LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Preliminary Design	**LIGO-T970172-A - D**	3/9/98

Global Diagnostics System Preliminary Design

Rolf Bork, Mark Pratt, David Shoemaker, Daniel Sigg

Distribution of this draft:

LIGO laboratory

This is an internal working note
of the LIGO Project.

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (818) 395-2129
Fax (818) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW: http://www.ligo.caltech.edu/

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the preliminary design of the Global Diagnostics System (GDS). The information in this document supersedes that presented in the design requirement document (T960107-00), the conceptual design (T960108-00) and interferometer diagnostics document (T970078-00).

1.2 Scope

This document details the current status of the global diagnostics system design effort. The GDS provides a means to diagnose the interferometer system and support the operations; access to the GDS is provided from the control room or any other display station located on the site. The GDS is meant to deliver real-time status information of the detector performance and to support initial installation and shake-down of the detector. It includes tools to view data on-line, to do fourier transformations on a large number of channels, to search for abnormal behavior and effects of the instrument while it is running, to do invasive diagnostics tests using an excitation system, and to record statistical summaries of a detection run (see Fig. 1). The GDS does not include a data acquisition system, but rather relies on the CDS DAQ system for gathering data at audio frequencies (T960009-A and T970136-00). It uses the remote-controlled cameras (owned by ASC) and digitizing oscilloscopes (owned by their subsystems) to obtain visual information and to look at wide bandwidth signals, respectively. It also does not provide back-end computing resources, but uses workstations in the control room (T970171-00) and computing power provided by the DAS (T970159-04 and T970160-06).

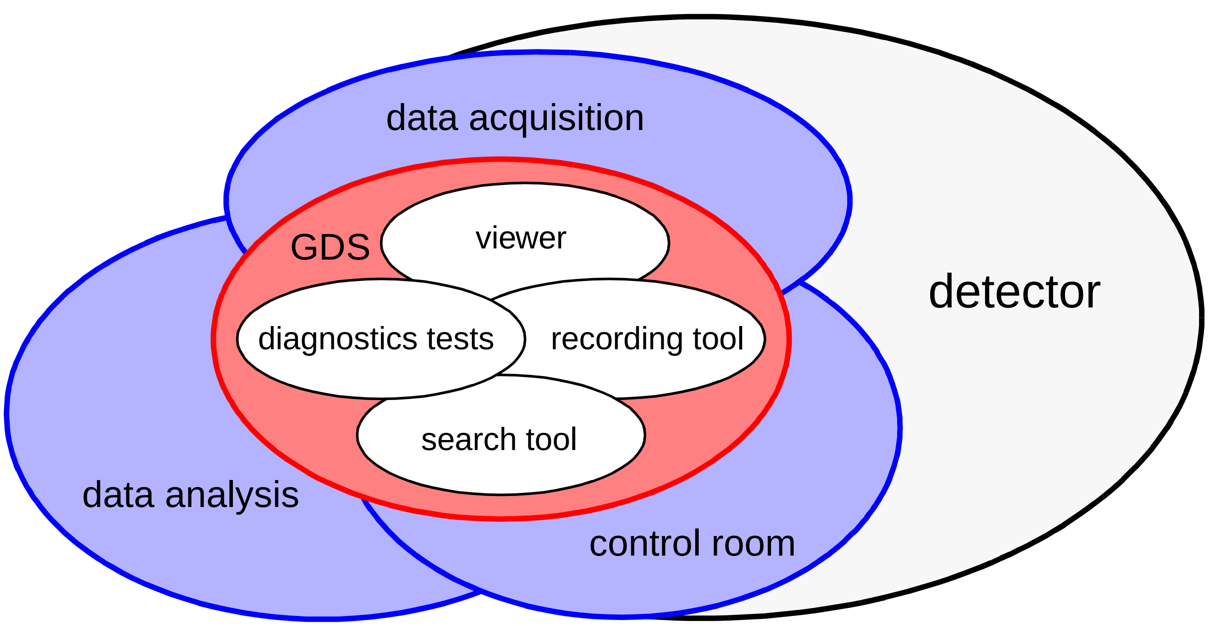


Figure 1: System overview.

1.3 Goals

The goals of the GDS can be summarized as follows:

- Assist the operators in the control room and in the experimental areas to successfully run the experiment. This means they must be able to view data in real-time, to make diagnostics tests involving an artificial excitation, to verify and monitor the performance of the instrument while running and to have access to a diagnostics summary of the detector over the past few months.
- Provide immediate answers (minutes rather than days or months) to important questions such as: what is the quality of the gravitational-wave data written to disk? And, are all of the subsystems working properly?
- Assist the operator in establishing and in automating diagnostics procedures which are used to diagnose and find problems with the instrument.
- Assist the user in learning about the behavior of the instrument, in classifying abnormal environmental events, in identifying the exact machine state, in correlating the signals of different sensors and, ultimately, in reducing the large amount of measured data to a set of relevant and comprehensible statistical quantities.

The proposed design is based on the following principles:

- Build a scalable system which can grow with the need of the users and which is able to incorporate the know-how accumulated during the commissioning and the first year of operations.
- Build a system which is incrementally useful, so that a bare-bone minimum system can be used during the early installation phase (as early as end of 1998) and then be extended piece-by-piece during the rest of the commissioning phase.
- Build a modular system where different components only communicate with each other when absolutely necessary. This will simplify the implementation of different diagnostics components in parallel and, hopefully, avoid dependencies which would render the complete system useless, only because one of its components isn’t ready yet.
- Provide a high-performance infrastructure of high versatility to access, transmit and store data, so that diagnostics modules can concentrate on their tasks, rather than spending their time reinventing and optimizing the same I/O routines again and again.
- Prefer commercially available systems over in-house developments to minimize time and costs of implementation.
- Build an expandable system to make it possible to accommodate future ideas — such as a system identification module for implementing adaptive control algorithms, or such as a reduced data set module which analyzes and concentrates the data stream in real-time before writing it to disk/tape — with minimal changes to the old system.

1.4 Definitions and Acronyms

The following acronyms are used in this document:

AF	Audio Frequency	I/O	Input / Output
AM	Amplitude Modulation	IOO	Input Optics
API	Application Program Interface	ISC	Interferometer Sensing and Control
ASC	Alignment Sensing and Control	LHO	LIGO Hanford Observatory
CDS	Control and Data Systems	LLO	LIGO Livingston Observatory
COC	Core Optics Components	LSC	Length Sensing and Control
COS	Core Optics Support	NTSC	Never The Same Color
DAQ	Data Acquisition	PEM	Physical Environment Monitor
EM	Electro-Magnetic	PSL	Prestabilized Laser
FPI	Fabry-Perot Interferometer	RF	Radio Frequency
FFT	Fast Fourier Transform	RTDD	Real-time Data Distribution
FM	Frequency Modulation	SEI	Seismic Isolation
GDS	Global Diagnostics System	SUS	Suspension System
GW	Gravitational Wave	SYS	Detector System Engineering
IFO	(LIGO) Interferometer	TBD	To Be Determined

When referring to the front- and back-end of the diagnostics system, we distinguish between the hardware working on the data in real-time and the workstations used to display, store and analyze the data after it is measured, respectively.

1.5 Basic Concepts

An overview of the global diagnostics system and its integration with other subsystems can be seen in Fig. 2.

1.5.1 Excitation System

The GDS relies on the detector subsystems to provide a means to apply an excitation. It, however, provides software drivers for remote-controlled function generators when analog signals are required. For digital subsystems these function generators are implemented entirely in software by the GDS and transferred to the digital subsystems through a digital link. To be able to synchronize excitation drive signals with the measured effects, each of them is read-back by the DAQ system and, therefore, is available to the GDS as an additional DAQ channel. For a more detailed description see section 3.4.3.

