwidth they wish in the box. This is one of the most important parameters in terms of required CPU simulation time, which should consequently be chosen with great care.

3.3.2.4 Simulation Options

Use Composite Channels

In a simulation of WDM system, it is better to check this option. It treats all the channels as one, composite channel, so the XPM between channels becomes SPM.

Use low memory model

This option reduces the number of samples and removes the buffers in Fiber. The memory can be cut by about half, but it will take longer time to simulate.

3.3.3 Transmitter Section

Now the Simulation Parameters have been set, the user is ready to design the optical network, starting with the Transmitter Section. The user may choose to use global variables and fill in the variables' names into any of the dialog box mentioned later on (see section 3.2.1.2).

3.3.3.1 Pulse Generators



- A NRZ and RZ rectangular pulse generator. To view its parameter box, double-click on the icon.

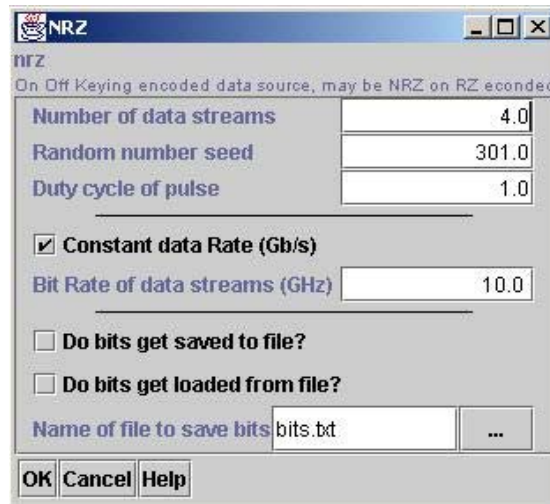


Figure 21 NRZ and RZ Pulse Generator Parameter Box

Number of data streams

It means the total number of channels being simulated in a test. Default value is 4.

Random number seed

In order to guarantee the randomness of the signal pattern generated, “Random number seed” has to be set. Pulse generators utilize those seeds to produce different sets of signal values in each run. The signal randomness is proportional to the number of seed. If a CW signal is required, input “0” here. Default value is 301.

Duty cycle of pulse

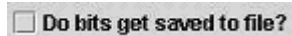
For NRZ signal, duty cycle is always 1. When a “1” or “0” is transmitted, the signal is at the High/On Level or Low/Off Level during the entire bit time.

For RZ signal, the value of duty cycle can be any number between 0 and 1. When a “1” is transmitted, the signal is at the High/On Level for as long as Duty Cycle x Bit Duration,

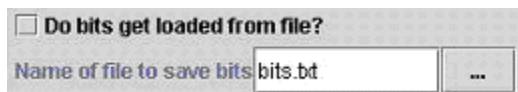
then it goes down to the Low/Off Level for the remaining time. When a “0” is transmitted, the signal is at the Low/Off Level for the entire bit time.





This parameter is used to set a reference data rate in [Gb/s] for the simulation. If the user wants a constant data rate for all the channels, just check the option and fill in the data rate value. If non-constant data rates are desired, do not check the option and input the values separated by commas (e.g. 10, 20, 30...). Default value is 10 Gb/s.



This option gives the user an opportunity to save the data bits to a file, in case the same bits needs to be used later on.



Data bits can also be loaded from a text file instead of being produced by FiberSim. Just check the option and write the file name in the box. The symbol  is a shortcut for “Browsing File in the Directories”. A window will pop up letting users to browse.

 - A Phase-Shift-Keying signal pulse generator. Its parameter box is shown in Figure 22.

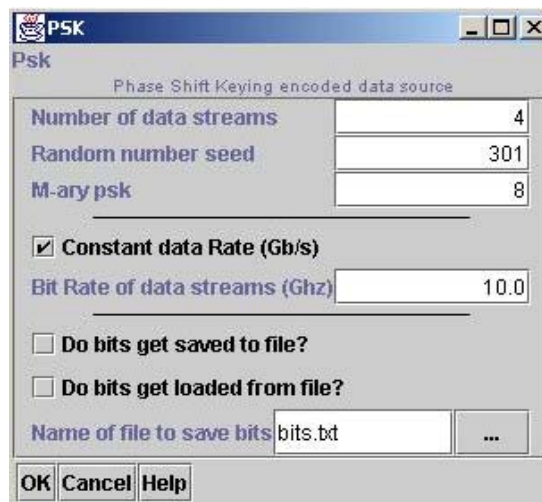


Figure 22 PSK Pulse Generator Parameter Box

There is only one parameter difference between PSK generator and NRZ/RZ generator. “Duty Cycle” is replaced by “M-ary psk”. This parameter tells how many different phases each transmitted bit has. Default value is 8.

3.3.3.2 Laser



- This icon simulates a laser, which is the optical source in optical network. Its parameter box looks like:

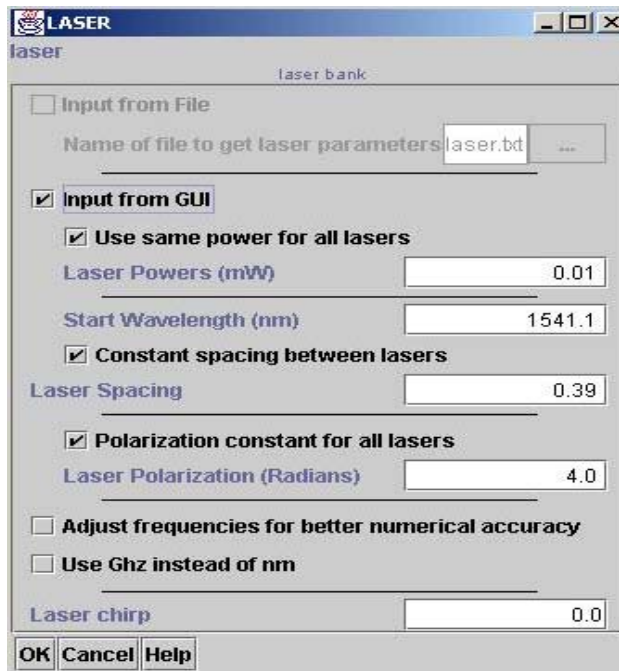


Figure 23 Laser Parameter Box

Laser parameters can be defined through GUI, or loaded from a text file.

To load a file, check Input from File and fill in file name.

To input from GUI, check Input from GUI.



If every optical signal has the same power, check the option and assign a value in [mW]. Otherwise, uncheck and input values separated by comma. Default value is 0.01 mW.



The user needs to input the starting wavelength in [nm] (the channel which has the highest frequency). Default value is 1541.1 nm.



For equally spaced channels, just input a value in [nm]. If the user prefers to use frequency instead of wavelength, remember to check Use Ghz instead of nm first, then input the frequency in [GHz] in the same box. Default spacing in wavelength is 0.39 nm.

If the channel spacing is not constant, uncheck the option and fill in values ([nm] or [GHz]) separated by comma.



This parameter defines the degree of polarization of each laser in [Rad]. Similar to previous parameters, one value is needed for constant polarization, or a set of values separated by comma for non-constant polarization. 0 Radian means the signal is aligned with the x-axis. Its default value is 4 Rad.

3.3.3.3 Modulator



- An ideal modulator is used to modulate electrical signals onto optical carriers. It has no parameters for users to set. The output power $P(t)$ is proportional to the electrical signal input $v(t)$. This modulator is similar to a \sin^2 modulator biased at $\pi/2$. (Its coding is implemented in Laser.C)



Figure 24 Modulator

3.3.3.4 Multiplexor



- A standard multiplexor combining optical signals into one composite channel. It has no parameters for users to set. (Its coding is implemented in Laser.C)



Figure 25 Multiplexor

3.3.4 Optical Fiber



- This symbol represents optical fiber. Double-click on the icon, its parameter box will appear (see Figure 26). This is the most important and complex dialog box of FiberSim, since it allows the user to introduce all the physical parameters of the fiber.

