SimLIGO 020704: How To Guide

Matt Evans

September 25, 2002

1 Introduction

1.1 Input Files

The primary inputs to SimLIGO, box files aside, are the detector configuration file (DCF, e.g., "LHO4k.mcr") which contains detector specific information, the E2E database file (DBF, "e2eDB.mcr") which contains non-detector specific settings, and the parameter file (PAR, "Detector.par") which contains time varying simulation parameters.

The detector simulated by SimLIGO can be changed by changing which DCF is included at the top of the DBF. The DCF provides information about each sub-system of the detector. In particular, all of the optical properties of the optics as well as their suspension systems are included in this file. Also included are the parameters describing the input beam (power, modulation frequencies, etc.) and the geometrical parameters of the interferometer (optical path lengths, out-of-plane reflection angles, etc.). At present, only the LHO 4km detector configuration is available.

Since SimLIGO's primary purpose is to study noise propagation, the time consuming process of lock acquisition is not simulated. Instead, the power in the IFO is allowed to build before the control loops are engaged. After control has been established, the various whitening and dewhitening filters are sequentially engaged and the noise sources are enabled. Much of the sequential enabling of filters and noise sources is done in the PAR file (see "Sequencing" near the end of the file). The whitening filters are an exception to this rule; their turn-on times are set in the DBF (e.g., "Controller.t_WhiteOn = 0.58").

1.2 The Fiddle Box

A box commonly used to inject signals and change gains is the "Fiddle" box. Fiddle takes a single input, multiplies it by a gain, and adds to it linear and/or sinusoidal functions of time. The parameters of a fiddle are gain, offset, slope, amp, freq, phase and reset. The output is given by

$$S_{out} = \text{gain} * S_{in} + \text{offset} + \int_{t_0}^{t} \text{slope}\ dt + \text{amp} * sin(\int_{t_0}^{t} \text{freq}\ dt + \text{phase}) \quad (1)$$

where t_0 is the last time "reset" was set to a non-zero value. Note that "offset" can be useful as an input point for modeler_freq.

1.3 Running Modeler

A little setup is necessary before running modeler. The first thing to do is find the E2E install directory, which will be left as an exercise to the reader, and make sure that the version of modeler is 020704 or later. (modeler reports its version number when run). The environment variable "E2E_PATH" will also need to be set to "<SIM_LIGO_BASE>/lib:<E2E_BASE>/lib" where <SIM_LIGO_BASE> is the full path to the SimLIGO directory (i.e., ends with SimLIGO) and <E2E_BASE> is the E2E install directory (i.e., such that <E2E_BASE>/bin/modeler is the modeler binary).

When running modeler, it is convenient to make an input record file (IRF, e.g., "Detector.in") that contains the responses to modeler's interactive interface. The IRF contains the duration of simulation run as well as the frequency and destination of output data. To run modeler with the default IRF execute the following command line in the SimLIGO/Boxes/ directory:

```
% cat Detector.in - | modeler -bin
```

The output data and settings record files (DAT and SET, e.g., "Detector.dat" and "Detector.set") will be created in the SimLIGO/Data/ directory.

1.4 Matlab

Start matlab in the SimLIGO/Matlab/ directory.

1.5 Time Domain Data

To load the data from a simulation run using the default IRF use

```
>> [dat, lgnd] = modLoad('../Data/Detector.dat');
```

The plotting function "modPlot" uses arrays of descriptive column selection structs (DCS). The default collection of DCSs is initialized with

```
>> DetectorCols
```

Each DCS contains a channel name, unit, scaling or data column combining expression, column index or indices, and a cell array of arguments to pass to plot (this is usually just the color of the line). The DCSs used to plot the outputs of each subsystem are described at the bottom of the subsystem descriptions given below.

The modPlot function requires two arguments: the data array and a plot description. The "plot description" is a cell array of struct arrays. The shape of the cell array will determine the placement of plots (i.e., a 2x2 cell array will result in a plot with the layout given by subplot(2, 2, 1)). The contents of the cell array determine what is plotted in each plot. An example usage of modPlot is

```
>> modPlot(dat, {s_aDC; s_dof; s_err; s_freq}, [0.5, 1])
```

where the third argument is the optional time interval. The time interval can be a single number, interpreted as the start time, or a pair of values for the start and stop times.