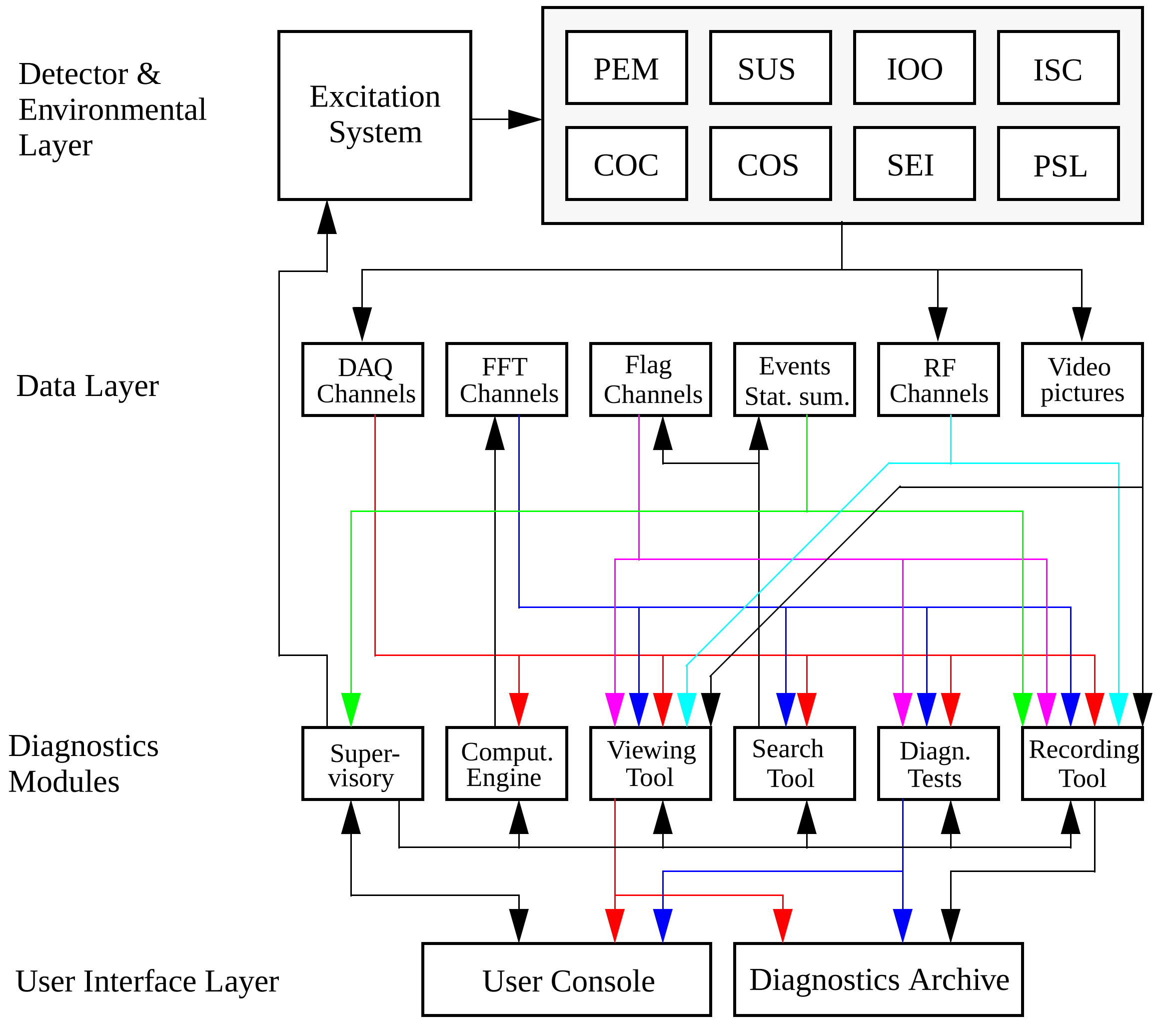


Figure 2: Overview of the global diagnostics system.

1.5.2 Data Layer

Data are available to the GDS in several flavors:

i) Channel data at audio frequencies: data which have a fixed sampling frequency and conforms to the data rates of the DAQ system. Channel data are available directly from the DAQ system (DAQ channels), or as the results of the FFT preprocessor (FFT channels) or the search tool (flag channels). FFT channels are the real-time FFTs of a set of DAQ channels, whereas flag channels are generated by the trigger outputs of the search engine.

ii) Video images: Video cameras connected to video monitors in the control room will look at laser beam profiles at the extraction ports and on the test masses, and other sources of visual information. The GDS will be able to store video frames for further analysis.

iii) Wide-bandwidth data: Fixed and mobile digitizing oscilloscopes — remote-controlled from the control room — will be used to look at RF signals at selected points in subsystems. Specifically, in LSC, ASC, PSL, and IOO, signals prior to the mixer input and at signals higher in frequency than the maximum Nyquist frequency of the DAQ system. These data will be available to the GDS.

iv) Events and statistical summaries: The search tool analyzes data channels important to monitor the detector performance and generates events whenever an abnormal behavior is detected. The GDS keeps a record of the current events and stores them in event queues. Statistical summaries of the measured data are calculated while the interferometer is running to record a diagnostics history.

1.5.3 Diagnostics Modules

1.5.3.1 Data Preprocessing

The GDS performs Fourier transformations in real-time (via a ‘computing engine’) of a substantial set of DAQ channels. For a more detailed description see section 2.1.2.

1.5.3.2 Viewing Tool

The viewing tool provides the capability to look at any data channel on-line and perform simple analysis and triggering tasks. Typically an operator selects the channels he/she wants to investigate and displays them on the diagnostics console. The preconfigured analysis tools allow for filtering, decimation, differentiation, averaging and histogram generation. For a more detailed description see section 2.2.

1.5.3.3 Search Tool

To give the operator in the control room an immediate measure to judge whether good data are being taken, the search tool performs an on-line search for events and circumstances which affect the detector performance such as enhanced seismic activity. The search is done using a trigger system implemented in software. The output of a trigger can either be used to generate an event when active or to directly write a flag channel. Flag channels can be used to implement sophisticated trigger capabilities for the viewing tool and the diagnostics test module, since they are treated like normal data channels. For a more detailed description see section 2.3.

1.5.3.4 Diagnostics Tests

Diagnostics tests such as a swept sine test are performed as scheduled or needed to determine noise coupling terms and other interesting interferometer parameters. These tests typically include an excitation. The diagnostics tests also support the initial installation and the shake-down the detector. They can be used both to help localizing unknown noise sources and to take snapshots of a dynamical behavior such as a locking attempt of the interferometer. For a more detailed description see section 3.

1.5.3.5 Recording Diagnostics Summaries

In order to make it possible to analyze physical trends in the environment and in the detector performance, the DAQ system stores trend frames of highly decimated and preconditioned data to disk. These data have a very a small footprint and can be kept on disk for extended periods of time. It is also stored during periods when the detector is down and no GW data is taken, so that the environmental record stays complete. For a more detailed description see section 2.4.

1.5.4 User Interfaces

1.5.4.1 Diagnostics Console

The diagnostics console allows to look at on-line data, to start and stop a diagnostics test, to view the results of a diagnostics test and to monitor the performance of the instrument. It is also responsible for interfacing the diagnostics supervisory control.

1.5.4.2 Diagnostics Archive

The diagnostics archive will be used to store a log of the events found by the search tool, to keep snapshots of data, to store both the data and the result of a diagnostic test, to keep a record of the diagnostics history and store diagnostics parameters. It consists of a large disk for fast data access and a tape unit to backup old data sets (equipment shared with the DAQ system).

1.5.5 Software Standards

The diagnostics system follows LIGO standards for programming languages, analysis software packages and file formats. Generally, the following guidelines apply:

Description	Standard
Programming language for VME hosted processors	C/C++
Programming language for UNIX based workstations	C/C++
Supervisory control	EPICS
Graphical user interface	TBD
General purposes analysis software package	Mathematica/Matlab
Data file format	netCDF/ASCII
Graphics file format	Postscript

1.6 Implementation Issues

1.6.1 Relations to Other Subsystems

The global diagnostics system defines its own set of requirements (see section 2 and 3). However, in a lot of cases equal or similar functionality is already provided by other subsystems. Generally, the implementation of the GDS will use already existing solutions wherever possible and concentrate on developing the components which are unique to the diagnostics systems. A list of functions, their solution and their providers is presented below:

Function	Solution	subsystem
data acquisition	ADCs, reflected memory	DAQS
data distribution	reflected memory, network, disk	DAQS
data storage, archive	frame builder, disk farm, tape	DAQS
viewing tool	xmgr, DaDisp	DAQS
recording tool	frame builder, trend frames	DAQS
performing FFTs	computing engine, DSPs	DAQS
search tool	front-end CPUs accessing the reflected memory	GDS
diagnostics test tool	supervisory control, software routines, test templates	GDS
excitation system	digital waveforms, analog signals, remote-control of arbitrary function generators	GDS
test equipment	digital oscilloscopes, video cameras, arbitrary function generators, etc.	subsystems
back-end computing	control room workstations, super computer	CDS / DAS
back-end analysis tools	routines for Mathematica, Matlab and C/C++	GDS

1.6.2 Priorities

Since the GDS is an important tool in helping to shake down the interferometer, parts of it have to be ready at an early time. The following list outlines the implementation priorities of each task or subsystem implemented by the GDS:

Hardware		Software	
Subsystem / Task	**Priority**	**Subsystem / Task**	**Priority**
Diagnostics test engine	high	Basic libraries (e.g. channel IO)	highest
Excitation system	high	Diagnostics tests	high
Search engine	medium	Excitation system	high
Computing engine	low	Search tool	medium
		Graphical user interface	low
		Computing engine (FFT)	low

2 REAL-TIME STATUS INFORMATION

The goal of the real-time status information subsystem is to assist the operator in acquiring the accurate and current status of the detector operations. The basic tasks which will be supported by the proposed system are:

- Monitor the quality of the gravitational wave data as it is written to disk.
- Inspect any data channel in real-time in order to judge whether the measured signals are looking ok; or, when there is a problem, help in localizing and finding the cause of it.
- With a system as complicated as LIGO too many independent things can go wrong and an operator might not realize it for quite some time. In order to make sure that the operator in the control room has a complete picture of the performance of the apparatus, automatic software probes will be supported. They will inspect the data of the detector in real-time and automatically trigger an alarm when values are outside of specifications.
- Sometimes it is important to know the history of the physical environment in order to make sense of the current behavior of the apparatus and in order to establish trends. Rudimentary data conditioning, compression, recording and storage capabilities will be supported directly by the GDS (this is not in competition with the full featured DAQ system).