Fiber
Full fidelity fiber model

Fiber Length (km)	100.0
Standard deviation of Fiber Length	0.0
<input checked="" type="checkbox"/> Auto Adjust Step Size	
Number of Steps	200.0
<input checked="" type="checkbox"/> Include Loss	
Loss (Db/km)	1.0
<input checked="" type="checkbox"/> Include Dispersion	
Dispersion Factor (ps/nm/km)	3.7706
<input checked="" type="checkbox"/> Include Second Order Dispersion	
Dispersion Slope Factor (ps/(nm ² *km))	0.1142
Dispersion Slope @ Wavelength (nm)	1550.0
<input type="checkbox"/> Include Birefringence	
Birefringence Coefficient (ps/km)	0.2
<input checked="" type="checkbox"/> Include PMD (birefringence overridden)	
PMD Coefficient (ps/(km ^{.5}))	0.2
Seed for polarization changes	1.0
<input checked="" type="checkbox"/> Include Self Polarization Phase Modulation	
<input type="checkbox"/> Include Cross Polarization Phase Modulation	
<input type="checkbox"/> Instantaneous Non-Linear Response	
<input type="checkbox"/> Full Delayed Non-Linear Response	
<input type="radio"/> 1st Order Moment Approximation	
<input type="radio"/> 2nd Order Moment Approximation	
<input type="radio"/> Average Riemann Method	
Skip Factor	1.0
<input type="radio"/> Direct Integration	
<input checked="" type="radio"/> FFT Method	
NonLinear Refraction Index (m ² /watt)	3.2
Effective Core Area (um ²)	53.0
Peak Raman Factor	0.0
Raman Bandwidth (THz)	14.0
tau1 Raman oscillation constant [fs]	12.2
tau2 Raman decay constant [fs]	32.0
Raman delay fraction	0.24

OK Cancel Help

Figure 26 Optical Fiber Parameter Box

3.3.4.1 Single-Mode Fiber

FiberSim is capable of simulating different types of fibers, such as Single-mode Fiber and Leaf Fiber (non-zero dispersion shifted fiber), depending on the dispersion coefficients the user input from GUI.

3.3.4.1.1 Basic Parameters

Fiber Length (km)

The estimated length of fiber is defined in [km]. Default value is 100km.

Standard deviation of Fiber Length

This parameter is defined in [km] also. If the actual length of the fiber is not known with good accuracy, a statistical variation can be taken into account. The actual length is evaluated as Input Length + Std. Length * Random Number. Default value is 0.

Auto Adjust Step Size
Number of Steps

Step size is a parameter associated with the simulation method (Split-Step, see section 5.2). The smaller the step size is, the longer the time takes to simulate. The user can decide how small they want the step size to be by filling in the number of steps. Then, the step size is equal to Actual Fiber Length / Number of Steps. Or, the user just need to choose the Auto Adjust option and let FiberSim set the step size. Default value of steps is 200.

Include Loss
Loss (Db/km)

This is the fiber attenuation parameter. If fiber attenuation is desired, input the value in [dB/km]. Default value is 1 dB/km.

3.3.4.1.2 Dispersion Parameters

Include Dispersion
Dispersion Factor (ps/nm/km)

This dispersion factor D [ps/nm/km] represents the dispersion value of the fiber. Its default value is 3.7706ps/nm/km. Also, fiber dispersion can be excluded in simulation by turning off the option.

<input checked="" type="checkbox"/> Include Second Order Dispersion	
Dispersion Slope Factor (ps/(nm ² *km))	0.1142
Dispersion Slope @ Wavelength (nm)	1550.0

Dispersion slope ($dD/d\lambda$) is a parameter related to the second-order fiber dispersion and is expressed in [ps/km/nm²]. Default value is 0.1142ps/km/nm². Since dispersion slope is a derivative, which means variation of the dispersion with respect to the wavelength, the user has to define the slope value at a specific wavelength. The default wavelength is 1550nm. Similar to Dispersion, the user has the choice to turn off 2nd-order dispersion.

3.3.4.1.3 Nonlinearity Parameters

Birefringence and PMD are two nonlinear parameters for polarization effects.

<input checked="" type="checkbox"/> Include Birefringence	
Birefringence Coefficient (ps/km)	0.2

If switching on the fiber birefringence, the user needs to input a value in [ps/km]. If the option is turned off, the parameter box will be disabled. Default value is 0.2ps/km.

<input checked="" type="checkbox"/> Include PMD (birefringence overridden)	
PMD Coefficient (ps/(km ^{1/2}))	0.2
Seed for polarization changes	1.0

If turning on the PMD option, the parameter box will become active. The PMD coefficient is expressed in [ps/km^{1/2}]. Its default value is 0.2ps/km^{1/2}. Since PMD effect is a statistical process, the user also needs to set a random seed.

NonLinear Refraction Index (m ² /watt)	3.2
---	-----

The nonlinear refractive index n_2 is a measure of the fiber nonlinearity. Its unit is [m²/watt]. Default value is 3.2 m²/w for most SMF.

Effective Core Area (um ²)	53.0
--	------

This parameter A_{eff} is used to evaluate the nonlinearity coefficient γ . It is expressed in [μm^2]. Default value is 53 μm^2 for most Leaf fiber.

Peak Raman Factor	0.0
-------------------	-----

This parameter is used to calculate Raman gain. The value filled in should be multiplied by a power of -14 by FiberSim.

Raman Bandwidth (Thz)	14.0
-----------------------	------

Stimulated Raman Scattering has a gain spectrum falling in a range of 10 to 14 THz. Users may decide how wide they want the gain spectrum to be. Default value is 14 THz.

tau1 Raman oscillation constant [fs]	12.2
tau2 Raman decay constant [fs]	32.0
Raman delay fraction	0.24

τ_1 [fs], τ_2 [fs], and f_R are three constants associated with Raman effect. Those values shown above are the most commonly used values for the three constants in optical system.

- Include Self Polarization Phase Modulation**
- Include Cross Polarization Phase Modulation**

FiberSim also allows the user to decide whether they want to include Self/Cross Polarization Phase Modulation in the simulation.

Switching on the first option, delayed SPM is included.
Switching on the second option, delayed XPM and delayed FWM are included.
(For explanation of delayed or instantaneous terms, please see section 5.3)

Of course, the two options can be turned on or off at the same time.

3.3.4.1.4 Simulation Methods

There are several simulation methods for the user to choose from, depending how fast and how accurate the user wants the result to be.

Instantaneous Non-Linear Responce

This simulation method only includes the instantaneous nonlinear effects (SPM, XPM and FWM). It takes the shortest time to simulate, but with the least accuracy.

Full Delayed Non-Linear Responce

To include delayed SPM and/or delayed XPM, which depends on users' choices, a full delayed nonlinear method must be used.

There are four different methods that are capable of doing full delayed response.

- **Direct Method**
This is the simplest method used in the time domain to evaluate the integrals of delayed response, $\int f(t-s)hr(s)$. $hr(t)$ is a fast-varying function in time domain, so in order to achieve high numerical accuracy, a huge sampling bandwidth has to be used. Therefore, the method is time-consuming and the numerical efficiency is poor.
- **1st-order and 2nd-order Moment Methods**
To reduce the computation time, Moment methods can be employed. In Moment methods, the integral is expressed as different orders of time derivatives at each sample point. The more higher-order derivatives are included, the more accurate the simulation is.
- **Fast Fourier Transform (FFT) Method**
Since convolution in time domain can be performed as multiplication in frequency domain, so what FFT method does is to multiply the spectrum of $f(t)$ and $hr(t)$ in frequency domain. In this way, the simulation bandwidth is determined by signal bandwidth instead of the bandwidth of $hr(t)$. FFT method achieves the best numerical efficiency and takes less time to simulate.
- **Average Riemann Method**
This method is no longer used by FiberSim.

3.3.4.2 Raman Fiber

FiberSim components family has a special member called “Raman Fiber”, in which the signals and pumps can co-propagate in the same direction at the same time along a single-mode fiber. Stimulated Raman Scattering effect is taken into account during the simulation. Compared with single-mode fibers, Raman fiber has two more parameters associated with pumps.

Pump Frequencies (Ghz)	300.0
Pump Power (mW)	1.0

The user may define the pump’s power [mW] and frequency [GHz]. Multiple pumping is also allowed, the user just needs to input the values separated by comma. Default pump frequency is 300GHz, and default pump power is 1mW.

3.3.5 Optical Amplifier



- This component simulates a standard Erbium Doped Fiber Amplifier (EDFA). Dialog box for this icon is shown in Figure 27.

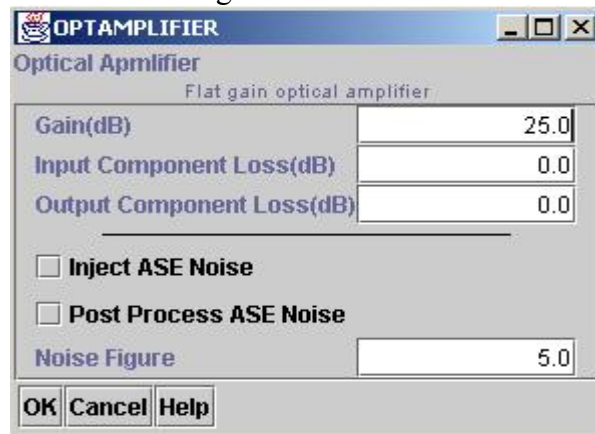
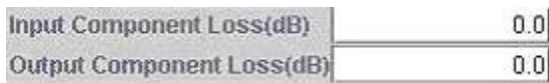


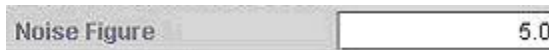
Figure 27 Optical Amplifier Parameter Box



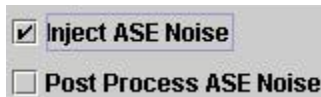
The EDFA gain is expressed in [dB]. Default value is 25dB.