1.6 PSD Data

A few functions are provided for loading PSD data from modeler. These functions are "modLoadPSD", "modSplicePSD", and "loadASQ". The last of these uses the first two to load the four part PSDs generated by SimLIGO as follows

```
>> pDat = loadASQ('../Boxes/Detector.DigitalISC.LSC.DARM_PSD');
```

With no output argument loadASQ plots the loaded data.

A few matlab functions are provided to study the noise curve.

```
>> pDat = loadASQ(...);     % load PSD (without return args to plot)
>> [f, a] = sensASQ(pDat); % make noise curve (without return args to plot)
>> darm;                    % plot matlab model noise curve
```

2 The Detector

There is only one parameter defined at the Detector level; Detector.AnalyzeOn is provided for use with modeler_freq. The value of AnalyzeOn (a data_in) is routed directly to the ANALYZE_FLAG output, which is used by modeler_freq to begin its analysis. The default PAR file sets AnalyzeOn to 1.0 after one second of running. This allows time for the power levels in the IFO to equilibrate before analysis begins.

3 PSL-IOO

The PSL offers no dynamically adjustable parameters. The DCF sets the input power level, modulation frequency, and sideband-order.

The IOO is responsible for producing intensity and frequency noise representative of the noise incident on the recycling mirror (RM). Noise with the approximate spectra for both of these noise sources is produeced in Detector.IOO.Noise. The amplitudes of these noises (FreqNoiseAmp and IntenNoiseAmp) can be found in the DBF under the Noise heading. Frequency and intensity noise are enabled after 0.7 seconds.

The frequency feedback path from the IFO is also simulated in the IOO. It is (and must be) enabled simultaneously with the mechanical feedback servos. This happens at 0.5 seconds into the simulation run (set at the top of the PAR file).

Detector.IOO

s_freq	frequency servo, noise and total

4 COC

4.1 Optics

Each of the as-built core optics is described in some detail in the DCF. While these as-built parameters may provide an accurate simulation, asymmetries and imperfections in the optics also introduce a great deal of complexity. To mitigate this effect, the COC can be made more or less “perfect” through adjustable parameters found in the DBF. Each of the parameters in the following table makes the corresponding feature “perfect” (i.e., no asymmetry or mismatch) when set to 0.0 and "as-built" when set to 1.0.

asym_BS	beam-splitter (BS) transmission-reflection asymmetry
tran_ITM	input test mass (ITM) transmission asymmetry
tran_ETM	end test mass (ETM) transmission asymmetry
loss_ITM	ITM loss asymmetry
loss_ETM	ETM loss asymmetry
curv_RM	RM front surface radius of curvature
curv_BS	BS front surface radius of curvature
curv_ITM	ITM front surface radius of curvature
curv_ETM	ETM front surface radius of curvature
coil_RM	RM coil efficiency mismatch
coil_BS	BS coil efficiency mismatch
coil_ITMx	ITMx coil efficiency mismatch
coil_ITMy	ITMy coil efficiency mismatch
coil_ETMx	ETMx coil efficiency mismatch
coil_ETMy	ETMy coil efficiency mismatch

4.2 Sensing: Analog ISC

The sensing system contains two categorically separate noise sources: shot noise and electronic noise. The existence of shot noise in the photodetectors is controlled with COC.ElecOut.SHOT_NOISE, which is found in the DBF. SHOT_NOISE can be set to 0 (no shot noise), 1 (simple stationary shot noise model), or 2 (complete shot noise model).

The amplitude of electronic noise in this subsystem is set with COC.ElecOut.NoiseAmp, also found in the DBF. As with other "NoiseAmp" parameters, 0.0 results in no noise and 1.0 results in the "as-accurate-as-possible" noise amplitude, with intermediate values resulting in intermediate noise levels.