2.1 DATA CHANNELS

The common denominator of all real-time status information tools is that they must have access to the measured data shortly after it is taken. We propose to develop a common infrastructure for accessing these data in real-time. Because of the usefulness of the frequency domain for a large number of diagnostics methods, such as analyzing noise characteristics or looking for lines at a fixed frequency, we also propose to perform Fourier transformations on many channels acquired by the DAQ system and make them available to all of the GDS.

2.1.1 Data Acquisition

The following section describes the data available to the global diagnostics system. It can be categorized into three sub groups: audio-frequency channels, RF channels and video pictures. Audio-frequency channels are limited to a sampling rate of 16384 Hz and are gathered continuously in real-time, whereas RF channels typically have a much higher bandwidth and are probed by taking snapshots.

2.1.1.1 Audio-Frequency Channels

For sampling and collecting signals at audio frequencies the global diagnostics system fully relies on the DAQ system (LIGO-T970136-00). The maximum sampling rate is 16384 Hz per channel and the length of a data block is $1/16$ sec. The maximum time delay between the time the signal is sampled and the time the data are available for the global diagnostics system shall be no more than 65 ms.

Most detector subsystems provide analog signals which can be easily sampled by the DAQ system. Since it is not practical for digital subsystems to provide analog representations of all internal digital data paths, we require that digital signals up to 16384 Hz can be fed directly into the DAQ system without loosing precision or timing accuracy. This makes it possible for the diagnostics system to obtain fully synchronized exact numbers of internal data and compare it with other interferometer outputs in real-time.

2.1.1.2 High Frequency Channels

To be able to look at high frequency signals (10 kHz – 500 MHz) and fast transients (1 ns – 10 μsec) commercially available digitizing oscilloscopes will be used (e.g. Tektronics TDS 640A). The scopes will be connected to the CDS network (National Instruments GPIB-ENET, ethernet GPIB controller) and will be remote-controllable from the control room using a commercially available software package (National Instruments LabView). This will be a stand-alone system which will hopefully require only a minimum of internal software development. The scopes will be provided by the corresponding subsystems. Similarly, one can use the GPIB/Labview combination to interface and to control rf network/spectrum analyzer and any other equipment with GPIB capability. GPIB equipment will generally rely on the workstation clock to synchronize data, or implement its own GPS clock for time critical data.

2.1.1.3 Video Pictures

The video system which measures beam profiles and looks for beam spots on the mirrors will be a commercially available system and will be provided by ASC (LIGO-T970061-00). This will be a stand-alone system which requires no internal software development. The video signals will be available in the control room as standard NTSC, and will also be acquired by the CDS Control and Monitoring system for viewing on computer screens and for later analysis. A video recorder with the capability to add a time stamp can be used for recording.

2.1.2 Preprocessed Data Channels

2.1.2.1 Reconstructed Signals

Some of the signals which are of interest for analyzing and monitoring are not sampled directly. Either they have to be reconstructed from other signals, and/or they were fed through whitening filters to match the dynamic range of the ADCs. For instance, at the antisymmetric port there are between 4 and 8 photodiodes for length sensing sharing the optical power; all of them are using whitening filters. However, for most diagnostics analyses only the sum signal of all their Q-phases — generally denoted as the GW channel — has any significance.

We therefore require to have the capability for signal reconstruction at the front-end. The reconstruction should occur as the first step after the data is sampled, so that these channels can be treated and viewed exactly the same way as any other sampled data channel. Signal reconstruction is typically just a linear superposition of other channels followed by an optional unwhitening filter. The following signals have to be reconstructed and assigned their own data channel:

i) The 4 length sensing signals and the 2 corresponding signals in the orthogonal RF phase,
ii) the 14 alignment signals of the test masses and the input beam direction,
iii) the sum of the 4 coil current read-backs of each test mass describing their axial motion.

Both *i)* and *ii)* are describing signals which will already be reconstructed by their corresponding subsystem in order to implement closed-loop control for length and alignment. It is therefore possible to pull these signals directly from the subsystem, rather than let the DAQ system reconstruct them by its own.

2.1.2.2 Fourier Transforms

In many cases, the power spectrum of a channel is a more useful view of the data. The global diagnostics system implements a number cruncher which will calculate FFTs of many audio-frequency channels in real-time. The Fourier transforms are treated as additional data channels which are available for viewing, triggering or performing diagnostics tests.

We ask for the capability to make real-time FFTs on a number of data channels described in an FFT channel dictionary. These FFTs are then treated as additional channels — called FFT channels. The maximum number of *fast* FFT channels (sampling rate of 16384 Hz) is given by the corresponding channel count, i.e., ~150 per interferometer — this determines how much has to be invested in processing power. We allow as many as 500 FFT channels in total (about the number of channels per interferometer) — this determines how much has to be invested in memory.

Description	Requirement	Unit
Number of FFT channels	< 500	
Number of *fast* FFT channels	< 150	
Number of points in the FFT	1024 or 16384	points
Window function	Hanning	
Refresh rate	16, 1, 1/16 or 1/256	Hz

The dictionary of the FFT channels contains a list of records, each describing the parameters of one FFT channel. This record consists of the following parameters:

Parameter	Description
ChannelName	name of the FFT channel (ASCII)
ChannelNumber	ID number of the FFT channel
DataChannel	data channel where the time series is taken from
FFTPoints	number of points used in the discrete Fourier transform
RefreshRate	rate at which the Fourier transform is recalculated

The above definition makes it in principle possible to do one real-time FFT on each data channel. However, one might decide that it is more useful leaving a couple of channels without FFTs and having a couple of them with more than one making use of different frequency spans and resolutions. The larger part of this FFT channel dictionary shall be fairly static, but can be changed when necessary. The reason for trying to freeze its definition as far as possible is that a trigger might have been written assuming that a certain FFT channel is there with a given refresh rate and resolution. (So, changing the FFT channel might require rewriting the piece code which depends on it).

The above solution is both flexible and scalable. For instance, if one wants to increase the number of FFT channels, one can simply install more processing power — as long as the memory is large enough to carry all the required channels.

It should be noted that this section only covers Fourier transforms done in real-time. The global diagnostics system will also provide a diagnostics test (see section 3.2.7.2) which will be able to make power spectra of a few channels with user defined frequency span, resolution and averages (these FFTs are calculated on the back-end).

2.1.2.3 Flag Channels

Flag channels are channels generated by the search tool of the diagnostics system (see section 2.3). They typically summarize the health of a detector subsystem or the occurrence of abnormal environmental conditions. Since flag channels can be generated from audio-frequency channels or their Fourier transformation, they are delayed at least 125 ms, but typically not more than 175 ms. If the delay is larger it has to be added to the channel information.

Description	Requirement	Unit
Maximum number of flag channels	100	
Maximum time delay of input	125	ms
Typical time delay of output (relative to input)	50	ms
Maximum sampling rate	16384	Hz

2.1.3 List of Data Channels

See Appendix A for a complete list of the DAQ channels.