If non-ideal amplifier is desired, the user can decide how much loss will come from the amplifier input and output in [dB]. The default values are 0dB loss, which represents an ideal amplifier characteristic.



Noise figure is a unitless parameter. Default value is 5.



ASE noise can be either included or excluded in the simulation. “Inject ASE Noise” means the noise will be considered in the numerical simulation, this option is usually checked when the user intends to obtain small signals or to see the noise effect on system performance. “Post Process ASE Noise” means that the ASE noise is not injected into the numerical signal, but FiberSim will be aware of its affect during post-processing Q-value calculation.

3.3.6 Receiver Section

Receiver section is at least composed of two components: de-multiplexer and photodiode. Optical filter and Measurement Plot can be inserted. For example:

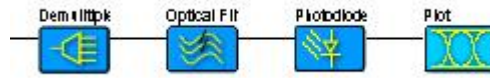


Figure 28 One example of System Receiver Section

3.3.6.1 De-multiplexer



- In FiberSim, the de-multiplexer includes an optical filter. It filters the optical signals and shifts them to the base-band. Then the photodiode will detect them.

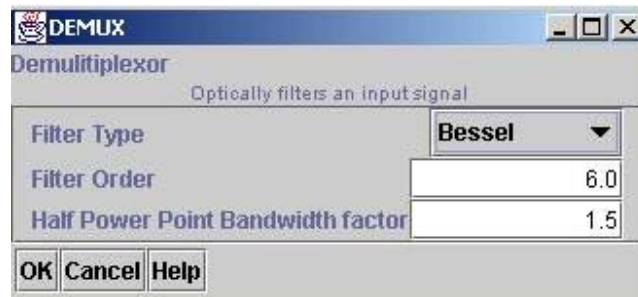


Figure 29 De-multiplexer Parameter Box

Filter Type

FiberSim offers four types of optical filters, the user may click on the downward arrow to



see the list: , left-click on one of them, the filter will be uploaded to the Filter Type box. Default filter type is Bessel.

Filter Order

“Filter Order” can be set by the user. A very narrow (higher-order) filter is not necessary good for system performance, because nonlinear signal distortion will become stronger in the pass-band. Default order is 6th.

Half Power Point Bandwidth factor 1.5

For an optical filter, its double-side 3dB bandwidth factor needs to be defined. Default value is 1.5, which is commonly used. For example, if the data rate is 10Gb/s (full-bandwidth is 20GHz), then the optical filter 3dB bandwidth is $10 \times 1.5 = 15$ GHz.

3.3.6.2 Photodiode



- This symbol represents a standard PIN diode, which detects and converts optical signals into electrical signals. It also serves as an electrical filter in FiberSim.

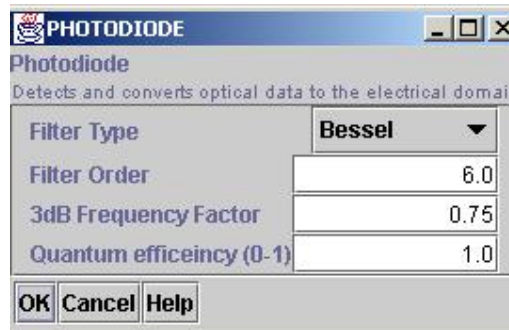


Figure 30 Photodiode Parameter Box

Filter Type

Similar to the DEMUX, the user can pick the filter type from the same list.

Filter Order 6.0

The user can define the order of the electrical filter. Default order is 6th.

3dB Frequency Factor 0.75

For an electrical filter, only single-side 3dB bandwidth needs to be set. Default value is 0.75. For example, if the data rate is 10GHz, which means the electrical signal's one-side spectrum is 10GHz, then the electrical filter's single-side 3dB bandwidth is $10 \times 0.75 = 7.5$ GHz.

Quantum efficeincy (0-1) 1.0

Quantum efficiency defines how many electro-hole pairs are generated per incident photon in the photodiode. The user can input any value between 0 and 1. In ideal case, the quantum efficiency is 1, which is the default value.

3.3.7 Filters

Although the Demux serves as an optical filter and the photodiode serves as an electrical filter, FiberSim has its own electrical and optical filters.

3.3.7.1 Electrical Filter



- An electrical filter. Actually the symbols for electrical and optical filters are the same, but after the user places the component into the drawing area, the component's name will appear on top of the icon, which will help the user to distinguish the two icons.

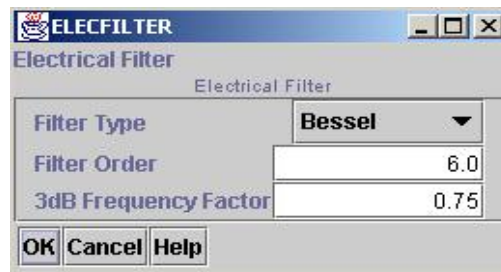


Figure 31 Electrical Filter Parameter Box

All of the parameters for electrical filters have been well explained in Photodiode section, please read 3.3.6.2 for details.

3.3.7.2 Optical Filter



- An optical filter.

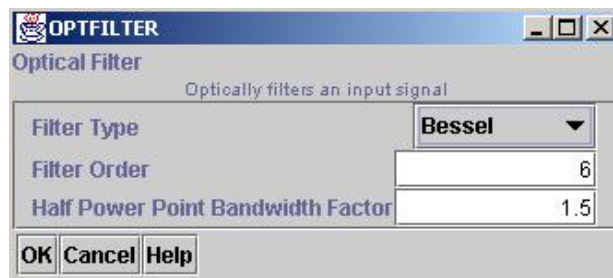


Figure 32 Optical Filter Parameter Box

All of the parameters for optical filters have been well explained in De-multiplexer section, please read 3.3.6.1 for details.

3.3.8 Measurement Plots



- This symbol tells FiberSim to output the measurement plots and Q-values. The user will see file lists (see Figure 33) shown in the Navigator Panel:

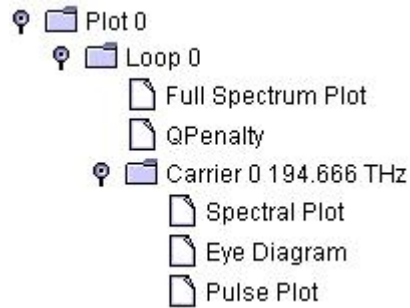


Figure 33 One Example of Graphical Output List

3.3.8.1 Spectrum Plots

- Full Spectrum Plot - This plots shows the entire spectrum, which includes all the channels in a WDM system.

For example, in a 4-channel system, the full spectrum plot is shown in Fig. 34.

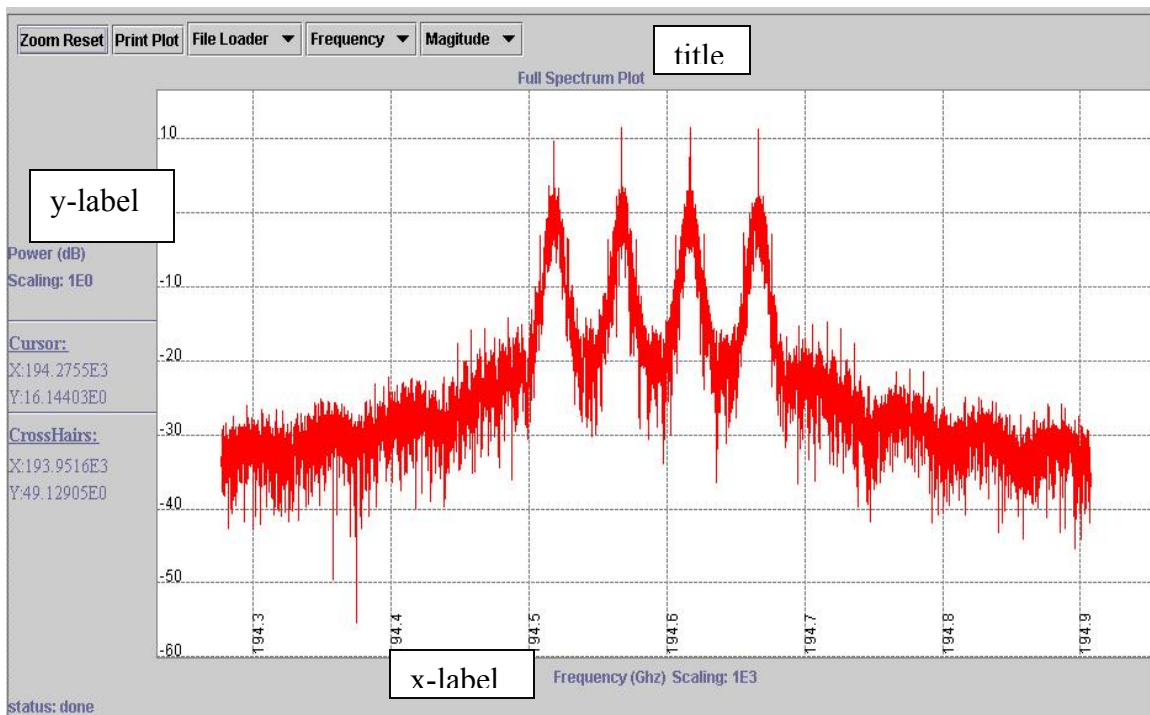


Figure 34 One Example of Full Spectrum Plot

The plot displays its title, x-label and y-label, and the units and scaling are also shown next to the labels.