Detector.COC.ElecOut.LSC

s_aDC	analog DC signals
s_aD1	analog in-phase RF signals
s_aD2	analog quad-phase RF signals
s_rDC	raw DAC output DC signals
s_rD1	raw DAC output in-phase RF signals
s_rD2	raw DAC output quad-phase RF signals
s_aDC	digital DC signals
s_aD1	digital in-phase RF signals
s_aD2	digital quad-phase RF signals

5 Digital ISC

5.1 LSC

There is no noise in the digital system, so the only adjustable parameters are the loop gains, which are found in the PAR file (DARM_CON.gain, CARM_CON.gain, etc.). However, the stability of the control loops is rather fragile and changing gains, especially in the CARM and PRC loops, may break or prevent IFO lock.

There are two sets of Fiddles, one on the error signal side (pre-filtering, DARM_ERR, CARM_ERR, etc.) and one on the control signal side (post-filtering, DARM_CON, CARM_CON, etc.) of the control filters.

Detector.DigitalISC.LSC

s_err	error signals (post-fiddle)
s_con	control (post-fiddle)
s_dof	actual DOF errors
s_mech_dof	DOF errors ignoring input beam frequency offset

5.2 ASC

The digital ASC system is not implemented in this version of SimLIGO.

6 LOS

6.1 Mechanics

A representative thermal noise spectrum is included in the mechanical simulation. The thermal noise amplitude can be set in the DBF (ThermalNoiseAmp) to values between 0.0 and 1.0. Its is sequenced to turn on with the control loops at 0.5 seconds (see PAR file).

Three sets of modified Fiddles are present in this box: “fs” at the suspension point, “ff” at the center-of-mass force input, and “fp” at the optic position output. The difference between these mechanical Fiddles and normal Fiddle boxes is that they have an extra parameter, “dim,” that selects the dimension to be fiddled with. Settings for “dim” are 0-5 corresponding to x, y, z, thetaX, thetaY, thetaZ.

Detector.COC.Suspensoins.Sus<optic>.Mech
<optic>= RM, BS, ITMx, ITMy, ETMx, ETMy

s_mech_<optic>	summary optic position (z, thetaX, thetaY)
s_mechPos_<optic>	optic position (x, y, z)
s_mechAng_<optic>	optic angles (thetaX, thetaY, thetaZ)
	(the following are in the optic’s coordinates)
s_susPos_<optic>	suspention point position (x, y, z)
s_susAng_<optic>	suspention angles (thetaX, thetaY, thetaZ)
s_basePos_<optic>	ground position (x, y, z)
s_baseAng_<optic>	ground angles (thetaX, thetaY, thetaZ)

6.2 Digital Suspension Controller

Noise in the local sensors (shadow and optical lever) is generated in the LOS. The amplitude of these noise sources are set with InputShadow.NoiseAmp and InputOptLev.NoiseAmp, both ranging from 0.0 to 1.0. The local sensors also have whitening filters (InputShadow.WHITE_ON and InputOptLev. WHITE_ON in the DBF) and whitening gains (InputShadow.WhiteGain and InputOptLev. WhiteGain in the DCF).

The actuation chains, which participate more directly in the detector noise characteristics, are also contained in the LOS. Noise in the actuation chain is accessed through OutputCoil.NoiseAmp in the DBF. The dewhitening filters are set to turn on after 0.58 seconds (see OutputCoil.t_WhiteOn in the DBF) and the coil-drivers can be toggled between Run and Acquire modes with OutputCoil.ACQUIRE_ON (also in the DBF).

Detector.COC.Suspensoins.Sus<optic>.Mech
<optic>= RM, BS, ITMx, ITMy, ETMx, ETMy

s_aSens_<optic>	analog local sensor signals
s_dSens_<optic>	digital local sensor signals
s_aOL_<optic>	analog optical lever signals
s_dOL_<optic>	digital optical lever signals
s_err_<optic>	local damping + ISC error signals
s_con_<optic>	local damping + ISC control signals
s_dCoil_<optic>	digital coil signals
s_rCoil_<optic>	raw analog DAC output signals
s_aCoil_<optic>	analog coil signals

7 Environment

7.1 Seismic Noise

Generation of seismic noise takes place in three steps: generation of white noise with proper coherence between locations, filtering to produce ground motion spectra, filtering through the appropriate stack model (including orientation) for each location. Seismic noise, like thermal noise, can be adjusted in the DBF (SeismicNoiseAmp) and is sequenced in the PAR file to turn on at 0.5 seconds.