2.2 VIEWING TOOL

The GDS has to support the visualization of the measured data at many different places. Additionally, it also has to support the capability to listen to an acquired data channel by using standard audio equipment. We distinguish real-time and off-line viewing and listening, respectively. Off-line viewing and listening is usually coupled with the need to analyze the data and is best done using a commercial software package, such as Mathematica or Matlab. On the other hand real-time viewing and listening usually requires a data throughput which is not available with these high-end packages. Furthermore, the analysis capabilities of these packages are most likely overkill for real-time purposes. In this section we present the tools necessary to view and to listen to data channels acquired by the DAQ system. We do not include the digital oscilloscopes and the video camera system here, because they relay on commercially available software packages to handle the data. The viewing tool described below is then limited to the following tasks:

- ➢ Provide a real-time view of data channels acquired by the DAQ system.
- ➢ Provide real-time audio for data channels acquired by the DAQ system.
- ➢ Provide monitors of the current performance of the instrument.

Of course, the viewing tool will also support storing, retrieving and printing of the displayed data (very similar to the capabilities of any decent scope).

2.2.1 Basic Requirements

The real-time viewing tool has the capability to look at interferometer and environmental channels with almost no delay, i.e. the viewed data is not more than 200 ms delayed. It can look at channels sampled by the DAQ system — both time traces and power spectra. Simple data preprocessing, simple statistical analysis and simple triggering can be performed on these channels, before they are displayed. Sometimes, it is useful to look at the same time at more than one view of a single channel, e.g. the filtered time trace and the averaged power spectrum. Conceptually, every viewing channel has exactly one data channel associated with it, whereas data channels can have none, one or several views at the same time. To limit the amount of CPU processing power needed for the viewing tool the maximum number of simultaneous viewing channels will be around 50 (this includes permanent diagnostics monitors and should support more than one concurrent user).

Description	Requirement	Unit
Maximum delay	200	ms
Channels	DAQ channels FFT channels flag channels	
Maximum simultaneous viewing channels	~50	
Data preprocessing (audio-frequency channels)	multi-pole filters decimation/smoothing differentiation averaging	
Statistical analysis (audio-frequency channels)	histogram mean value, standard deviation, r.m.s., maximum, minimum, highest, lowest	
Trigger capability (audio-frequency channels)	simple level trigger	
Storage and print capabilities	snapshots and data traces on user request	

2.2.2 On-Line Analysis Capabilities

Data preprocessing is an important capability of the real-time viewing tool. It allows the operator to separate useful information from background noise and irrelevant information. A lot of signals acquired by the DAQ system have a strong frequency dependence. Sometimes, one is interested in the higher frequency components and wants to low-pass filter the data before viewing it; at other times, one would like to look at low frequency fluctuations and wants to smooth and bin the data before viewing it. Also, some features may only show up after taking averages over long durations. Then, one might only be interested in a statistical summary taken over a certain time. Or, one is interested what happens in one channels when something else happens in an other channel. The following section describes the proposed on-line analysis and triggering capabilities of the real-time viewing tool in detail.

2.2.2.1 Data Preprocessing

For each raw data channel additional preprocessing steps can be implemented (see Fig. 3): two serial filter stages can be configured as either high pass, low pass or notch using arbitrary or predefined corner frequencies.

Description	Requirement	Unit
number of filter stages	2	
type of filters	multi order low-pass or multi order high-pass or multi order notch	
predefined corner frequencies	0.01, 0.03, 0.1,... 3k.	Hz

After the filter stage the data can be either smoothed or binned by a given amount of channels. Two optional differentiation stages are provided to simplify the search for peaks and to monitor the rate of change, rather than the absolute values. They proceed an optional histogram stage

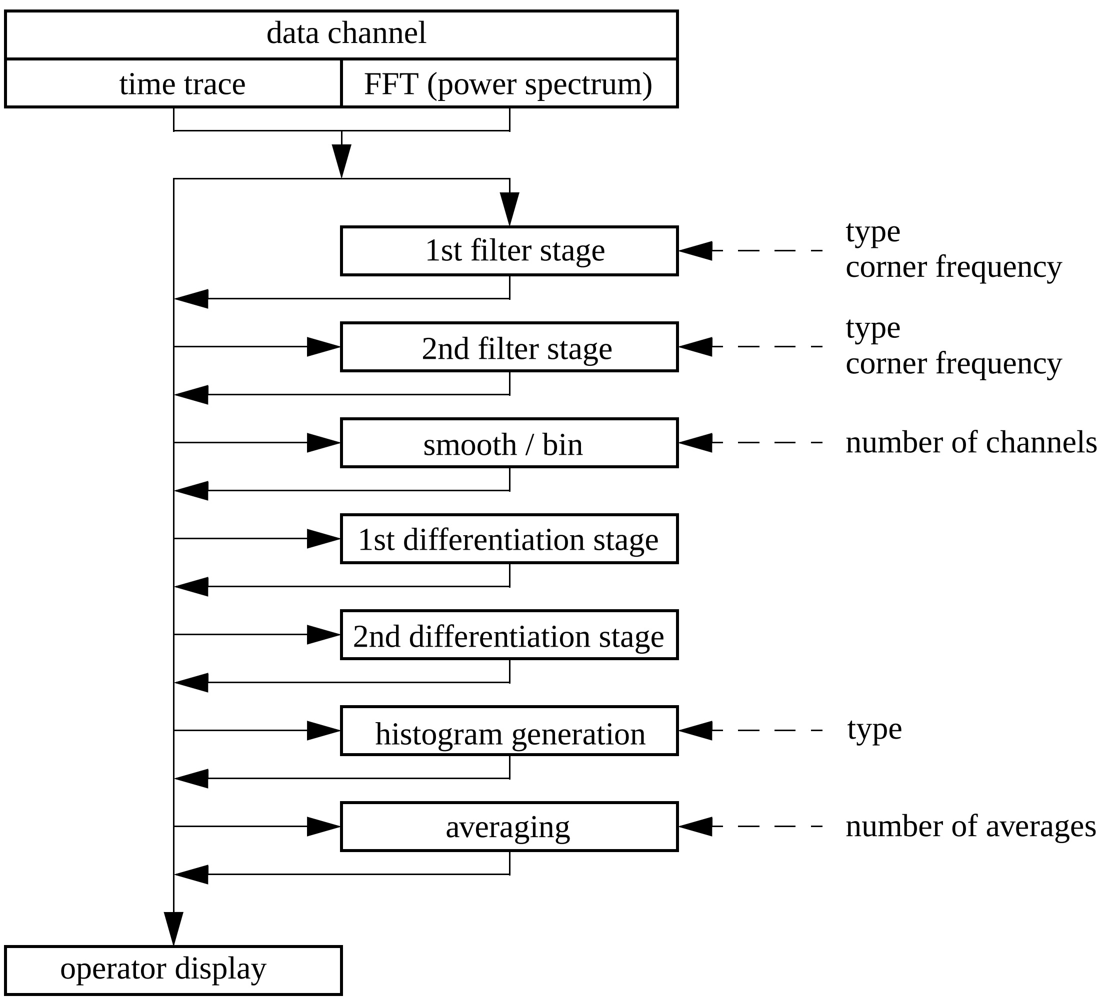


Figure 3: Data preprocessing for quick-look capability.

where the data can be sorted by pulse height or χ^2. When differentiating a power spectrum, the difference is taken in respect to old value of the frequency bin, rather than the neighboring frequency bin. The last stage is used for averaging spectra: both a moving average and or a fixed number of averages can be selected. A moving average will always display the average over the last few spectra, whereas the fixed number of averages will stop when the number of desired spectra is reached. For a histogram the spectra are simply added instead of averaged.

2.2.2.2 Statistical Functions

The following statistical functions can be calculated on all of the resulting spectra: the mean value, the standard deviation, the r.m.s. value and the value and the number of the channels with the highest and lowest values. These functions are evaluated either on the full spectrum or on a window within the spectrum set by window markers. To make it easier to look at long time series the highest and lowest values ever can be evaluated. Additionally, the user can select a channel marker and the value and the number of the corresponding channel will be displayed. The channel marker can be set automatically on the peak value lying within the displayed spectrum or within the window markers. To support harmonic analysis the multiples of the channel marker can be shown as well.

Function	Description
Mean value	mean value of the data in the displayed plot
Standard Deviation	standard deviation of the data in the displayed plot
r.m.s. value	r.m.s. value of the data in the displayed plot
Maximum	value and channel number of the channel with the highest value in the displayed plot
Minimum	value and channel number of the channel with the lowest value in the displayed plot
Highest ever	value and time of the highest value in a channel ever
Lowest ever	value and time of the lowest value in a channel ever
Marker values	displays position and value of the data indicated by the marker

2.2.2.3 Simple Triggering

A simple level trigger can be used to search for peaks in the data. It can work across all viewed channels, i.e. if a trigger is detected all displayed spectra will be frozen simultaneously. The trigger is typically placed in the center of the displayed spectra and can be either continuous or single shot. Additionally, the operator can stop the automatic updating of the plots at any time to take a deeper look at the displayed data.