Cursor: The plot also displays the cursor coordinates next to the y-axis, and the coordinates will keep changing according to the position of the cursor when the mouse is moving across the spectrum.

CrossHairs: not important, can be ignored

status: done Plot status is showing at the left-lower corner of the plot. Since some plots may take a long time to display, this mark will let the user know which step the plotting process is at. The usual procedures are: Bit Creator → File Loader → (type name) Filter ... → Inverse Fourier Transform → done.

There are a few commands showing at the left-upper corner of the plot:

Zoom Reset This button allows the spectrum return to the original scale after zoom-in and/or zoom-out. To zoom-out, just right-click the picture as long as the cursor is inside the grid area. To zoom-in, use the mouse to draw a rectangle over the spot the user wish to see, then release the mouse, the enlarged portion will appear.

Print Plot To print the plot, click on this button, a printing window will pop up. Then, the user may pick a printer to print. (Of course, at least a printer has to be connected to the computer)

File Loader not important, can be ignored.

Frequency
Time
Frequency This little menu gives the user the choices to see either the spectrum in frequency domain or the pulse plot in time domain. Actually, the user may just go to the Navigator Panel and click on the “Spectra Plot” and “Pulse Plot”.



This menu allows the user to view different portions of the plot.

“Magitude” —a magnitude plot combining x-polarized and y-polarized signals


“X real” –a magnitude plot only shows the x-polarized signal

“Y real” –a magnitude plot only shows the y-polarized signal

“X Power” –a power (magnitude²) plot only shows the x-polarized signal

“Y Power” –a power (magnitude²) plot only shows the y-polarized signal

For example, if the signal launched is only x-polarized, the user should see nothing from the “Y real” and “Y Power” plots, also, the “Magitude” plot and “X real” plot should be the same.

-  Spectral Plot - This plot only displays the spectra of one channel

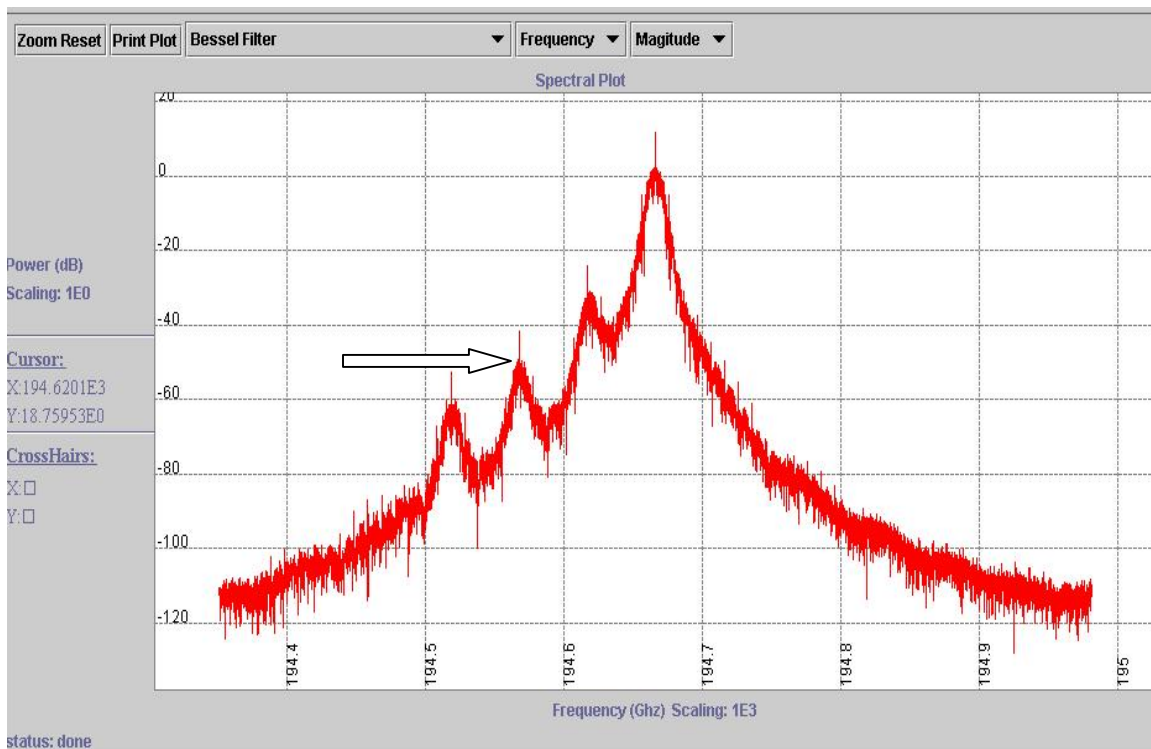



Figure 35 Spectra Plot of one channel

The plot commands and functions work the same way as in “Full Spectrum Plot”. Those small peaks on the side of the spectra are caused by nonlinear distortion.

3.3.8.2 Pulse Plot

 Pulse Plot - the pulse plot of one channel.

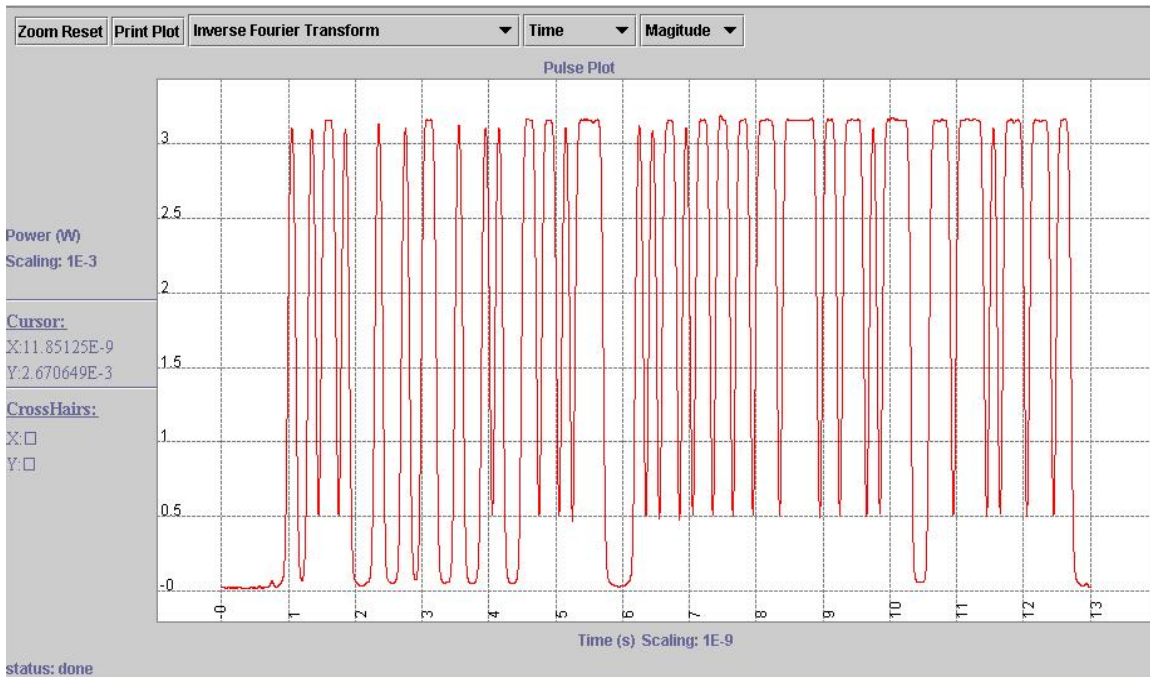



Figure 36 Pulse Plot of One Channel

The example shown above is the pulse plot of a NRZ signal. The plot commands and functions will not be repeated here.

3.3.8.3 Eye Diagram

 Eye Diagram - the eye diagram of one channel

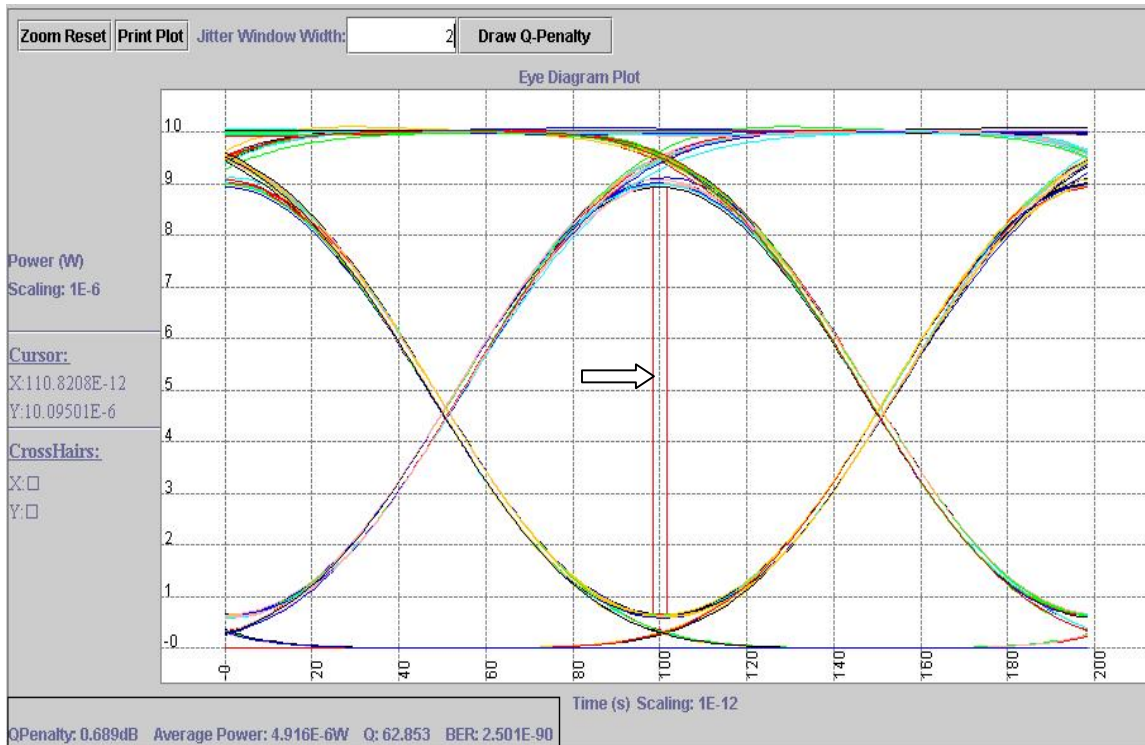


Figure 37 Eye Diagram of One Channel

A couple of features need to be mentioned here for eye diagram. One is that Q-value, Eye Penalty, BER value and Average Power of this channel are shown at the lower-left corner of the eye diagram (also see section 3.4).

Secondly, the jitter window width of the eye diagram can be enlarged or reduced by the user by inserting a number in the **Jitter Window Width:** , which is shown on the top of the plot. Default width is 0 (a straight line).

For certain tests, there is lots of noise in the system and FiberSim may not be able to find the best jitter window for the eye diagram. Then the user may draw the jitter window using **Draw Q-Penalty**. Just click on it and plot a rectangle over the eye opening area.

The Q-value mainly depends on the height of the window. Changing the width of the window will not affect the Q-value much.

3.3.9 Multiple Spans



- These two icons represent “loop begins” and “loop ends”, respectively, which allow FiberSim to simulate multiple spans.

The two symbols groups a portion of the network and iterates it N times (set by users). The resulting structure is equivalent to a chain of identical components, the output of one feeding the other. For example, a fiber followed by an amplifier is often iterated. To set the iteration times (6 spans by default), click on the “loop begins” icon.

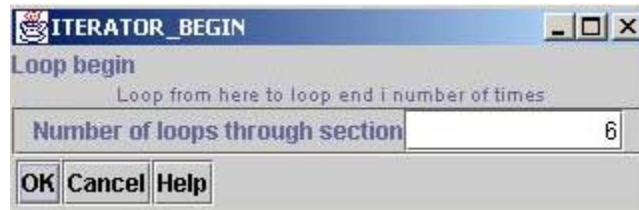


Figure 38 Iterator Parameter Box

A simple example of multiple spans is shown below.



Figure 39 One Example of Multiple Spans

There is no parameter associated with “loop ends” icon.



Figure 40 Loop End Dialog Box

3.3.10 User Specified Components

- User specified electrical and optical filters



- Symbol for user specified electrical/optical filters. The user may distinguish them from their titles shown on top of the icon.

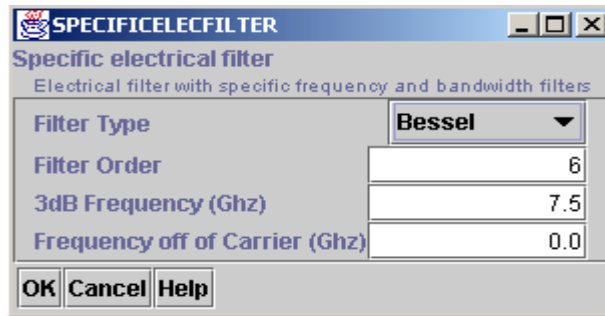
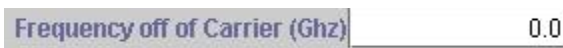
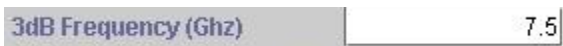


Figure 41 User Specified Electrical Filter Parameter Box



The user can define the center frequency of the electrical filter according to the system requirement. For example, in a simulation using Vestigial-Sideband (VSB) modulation, only one side of the signal is needed. If the electrical signal bandwidth is 10GHz, the user may set the electrical filter center frequency at 3.5GHz with 3dB Frequency bandwidth (single-side) of 4GHz, so one side of the signal is filtered. The default value is 0GHz (at baseband).



This parameter defines the 3dB bandwidth (single-side for electrical) of the filter in [GHz], not the “3dB frequency factor” mentioned in section 3.3.6.2.

The rest parameters have been discussed in 3.3.6.2.

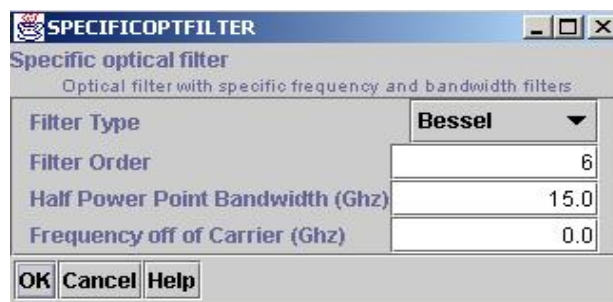
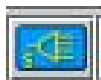


Figure 42 User Specified Optical Filter Parameter Box

Half Power Point Bandwidth (Ghz)

This parameter defines the 3dB power bandwidth (double-side for optical) of the filter in [GHz], not the “3dB power bandwidth factor” mentioned in section 3.3.6.1. Similarly, this feature may be also helpful for VSB modulation. Default value is 15GHz.

- User specified De-multiplexer



- User specified Demux



Figure 43 User Specified Demultiplexer Parameter Box

Since the Demux in FiberSim serves as an optical filter, the dialog box of “User specified Demux” is the same as the one of “User specified optical filter”.

- User specified photodiode



- User specified Photodiode

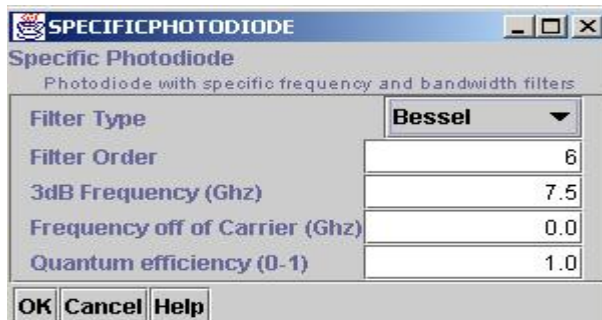


Figure 44 User Specified Photodiode Parameter Box

A regular photodiode in FiberSim detects the signal and serves as an electrical filter. Comparing with the normal one, a user specified photodiode allows the user to specify the center frequency of the electrical filter. The rest parameters are identical.

- Custom electrical and optical filters



- Symbol for custom electrical and optical filters.



Figure 45 Custom Electrical Filter Parameter Box



Figure 46 Custom Optical Filter Parameter Box

Name of file that contains the filter points filter.dat ...

Users may load filter data from a file, so they will have the exactly filter they wish for.

Frequency offset 0.0

The electrical/optical filter center frequency needs to be set in [GHz]. Default value is 0GHz.