2.2.2.4 Sophisticated Triggering

More complicated triggers are supported by using the trigger concept of section 2.3 in conjunction with a flag channel. (Flag channels are channels reporting the status of background triggers and are treated like normal data channels.)

One of the sophisticated triggers will be a fast-update pulse height analyzer. This will be important during commissioning to track down the “wee dwangies”. The pulse height analyzer will implement the following processing steps:

Analysis step	Description
band-pass filer	get rid of the unimportant part of the frequency domain
matched filter	~100 ms to 10 s; look for the wee dwangies
threshold detector	discriminate against non-matches
peak find	find time of occurrence and strength of the wee dwangies
write flag channel	write amplitude of found wee dwangies back as trigger channel data

The viewing tool can the trigger on the flag channel and either generated a histogram of the number of wee dwangies as function of their strength, or display the time series which let to the trigger event.

2.2.2.5 Storing Data

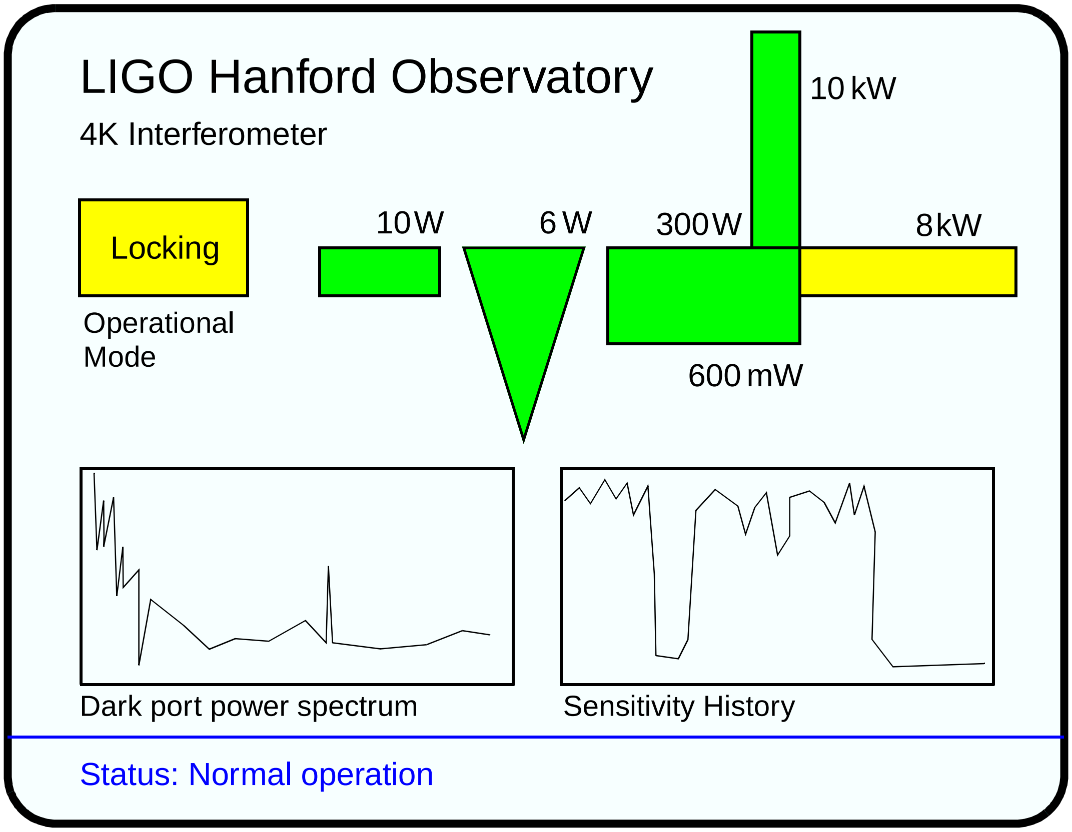
Graphical snapshots and data dumps of the displayed spectra can be stored to disk or send to the printer for documentation purposes or further analysis.

2.2.3 User Interface

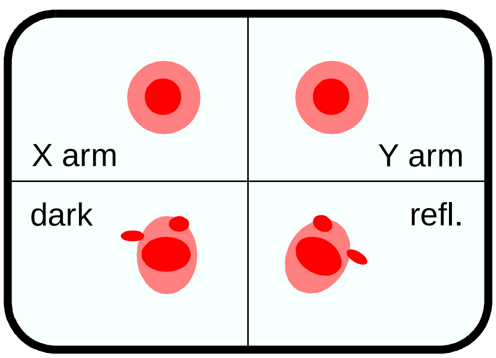
2.2.3.1 Permanent Monitors

The most important diagnostic information will be permanently displayed on separate screens (see Fig. 4). This permanent diagnostics screen consists of the following monitors: operational mode indicator, interferometer subsystem status, video camera images, power levels, alignment state, power spectrum of the signal at the antisymmetric port and a performance history.

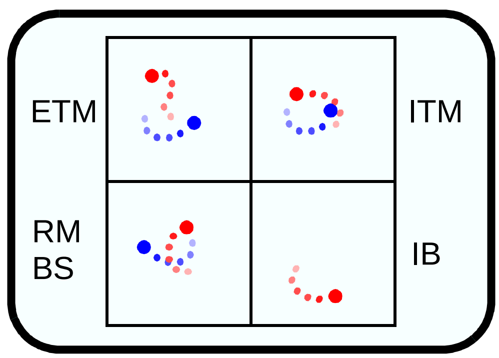
Monitor	Description
Operational mode	Large indicator; color code: green: detection mode yellow: locking, going up red: down blue: diagnostics mode
Subsystem status	pictorial representation of the laser, mode cleaner, recycling cavity and the arm cavities; color code: green: locked and running yellow: changing state red: down
Power levels	power level values at the output of the laser, at the output of the mode cleaner, in the recycling cavity, in both arm cavities and at the antisymmetric port
Antisymmetric port FFT	moving average of the power spectrum of the signal measured at the antisymmetric port (histogram updated every second, displayed in persistence mode)



(a)



(b)



(c)

Figure 4: Permanent diagnostics screens: a) overview, b) video and c) alignment. Not shown is the forth display showing the time histories of the length state and the values of the flag channels.

Monitor	Description
Performance History	Sensitivity history over the last 12 h (trace displayed in roll mode, update every minute)
Length state	The r.m.s. values of the length deviations over the last one second is plotted against time
Alignment state	XY display of the alignment state of all interferometer mirrors and the input beam direction (displayed in persistence mode)
Flag state	The values of some of the flag channels plotted against time
Beam Profiles	video images of the beam in reflection of both arm cavities, at the antisymmetric port and in reflection (color code for intensity)

2.2.3.2 Plots of Audio-Frequency Channels

A commercially available GUI library running on an X-windows workstation will be used for displaying real-time data. This displays are primarily available in the control room and in the experimental areas. Screen dumps can be printed to postscript printers and traces can be stored to disk. The plots which will be supported by the real-time viewing tool are summarized in Table 1.

Table 1: List of the supported plots for the real-time viewing tool.

Data	Type[1]	Description
Time traces	tY standard	shows channel values against time; continuously refreshed
	tY roll	shows channel values against time; roll mode
	tY multiple	shows values of up to 16 channels against time; continuously refreshed; color scheme
	tY roll multiple	shows channel values of up to 16 channels against time; roll mode
	tY minmax	shows channel values plus their minimum and maximum against time; continuously refreshed
	tY persistence	shows channel values against time; continuously refreshed; old time traces are faded away using a color scheme
	tY stack	shows channel values against time; continuously refreshed; old time traces are stacked towards the upper right
	XY standard	shows one channel value against another; continuously refreshed
	XY multiple	shows up to 16 channel pairs; continuously refreshed
	XY persistence	shows one channel value against another; continuously refreshed; old points are faded away using a color scheme
	XY multiple persistence	shows up to 16 channel pairs; continuously refreshed; old points are faded away using a color scheme
Power spectra	fY standard	shows a power spectrum; continuously refreshed
	fY multiple	shows up to 16 power spectra; continuously refreshed
	fY minmax	shows a power spectrum plus maximum and minimum envelopes; continuously refreshed

Table 1: List of the supported plots for the real-time viewing tool.

Data	Type[1]	Description
	fY persistence	shows power spectra; continuously refreshed; old spectra are faded away using a color scheme
	fY stack	shows power spectra; continuously refreshed; old spectra are stacked towards the upper right
	tf standard	power spectra are shown against time; a gray scale is used to indicate the height of each frequency
Histograms	hN standard	shows a histogram; continuously refreshed
	hN multiple	shows up to 16 histograms; continuously refreshed
	hN minmax	shows a histogram plus maximum and minimum values; continuously refreshed
	hN persistence	shows histograms; continuously refreshed; old histograms are faded away using a color scheme
	hN stack	shows histograms; continuously refreshed; old histograms are stacked towards the upper right
	th standard	histograms are shown against time; a gray scale is used to indicate the height of each histogram bin

1. t: time, X: channel 1, Y: channel 2, f: frequency, h: histogram bins and N: number

The viewing tool displays one plot per window. However, operators can open multiple windows of plots. Each plot is supported by the following display functions:

Function	Description
Acquisition mode	continuous, single shot, continuously triggered, triggered single shot
Cursor	single/pair of X, Y, X/Y, delta or sideband (with measurement)
Axes	linear, logarithmic
Axes label	automatic, user defined
Grid lines	none, automatic, user defined
Horizontal zoom	cursors, user defined
Horizontal binning	automatic, user defined
Screen dump	postscript to printer/file; ASCII traces to display/printer/file
Legend	automatic, user defined

Function	Description
Statistics	none, full plot, defined by cursors
Menu, taskbar, property sheets	select channels, filters, preprocessing, trigger levels and above display functions
Mouse	cross position: show plot values left-click: set cursors double-click: show property sheets right-click: show menu middle button: zoom, pan

Each viewing tool supports three different views which can consist themselves of multiple windows showing different channels. Because triggers are working across plots belonging to the same view, this feature allows up to three independent users to investigate the behavior of the detector. However, the total number of simultaneously open channels can not exceed the maximum allowed for a viewing tool.

2.2.3.3 Audio Headphones

The operator will be able to listen to one of the DAQ channels almost in real-time by using headphones. Due to the delay the data arrives at the global diagnostics system the audio channel will be delayed as well. Audio channels have the same filtering capabilities as the channels fed into the real-time viewing tool.

Property	Value	Unit
number of simultaneous audio channels	2	
maximum bandwidth of the audio signal	8	kHz
typical delay of the audio signal	0.1	sec

2.2.3.4 Digital Oscilloscopes and Video Cameras

A commercially available software package running on X-windows will be used to look at the scope data (e.g. National Instruments Labview). Screen shots and data from traces can be stored to disk for further analysis. But, no further integration with the rest of the global diagnostics system or on-line analysis capabilities are foreseen in the initial instantiation; the data are available for further analysis when desired.

Video images can be viewed and stored to the diagnostics archive from the control room. If supported by the hardware of the video system, taking pictures can be triggered using a simple level trigger on a audio-frequency channel. No further integration with the rest of the global diagnostics system nor the implementation of an on-line image processing system are foreseen in the initial system.

2.2.3.5 Supervisory Control Channels

The operator can look at and alter arbitrary supervisory channels using EPICS software.

2.3 Search Tool

The primary goal of the search tool is to constantly monitor the behavior of the detector and the physical environment in real-time and to automatically raise an alarm when something goes wrong or looks odd. The search tool has the ability to perform any kind of analysis on the data acquired by the DAQ system in real-time, check the results against established limits and generate an event when these limits are exceeded.

We roughly distinguish between performance monitors which search for abnormal behavior and broken channel indicators which should make sure that the individual channels are working at all. Performance monitors can be used to look at beam intensities, excess noise in the gravitational wave channel, saturations in servo control signals, enhanced seismic excitations, weather patterns and a lot more. All these monitors and indicators will use a common infrastructure which is organized as a trigger system. This means that each monitor will have a software condition associated with it which will be evaluated periodically (as fast as 16 times per second). When the condition is met, it is said that the trigger is on. Triggers by themselves do not report to the operator, but instead can be used to generate an event-trigger[1] which is a persistent data object carrying its own time stamp. The separation between triggers and event-triggers is done to enable logical operations between triggers. For instance, a coincidence trigger is nothing but the logical ‘and’ between other triggers; i.e. it is only on when all its input triggers are on at the same time. It is then easy to write monitors which only report — i.e. generate an event-trigger — when several conditions are met simultaneously.

Sometimes, it is useful to look at the original data which produced the event-trigger. The search tool will provide a means to reconstruct an event from its event-trigger and the original data. Another advantage of conceptually separating triggers and event-triggers is that one can now take the output of a trigger and write it back as an additional flag channel. This flag channel can then be used to conveniently record the time series of a performance monitor, rather than relying on its event based representation. A flag channel can also be used as an input channel by other diagnostics tools; e.g. the real-time viewing tool can implement complicated triggers this way.

2.3.1 Conceptual Overview and Basic Requirements

Events are defined as short snapshots of a subset of the interferometer data showing an unusual behavior. Every event has an event time (time of occurrence) associated with it and is classified by an event type. The **event-trigger** is an event removed from its data, only carrying the information of the event time, the event type and a reference to its associated data. The event can always be reconstructed from its event-trigger and the full data set. On the other hand a **trigger** or **veto** is always current; it is simply either on or off. It is not associated with a specific event; but rather is its transition from off-to-on used to generate the event (or event-trigger).

Triggers and vetos are calculated directly from the raw data or from other triggers and vetos (implementing logical operations such as AND and OR); whereas event-triggers are generated from triggers and vetos only. An overview is shown in Fig. 5. Event-triggers are automatically collected and put into event-trigger queues. Different purposes, such as user display and event logging, may require different queues. One of the event-trigger queues is dedicated for event

1. An event-trigger is in concept similar to an event, except that it has no data associated with it.

reconstruction; resulting events are put again into appropriate event queues. The output of a trigger can also be used as a flag which is written into a data channel at the rate the trigger is evaluated. This flag channel will typically be an on-off indicator for a subsystem or a yes-no indicator for a physical noise source.

Since the data are available in sets of 1/16 sec, triggers are also calculated at this rate. Additionally, a possibility for slow triggers exists which are calculated only every 1 sec. Each trigger has an active-time and a delay associated with it. With an active-time of length 0 an active trigger is valid only in the time interval where it is calculated; whereas for greater values the trigger stays active for consequent time intervals. A non-zero delay value will delay the active state of a trigger by the specified amount. Furthermore, triggers are organized in levels with