- Custom De-multiplexer




Figure 47 Custom Demultiplexer Parameter Box

Since the demux works as an optical filter, the user can load the filter data from a file too, and set the center frequency.

3.3.11 Running and Stopping Simulation

When a network has been built and the parameters have been defined, it is the time for the user to start the simulation.

Click on  from the tool bar, the simulation window will pop up:

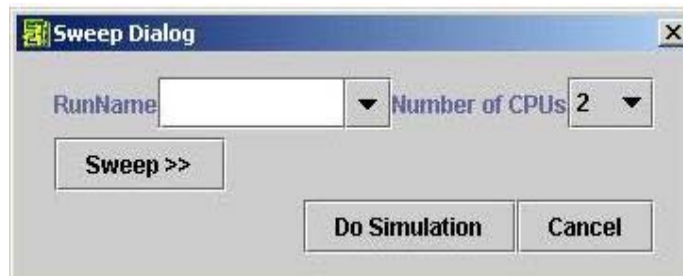
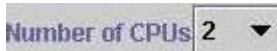


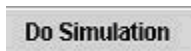
Figure 48 View of Simulation Window



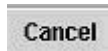
First, the user needs to assign a name to the simulation. Space is allowed in the name. Click on the downward arrow, a menu of the old simulation tests will appear. The user may choose a used name, then, the new simulation will replace the old one. Remember, the old test will be lost.



If the computer operating system has multiple CPUs, the user may decide how many CPUs they want to simulation the test. More CPUs will take shorter time to simulate. However, if there is only one CPU, just choose “1”. To upload the number of CPUs, click on the downward arrow to view a list, then click on the number.



will let FiberSim start Calculating, and



will cancel the action.

There is an option called , click on it, the window will become:

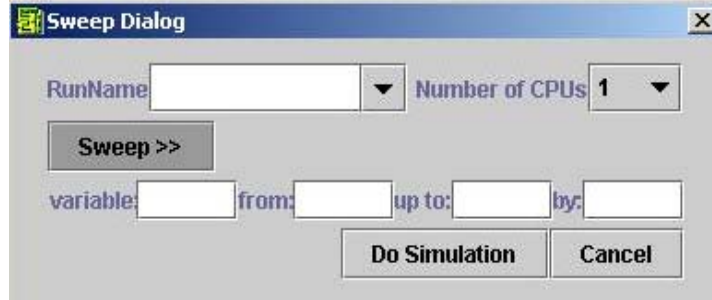


Figure 49 View of Sweep Window

If the user wants to simulate a series of tests with only one different variable value, this feature may become handy. The following example explains how to use it.

Say the user needs to simulate four tests. The only difference between the tests is the laser power will increase by 1mW every time starting with 1mW. First, pick a variable name for the laser power such as “LaserP” and insert it in the laser dialog box (see 3.3.3.2). Then, assign a name “diff_LaserP_tests”.

Fill in . Press “Do Simulation” button. The simulation will repeat four times automatically with different laser power values and outputs four test files shown in the Navigator Panel with names like:

diff_LaserP_tests-LaserP=1.0,
 diff_LaserP_tests-LaserP=2.0,
 diff_LaserP_tests-LaserP=3.0,
 diff_LaserP_tests-LaserP=4.0.

The user may click on any of the tests to view the results.

After pressing the “Do Simulation” button, two windows will appear. One is the simulation status window shown in Figure 50.

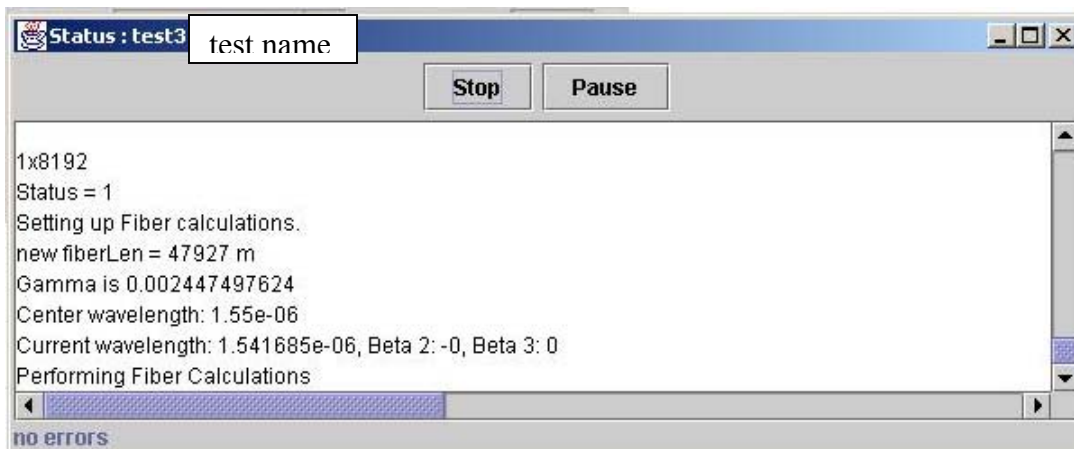


Figure 50 Simulation Status Window

The other is a black window created by the source code. In Windows 2000 operating system, nothing shows up inside it. However, if running FiberSim in DOS environment, some calculations will be displayed here.

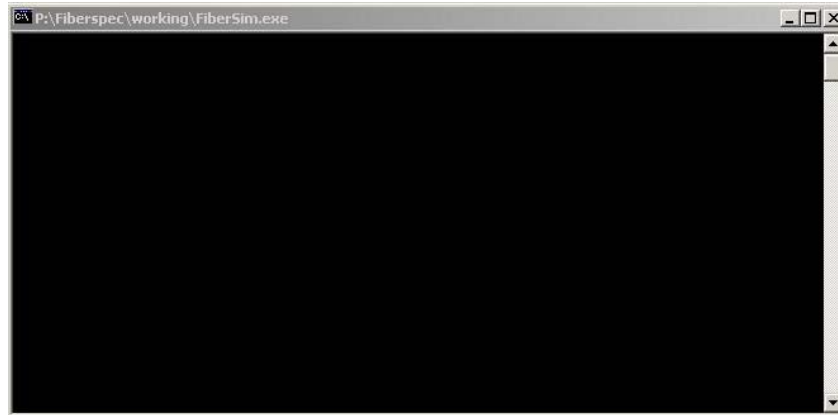


Figure 51 Simulation Window II

To pause the simulation, just click on **Pause** (Figure 50). The same button will change to **Resume** right away, which resumes the simulation.

To stop the simulation, click on **Stop**, two simulation windows will disappear. The output data until the stop point can be viewed by clicking on **OUT** in the Navigator panel.

3.4 Q-value and Eye Penalty Calculation

FiberSim is able to calculate and output the Q-value, eye penalty, bit error rate (BER) and average power of every channel. Click on button **QPenalty** shown in the Navigator Panel, a status window will appear:



Figure 52 Q-value Calculation Status Window

This window tells the user which carrier Q-value is being calculated and allows the user to stop the calculation by pressing **Stop**.


At the same time, the calculated values are being displayed in the drawing area. For example:

Carrier: 0 SubCarrier: -1 QPenalty: 2.084db	Q: 14.628 Average Power: 3.717E-4W	BER: 1.002E0
Carrier: 1 SubCarrier: -1 QPenalty: 2.209db	Q: 14.256 Average Power: 4.651E-4W	BER: 1.016E0
Carrier: 2 SubCarrier: -1 QPenalty: 2.125db	Q: 14.323 Average Power: 4.346E-4W	BER: 1.007E0
Carrier: 3 SubCarrier: -1 QPenalty: 2.388db	Q: 13.93 Average Power: 4.672E-4W	BER: 1.04E0
Carrier: 4 SubCarrier: -1 QPenalty: 2.271db	Q: 14.254 Average Power: 4.031E-4W	BER: 1.024E0
.....		

Figure 53 Q Calculation Results

Each row shows the values of Qpenalty (eye penalty), Q, Average Power and BER of one carrier. The units are included.

3.5 Output File

In the Navigator Panel (see 3.2.4), the user always sees a button  under any test. By clicking on it, the complete output file (printed by source code) will show up in the drawing area.