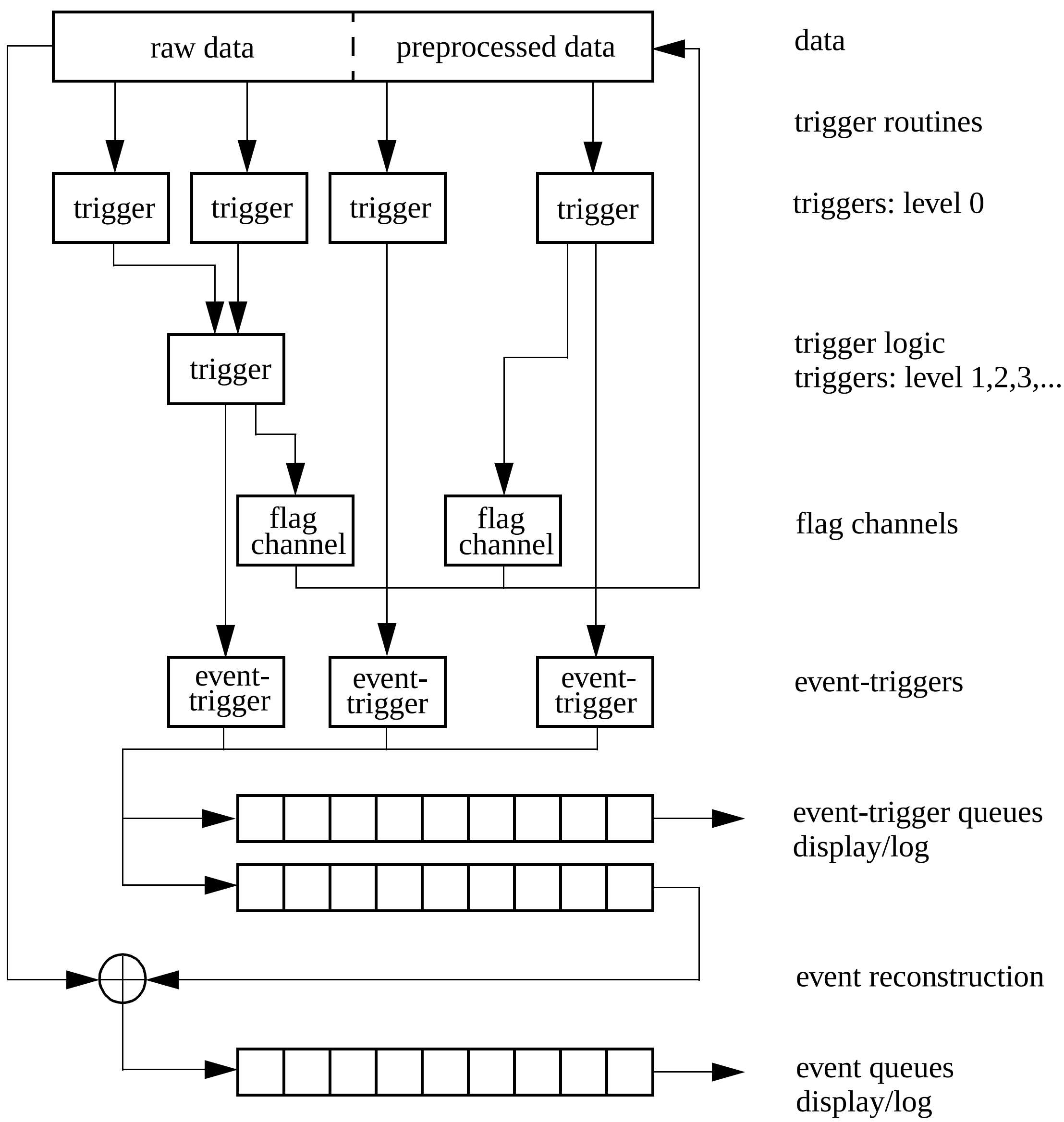


Figure 5: Overview and dependencies of triggers, event-triggers and events.

triggers at lower levels guaranteed to be calculated before triggers at higher levels within each time interval. This makes it possible to implement a complicated trigger logic. Some of the triggers have an event-trigger associated with them. Event-triggers are generated from these triggers whenever their state changes from inactive to active.

The requirements of the real-time status information diagnostics subsystem are summarized in Table 2.

Table 2: Requirements of the real-time status information diagnostics subsystem.

Description	Requirement	Unit
Data access	all audio channels and their FFT	
Maximum data delay	125[1]	msec
Minimum data history	2[2]	sec
Trigger rate	16 (normal) and 1 (slow)	Hz
Trigger interval	inverse of the trigger rate	sec
Maximum delay for flag channels (relative to input)	50	msec
Maximum number of triggers	2000[3]	
Evaluation time for all triggers	50[4]	msec
Maximum number of queues	10	
Maximum queue length	100	entries

1. Has to wait for FFT data.
2. i.e. a 3 sec ring buffer.
3. ~500 channels/interferometer; 4 triggers/channel (1 event, 1 logical, 1 broken and 1 spare).
4. The evaluation of all trigger has to be finished before the next trigger interval starts.

2.3.2 Triggers and Vetos

2.3.2.1 Common Properties

The main purpose of a trigger or veto is to implement a condition which is re-evaluated each trigger cycle. (The length of trigger cycle is determined by the trigger rate.) If the condition is met, the trigger (veto) becomes active (inactive) after taking into account several other trigger parameters which specify delay, active time and dead time. Triggers can change their state not faster than the trigger rate. To support finer timing resolution the trigger condition also returns a relative event time. This will allow a timing resolution restraint only by the signal, however, it will not allow to catch more than one event within one trigger interval. Each trigger/veto has the following common parameters associated with it:

Parameter	Description	Default
ID	Trigger identity number	must be unique
TYPE	The type of trigger; used for classification	0 (General)
LEVEL	Trigger level; triggers at lower levels are calculated first.	0

Parameter	Description	Default
EVENT_TIME	The exact time of the triggered event relative to the beginning of the trigger interval	0
EVENT_SIZE	A value which can be used to describe the amplitude or size of an event	1
EVALUATION	A flag which is used to determine for which interferometer states the trigger is evaluated	detection mode only
CONDITION	not a parameter, but rather a subroutine which returns True if data of the present trigger cycle meets the trigger condition	must be provided for each trigger
VETO	If True, the trigger state is normally active and becomes inactive when the trigger condition is met	False
STATE	The state of the trigger (active or inactive)	inactive (trigger) active (veto)
TRIGGER	True if the state has changed from inactive to active (trigger) or active to inactive (veto) in the present cycle	False
EVENT_TRIGGER	Specifies whether an event-trigger is generated when TRIGGER becomes True	True
FLAG_CHANNEL	Specifies the flag channel number when non-zero; zero means that trigger won’t be written into a flag channel	0
DELAY	The delay before STATE becomes active (inactive for a veto) after the trigger condition is met (in units of trigger cycles)	0
ACTIVE_TIME	The time the trigger state stays active (inactive for a veto) after becoming active (inactive)	0 (present cycle only)
DEAD_TIME	The minimum time the trigger can not become active again (inactive for a veto) after becoming active (inactive).	0
private	Triggers can have their private data block	none

2.3.2.2 Trigger Condition

The trigger condition is the core of a trigger. It is evaluated once per trigger interval and has access to *all* audio-frequency data channels of the current and the previous trigger interval. Typically the trigger condition will evaluate a statistical function on one of these channels, compare it against a trigger level and return an on-state when the statistical value exceeds the trigger level and an off-state otherwise. When finding an event, it will also return the precise time when it happened and the amplitude or size of the event. It is important to note that the state returned from the trigger condition doesn’t directly correspond to the trigger state, but rather is used to determine it taking into account trigger delay, active time and dead time. Since the trigger condition is only given access to data from the current and previous trigger interval, any trigger which needs a longer time history has to keep the relevant information stored in its private data block. Each trigger condition has its own set of parameters describing when the trigger condition is met and when not. These parameters are stored in the detector database and are read in from there whenever the trigger system is initialized.

2.3.2.3 Trigger Logic

The trigger logic is implemented by using predefined triggers which — instead of looking at the Interferometer data — use the output of other triggers as inputs. To guarantee that logical expressions are calculated after the data triggers, logical triggers must use a trigger levels greater than zero. Similarly, logical triggers which use other logical triggers as inputs have to have a higher trigger level than their input triggers. To make sure that logical expressions are calculated correctly when the triggers which serve as inputs are slightly out of synchronization, these triggers should generally have an active time equal or larger than one trigger cycle. All logical triggers have multiple inputs which can be used as inverted inputs when necessary. Logical triggers inherit their time and size (EVENT_TIME and EVENT_SIZE) from the last input channel, which by changing, causes their output transition. Predefined logical operations are:

TYPE	Description
AND	The output is active (inactive for a veto) if all triggers which serve as inputs are active (or inactive for an inverted input)
OR	The output is active (inactive for a veto) if at least one of triggers which serve as inputs are active (or inactive for an inverted input)
XOR	The output is active (inactive for a veto) if exactly one of triggers which serve as inputs are active (or inactive for an inverted input)
MIN	The output is active (inactive for a veto) if the number of active (or inactive for an inverted input) triggers which serve as inputs is larger or equal the specified minimum
COUNT	The output becomes active (inactive for a veto) after a certain number of active (or inactive for an inverted input) input triggers is registered.
RATE	The output becomes active (inactive for a veto) if during the past few cycles a certain number of active (or inactive for an inverted input) input triggers was registered.

2.3.3 Flag Channels

A flag channel is normal data channel which reports the status of a trigger. Its value is zero, when the trigger is off, and it is equal to the EVENT_SIZE, when the trigger is on[1]. Due to the time needed to calculate the triggers the data in a flag channel is valid one trigger interval after the original data was made available. Flag channels can be used as monitor channels watching over the health of the detector or its subsystems. They can also serve as input for the simple level triggers used by the data viewing tool or to start a diagnostics test. Flag channels must have data rates compatible with the data acquisition system.

2.3.4 Event-Triggers

An event-trigger is a dynamic object which is generated whenever a trigger which has the event-trigger flag set fires. Event-triggers are then put into queues which are used for archiving, operator

1. Bit-level triggers are possible by specifying EVENT_SIZE = 1; multiple bit-level flag channels may be packed into a word-level flag channel.

warnings and event reconstruction. After being removed from all of the queues the event-trigger is finally disposed. Event-triggers have the following parameters associated with it:

Parameter	Description	Default
EVENT_TIME	Time the event happened which is associated with the event-trigger	trigger time
TYPE	The type of the event associated with the event-trigger; used for classification	same as trigger
SUBTYPE	The subtype of the event associated with the event-trigger; used for classification	trigger ID
PRIORITY	Priority of the event associated with the event-trigger; used for filtering	1
LOG_QUEUE	Event-trigger will not be put into the archiving (storage) queue, if False.	True
DISPLAY_QUEUE	Event-trigger will not be put into the operator warning queue, if False.	True
EVENT_QUEUE	Event-trigger will not be put into the event reconstruction queue, if False.	True
DATA	List of data channels where the event was detected (maximum of 10 channels)	no data

The following priorities are supported:

Priority	Value	Description
LOW	0	Events with a low priority can usually be ignored
NORMAL	1	Events with a normal priority are important enough to report, but do not require any user intervention.
YELLOW	2	Events with a yellow priority are important and require an acknowledgment by the operator to be cleared.
RED	3	Events with a red priority are very important and require an immediate response by the operator.