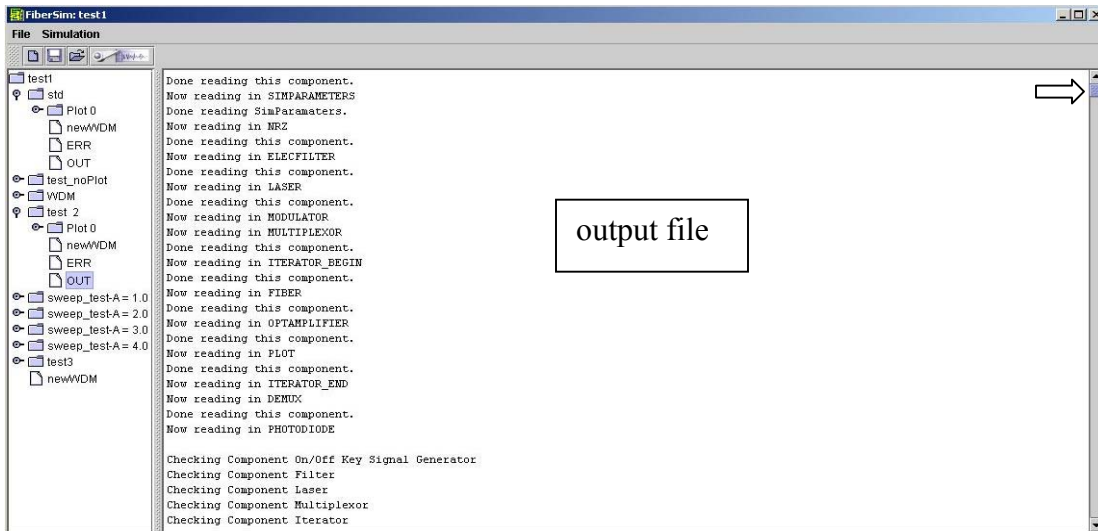


Figure 54 Output File Window

Usually, the output file is very long. The user can move the scroll bar or click on the up/down arrow to view the whole file.

- At the beginning of the file, the user may see something saying “reading and checking the component XXX”.

- Then, it is the simulation section including parameters such as simulation bandwidth, wavelength, data rate, ...etc.
- Following statements tell the user that each component is being executed, and related parameters are also displayed. The user will see sentences like “Now Executing: Component Name”. For example,

```

-----
Now Executing: Laser
    ISINPUTFILE = False
    LASERFILENAME = laser.txt
    ISINPUTGUI = True
    ADJUSTFREQUENCIES = False
    CHIRP = 0
    WAVELENGTHS = 1.5411e-06 1.54149e-06 1.54188e-06 1.54227e-06

    ISUNIFORMPOWER = True
    LASERPOWERS = 1e-05 1e-05 1e-05 1e-05

    ISCOPOLARIZATION = True
    POLARIZATIONS = 0 0 0 0
    SPACING =
    ISFREQ = False
    ISSPACINGCONSTANT = True

lx8192
Status = 1
-----

```

Figure 55 A Portion of Output File

The laser’s power, wavelength, polarization ...etc. are showing in the file.

- At the end, simulation time will be displayed.

The output file includes lots of information and may become very helpful for debugging. The parameters stated have meaningful names, so the user should not have much difficulty to understand the output file.

3.6 Error Messages

A general error message is shown below:

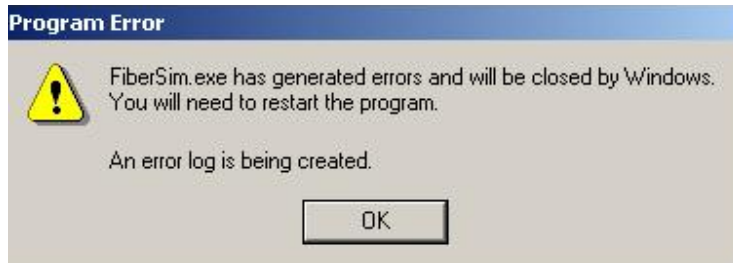



Figure 56 An Example of Error Message

It is generated by the source code during simulation. The user may start with checking the parameter values and the network design.

If there is a memory problem associated with the simulation, a similar error message will pop up and state that the memory is not enough.

Sometimes the simulation is finished, but the results cannot be viewed. The user can click on  in the Navigator Panel. Some types of errors may be shown, some may not.

Appendix A

Fiber parameters for Single-Mode Fiber and Leaf (NZ-DSF) Fiber

	Single-Mode Fiber	Leaf Fiber
Fiber Loss α [dB/km]	0.25	0.25
Dispersion D [ps/nm/km]	16.7137	3.7706
Dispersion Slope S_0 [ps/km/nm²]	0.09	0.1143
Effective Core Area A_{eff} [μm^2]	83	72
Nonlinear Refractive Index n_2 [m²/Watt]	3.2	3.2
Raman Oscillation Constant τ_1 [fs]	12.2	12.2
Raman Decay Constant τ_2 [fs]	32.0	32.0
Raman Delay Fraction f_R	0.24	0.24

Appendix B

Maintaining and Updating FiberSim Source Code

An Overview

FiberSim is written by the people who work in the Lightwave Communication Lab of Information & Telecommunication Technology Center (ITTC) at the University of Kansas. It resides on ITTC network under a public directory `\\devnull\projects\Fiberspec`. The original source code of FiberSim is maintained and updated through WinCVS (Control Version System), a software allowing multiple people to view and modify the code together. CVS can keep different versions of the source code on the network and protect them from outside users. CVS can be either downloaded from: `\\devnull\projects\Fiberspec\windows\Tool\WinCvs120.zip` on ITTC network, or from the Internet.

To modify the source code, the user has to check out the code to a local machine from the ITTC network using CVS. In this way, the original code will never be touched and is kept safely in the network. After modifying the code, the user needs to make sure the new version of the code is truthful and correct, then, they can update the new version into the network again using CVS. When CVS is used at the first time in the local machine, a “fs” directory will be created by CVS in this computer. (Note: the directory name has to be called “fs”)

The source code (C++ & Java), and Make files will be checked out into \fs directory. The C++ code (taking care of simulation) is located in “\fs\fibersimSCM” folder, and the Java code (taking care of the GUI) is saved in “\fs\gui.Horizon” folder. There is another folder called “jni” (Java Native Interface), which serves as a bridge between the C++ code and Java code.

Folder fibersim, fibersimRFF, fibersimRFF.nff, gui, and gui.nff hold old versions of the source code. They are no longer used.

The Make files under \fs take care of compiling the source code. There are three of them: Makefile.win, Makefile.linux, and Makefile.alpha, which are used for Windows system, Linux system, and Alpha system, respectively.

After successfully checking out the code onto the local machine, the user needs to compile it in a Command window using the Make files. Compiling creates a “working” folder under \fs directory, which contains executable files, Image files, and Channel files. The Make files allow the C++ & Java code to be compiled to form executable files, create Channel files and Image files, and put all the files together in the \fs\working directory.

As soon as the compiling is done, FiberSim can be run in either Window environment or by typing in commands in the Command window. When the FiberSim GUI pops up, it is time for the user to create or open a project (see section 3.3.1).

Detailed Procedures

When maintaining and modifying the source code of FiberSim in a local computer, WinCVS, GCC (C++ compiler), and JDK (Java compiler) are needed. These software/compiler packages can be downloaded from the Internet. Also, they have been saved to the ITTC network. To download, and use the software/compiler, please read the following.

Step 1: Creating a map to the ITTC projects directory

This step will create a P:\ drive that maps to [\\devnull\projects](#) in the local machine.

How to map the drive.

- 1.) Right click on the “*My Computer*” Icon on the desktop.
- 2.) Select “*Map Network Drive*”
- 3.) Change the Drive letter to **P:**
- 4.) Enter in the “*Folder:*” box: “*\\devnull\projects*”
- 5.) Make sure that the “*Reconnect at logon*” checkbox is checked.
- 6.) Click the “*Finish*” button.

Step 2: Installing An Unzipping Tool

If the local machine has Winzip 8.0 or above, just skip this step. Otherwise, an unzipping tool is necessary for completing the installation.

The Installer is located at “*P:\Fiberspec\windows\Tools\winzip80.exe*”

Execute the installer and follow it’s directions.

This should be fairly straightforward.

Step 3: Installing Java JDK 1.3

Java is an integral part of the program and by installing it, java will make running the code that much easier. Also most of the debugging code is dumped to a command window, so it is important to follow all of the steps so that it will be easier to use the command line tools for debugging the system.

How to install JDK 1.3

- 1.) First execute the installer located at
“*P:\Fiberspcec\windows\Tools\j2sdk1_3_0-win.exe*”
- 2.) When the program asks for where to install the software change the path to
“*C:/jdk*”
- 3.) When installation finishes the command line path needs to be set.

For Windows NT/XP/2000

- a. Right click on the “*My Computer*” Icon on the desktop.
- b. Select “*Properties*”
- c. Select the “*Advanced*” Tab on the top of the window.
- d. Select “*Environment Variables...*”
- e. Inside the box labeled “*User Variables for who*” Click on “*New*”
- f. Enter in the “*Variable Name:*” Field “*PATH*”

- g. Enter in the “*Variable Value:*” Field “C:\jdk\bin”
- h. Click “OK” to finalize everything.

For Windows 95/98/ME

- a. Edit “C:/Autoexec.bat”
- b. Add the line “PATH=%PATH%;C:\jdk\bin”

Step 4: Installing GCC

GCC is a free compiler. It is used because it is also installed on the Linux, and Alpha boxes, and just about any other machine in ITTC.

How to install GCC

- 1.) Open the zip file “P:\Fiberspec\windows\Tools\gcc.zip”
- 2.) Extract the files to “C:/gcc”
- 3.) Edit the environment variables.

For Windows NT/XP/2000

- i. Right click on the “*My Computer*” Icon on the desktop.
- j. Select “*Properties*”
- k. Select the “*Advanced*” Tab on the top of the window.
- l. Select “*Environment Variables...*”
- m. Inside the box labeled “*User Variables for who*” Double Click on “*PATH*”
- n. Add in the “*Variable Value:*” Field “;C:\gcc\bin” to the end of the text there.
- o. Click “OK” to finalize everything.

For Windows 95/98/ME

- c. Edit “C:/Autoexec.bat”
- d. Add the line “PATH=%PATH%;C\gcc\bin”

Step 5: Installing WinCVS

CVS is version control software. It allows the group to work together without too many headaches.

How to Install WinCVS

- 1.) Open the zip file “P:\Fiberspec\windows\Tool\WinCvs120.zip”
- 2.) Extract the files to “C:\temp”
- 3.) Execute the setup.exe in “C:\temp\WinCvs120”
- 4.) Follow default setup instructions.
- 5.) Edit the environment variables.

For Windows NT/XP/2000

- p. Right click on the “*My Computer*” Icon on the desktop.
- q. Select “*Properties*”
- r. Select the “*Advanced*” Tab on the top of the window.
- s. Select “*Environment Variables...*”
- t. Inside the box labeled “*User Variables for who*” Click on “*New*”

- u. Enter in the “*Variable Name:*” Field “*CVSROOT*”
- v. Enter in the “*Variable Value:*” Field “*:local:p:\Fiberspec\CVS*”
- w. Click “*OK*” to finalize everything.

For Windows 95/98/ME

- e. Edit “*C:/Autoexec.bat*”
- f. Add the line “*CVSROOT=:local:p:\Fiberspec\CVS*”

Step 6: Using WinCVS 1.2

Checking out the source code (latest version).

This is useful for the first time that WinCVS (version 1.1 or 1.2) is run, and when things get corrupted.

- 1.) Create the directory “*C:\fs*”
- 2.) Open WinCVS: “*Start -> Program -> GNU -> WinCvs 1.2 -> WinCvs*”.
- 3.) Go to “*Admin -> Preference*”.
- 4.) Type in the path where the CVSROOT is stored: “*P:\Fiberspec\CVS*” in the column “*Enter the CVSROOT:*”, then click “*OK*”.
- 5.) Go to “*Create -> Checkout Module...*”.
- 6.) The directory for checking out needs to be set to “*c:\fs*” and click “*OK*”
- 7.) In “*Enter the module name and path on the server:*” enter “*fs*” and click “*OK*”
- 8.) Wait while it pulls all of the files from the server.

Note: if using a different version of WinCVS, the descriptions in the popping-up windows may be different, but the procedures should be the same.

The source code includes C++ code (taking care of simulation) and Java code (generating GUI).

- C++ code (inside fibersimSCM folder)

1) C++ Header Files (FileName.h)

Barrier.h	BesselFilter.h
BitRand.h	ButterworthFilter.h
ComplexMath.h	Component.h
Consts.h	CustomFilter.h
EnvelopeMatrix.h	extended_ifstream.h
FFTWrap.h	Fiber.h
Filter.h	fsthread.h
getopt.h	IdealFilter.h
Laser.h	MachZehnderModulator.h
Matrix.h	Multiplexor.h
NormalRand.h	NotchFilter.h
OOKGenerator.h	OptAmplifier.h
PhotoDiode.h	Plot.h

PSKGenerator.h	RamanFiber.h
SignalGenerator.h	SimParameters.h
SimulationGroup.h	Simulator.h
Stropts.h	Threadable.h
ThreadManager.h	UniformRand.h
Varlist.h	Vartypes.h

2) C++ Source Files (FileName.c)

Barrier.c	BesselFilter.c
ButterworthFilter.c	Component.c
CustomFilter.c	EnvelopeMatrix.c
extended_ifstream.c	FFTWrap.c
Fiber.c	Filter.c
fsthread.c	IdealFilter.c
Laser.c	MachZehnderModulator.c
Main.c	Multiplexor.c
NotchFilter.c	OOKGenerator.c
OptAmplifier.c	PhotoDiode.c
Plot.c	PSKGenerator.c
RamanFiber.c	SignalGenerator.c
SimParameters.c	SimulationGroup.c
Simulator.c	Stropts.c
ThreadManager.c	UniformRand.c

3) C++ Object Files (FileName.o)

Barrier.o	BesselFilter.o
ButterworthFilter.o	Component.o
CustomFilter.o	EnvelopeMatrix.o
extended_ifstream.o	FFTWrap.o
Fiber.o	Filter.o
fsthread.o	IdealFilter.o
Laser.o	MachZehnderModulator.o
Main.o	Multiplexor.o
NotchFilter.o	OOKGenerator.o
OptAmplifier.o	PhotoDiode.o
Plot.o	PSKGenerator.o
RamanFiber.o	SignalGenerator.o
SimParameters.o	SimulationGroup.o
Simulator.o	Stropts.o
ThreadManager.o	UniformRand.o

- Jave code (inside gui.Horizon folder)

Since there are over 300 files in gui.Horizon folder (Java Source Files and Java Class Files), a list of the file names will not be included here.

Updating code.

After editing the code files in whatever text editor the user chooses (Xemacs, Notepad, Word, etc). The file will show up in a red color, this indicates changes. If there is a “C” next to the file it means that there is a conflict in the code, which must be resolved before committing the code to CVS. The following are the steps involved in updating and comparing files.

- 1.) Right click on the “fs” directory in WinCVS (should be on the left side of the window).
- 2.) Select “*Update Selection...*” CVS will take a little while to check everything and update files. In the bottom right corner will be an area with a listing of everything that occurred. It will show files updated with a “U”, files that have been modified with “M”, and files that have a conflict (modified by more than one person) with a “C”. Conflicts will need to be resolved by editing the code files and removing any conflicts. Also note that the files themselves that have been modified added by the user and are not committed yet will be displayed in red.

Viewing Changes.

Just simply right click on the file and select “*Diff Selection*” This will display a list of changes made to the file in the bottom right area.

Committing Changes

Simply right click on the red file in question and select “*Commit Selection*” This will place the change to the file in the CVS archive and allow for everyone to receive the updating code, which is a lot easier than emailing changes all of time.

Other Uses for CVS (not explained in detail)

- 1.) Branching - If a major change is needed it is sometimes best to branch out and create separate codes for both branches and later merge changes, this allows progress to occur without loosing changes by anyone and without breaking the program altogether. Merging is a tedious process so please try to avoid branching if possible.
- 2.) History Retrieval – If all of a sudden the code starts going crazy the user can use CVS to back track changes and find out when the change that is causing problem was committed. And even back up to that point and start from there.

Step 7: Compiling the source code

After checking out or updating the code from the network, compiling has to be done on the local machine, which creates the executable from the files. There are three Make files in the \fs directory: Makefile.win, Makefile.linux, and

Makefile.alpha. These Make files will take care of compiling the code in different operating systems since the compiling commands are included in the files. There is also one file called “**Makefile**” inside \fs directory, which is created from previous compiling process. Please **delete** it before compiling.

How to compile the code in Windows NT/XP/2000

- 1.) Open a Command Window
 - a. Click Start
 - b. Select Run
 - c. Type in “*command*”
 - d. Click OK
- 2.) Change directory – Type “cd C:\fs
- 3.) Copy Makefile – Type “copy Makefile.win Makefile” ↵
- 4.) Compile the code – Type “make” ↵
- 5.) The “make” command will create a “working” directory under \fs. Also, an “icons” folder holding the Image files will be created under “working” directory (see section 2.1.2). The user will see the compiler popping up a question during compiling: Icons is a filename (F) or a directory (D)? Just type D or d. The process will continue on.

The whole process should take a while. The user will see a list of compiled files showing up in the screen. After compiling, a “working” directory will be generated automatically under C:\fs. Three executable files (FiberSim.exe, FiberSim.jar, fsloader.dll), Channel files, Image files, and FSP files generated from compiling are save under the “working” folder.

Sometimes the user may need to clear all of the object files and the execution files, simply type “make clean” and then “make”.

Step 8: Starting FiberSim

After the compiling is done, FiberSim is ready to run. To run the code either double-click on the “*FiberSim.jar*” file in the working directory or type in the following command “*java -jar FiberSim.jar*” in the Command window under “C:\fs\working. Either command will start the program.

Further information are located on the “\fs\docs” directory (this documentation may not be up-to-date).