2.3.5 Event-Trigger Queues and Event Queues

The main purpose of the event-trigger queues is to interface the synchronous trigger system with an asynchronous system such as the operator console. Event-triggers are copied into the appropriate queues immediately after they are generated. A queue entry is cleared from the display queue after the operator has acknowledge to have seen it; and it is cleared from the log queue after it is archived to disk. Each queue has the ability to filter event-triggers (or events) according to their priority, their type and their subtype. The following parameters are supported by a queue:

Parameter	Description	Default
PRIORITY_LOW	Lowest priority of an event-trigger or event which will be accepted by the queue	0 (log) 1 (event recon.) 2 (operator)
REJECT_TYPE	List of event-triggers or events types which will be rejected from the queue	none
REJECT_SUBTYPE	List of event-triggers or events subtypes which will be rejected from the queue	none

2.3.6 Event Reconstruction and Events

One of the event-trigger queues is dedicated for event reconstruction. Using the information provided by the event-trigger which of the channels were responsible for raising the trigger, the event data are fetched from the global data set and attached to the event[1]. The parameters for an event are identical to the ones of an event-trigger with the exception of the DATA field which is now a list of channel data. Reconstructed events are put into event queues for further processing.

2.3.7 Classes of Triggers and Event Triggers

2.3.7.1 Performance Monitors

Performance monitors are used to monitor the performance of the detector during detection mode. If none of the performance monitors indicates a problem, the operator can be pretty sure that the instrument is working fine and that good data are taken. Performance monitors can be divided into the following categories:

Monitor	Description
GW noise	Looks for excessive noise in the gravitational-wave band of the antisymmetric readout
Calibration	Checks the validity of the current calibration of the antisymmetric port readout
Beam intensity	Checks the power levels in the Interferometer, the mode cleaner and at the input
Laser source	Looks for higher than usual FM and AM noise on the laser light and spurious signals in the reference cavity and pre-mode cleaner control signals
Modulation	Checks modulation depth and modulation frequencies
Servo r.m.s.	Looks for larger than usual r.m.s. fluctuations in the control and error signals of the interferometer length, mode cleaner length and alignment servo loops
Oscillations	Looks for oscillations and gain peaking in the length and alignment servos
Narrow band features	Looks for suspension violin string resonances and test mass resonances at the antisymmetric port

1. This will require to access the on-line LDAS archive.

Monitor	Description
Actuator saturation	Looks for saturation in the control signals
Photodiode	Monitors the health of the RF photodiodes, watches temperature, bias voltage and increased dark current
Excitation	Monitors the excitation and calibration system to make sure that it is really off when it should be turned off (as during detection mode)
SUS Coil driver	Looks for spurious signals above 50 Hz in the SUS coil driver read-backs
SUS local sensor	Looks for spurious signals measured by the local SUS sensors
Optical lever	Looks for spurious alignment fluctuations measured by the optical levers
Earthquake	Looks for increased seismic activity measured by the seismometers, the tiltmeters and the accelerometers
Vibration	Looks for excessive vibrations of the beam tube and the vacuum tanks measured by the accelerometers
EM storm	Looks for spurious signals in the magnet field detectors and increased activity in the EMI sensors
Acoustic noise	Looks for acoustic activity above the usual background noise
Weather storm	Looks for extreme weather conditions

The triggering of the performance monitors has the following common properties:

Property	Value
Trigger rate	16 Hz for GW noise, laser source, SUS coil driver 1Hz for the rest
Priority	Normal for “5 - 20 σ” events Yellow for events beyond “20 σ” Red for a factor of 2 or more degradation of the GW sensitivity
Delay	0 sec
Dead time	1 min for normal priority events 15 min for yellow priority events 5 min for red priority events

The following tests will be used for monitoring the performance:

Test	Description	Category
Power	Calculates the power in a frequency band using the FFTs of the data channels; uses a higher limit for the trigger	GW noise Laser source SUS coil driver Vibration Acoustic noise Servo oscillations
Burst	Searches for (unknown) peaks in the FFTs of the data channels	GW noise Laser source SUS coil driver Servo oscillations
Peak	Differentiates the time trace and checks the resulting trace against a higher limit	GW noise Laser source SUS coil driver SUS local sensor Optical lever Earthquake Vibration
R.M.S.	Calculates the r.m.s. of the time trace; uses a higher limit for the trigger — lower limit for dead channel test	Servo r.m.s. SUS local sensors Optical lever Earthquake EM Storm
Limit	Checks the time trace against lower and higher limits	Beam intensity Modulation Photodiode Excitation Weather storm

2.3.7.2 Broken Channel Indicators

The purpose of a broken channel indicator is to monitor the activity on channels related to sensor readings and raise an alarm when the channel stops working, i.e., its activity falls below a lower limit. To avoid a large number of alarm messages when a channel is broken and it is decided to fix it later, the operator can reduce the alarm rate of each individual indicator to a very low value. Broken channel indicators have the following properties:

Property	Value
Trigger rate	1Hz
Trigger logic	rise event-trigger if trigger rate is greater than 100 in the last 10 min
Priority	Low for single channel event-triggers working at the low alarm rate Normal for single channel event-triggers Yellow if the event-trigger rate (normal priority only) of all broken channel indicators together is higher than 3 in the last 30 min
Delay	0 sec
Dead time	10 min for normal priority events 30 min for yellow priority events 12 h for channels with a low alarm rate
Test	limit on standard deviation

2.3.8 Evaluation Hierarchy

During different states of the interferometer some of the performance monitors — such as a servo control signal saturation alarm during locking — will not work as intended, nor it is useful that they are even evaluated. For other performance monitors — such as the ones associated with the physical environment — it might be interesting to watch them permanently. Do make it easier to switch triggers on and off during different states of the detector, they are organized into trigger banks. The evaluation hierarchy of triggers can then be controlled via their corresponding trigger bank. Conversely, each state of the detector has an corresponding list of trigger banks which are evaluated during this state.

2.3.9 List of Triggers and Event Triggers

See Appendix B.

2.4 Recording Tool

2.4.1 Conceptual Overview and Basic Requirements

The purpose of the diagnostics recording tool is to make summaries of the environmental history and to enable trend analysis over long periods of time (e.g., a full year). It also allows a monitor of the detector performance and makes possible a correlation with the physical environment. The data stored by the recording tool are typically strongly filtered and decimated (e.g., one barometric pressure read-out every minute), so that the total storage space is kept small. Storage of sequential data in the diagnostics archive is typically organized on a one-channel-one-file basis, so that summaries of a few channels can be kept on-line for the full lifetime of the detector. The recording tool is also responsible for writing the log file of the events found by the search tool.

The recording tool is always running. To keep the amount of data small, a filter and decimation stage reduces the data at the front-end, before it is transferred to a workstation. An overview of the data flow is shown in Fig. 6. Viewing and the analysis of data written to disk by the recording tool is done off-line and relies on commercially available software packages.

We require that the data written by the recording tool over a full year can be kept on disk for fast retrieval. The data rate of the recording tool written to the diagnostics history archive shall not exceed 1 kB/sec over extended periods of time. However, it can be higher in shorter time intervals. The data rate requirements are summarized below.

Description	Applicable time period	Maximum data rate
Burst data rate	1 minute	1 MB/sec
Short period data rate	1 hour	100 kB/sec
Long period data rate	1 month or longer	1 kB/sec

2.4.2 Data Flow

An overview of the data flow is shown in Fig. 6.

2.4.2.1 Data Acquisition

The recording tool has access to all data available to the GDS, i.e., channel data including FFT and flag channels and event queues. RF signals and video images will be stored using a commercially available software package.

2.4.2.2 Data Preselection and Data Transfer

The selection of data read by the recording tool is performed at the front-end with a channel selector and an event filter. The channel selector simply determines which channels are read into the filter and decimation stage and which are not. The filter and decimation stage is used to reduce the data rate of channel data from the rate determined by the DAQ system to the one specified by the recording tool. The filter and decimation stage can be configured for each channel separately. After the data are passed through the data conditioner they are transferred to the diagnostics workstation using front-end and back-end buffers. Events from the event queues pass through a simple event filter before they are transferred unchanged to the diagnostics workstation.

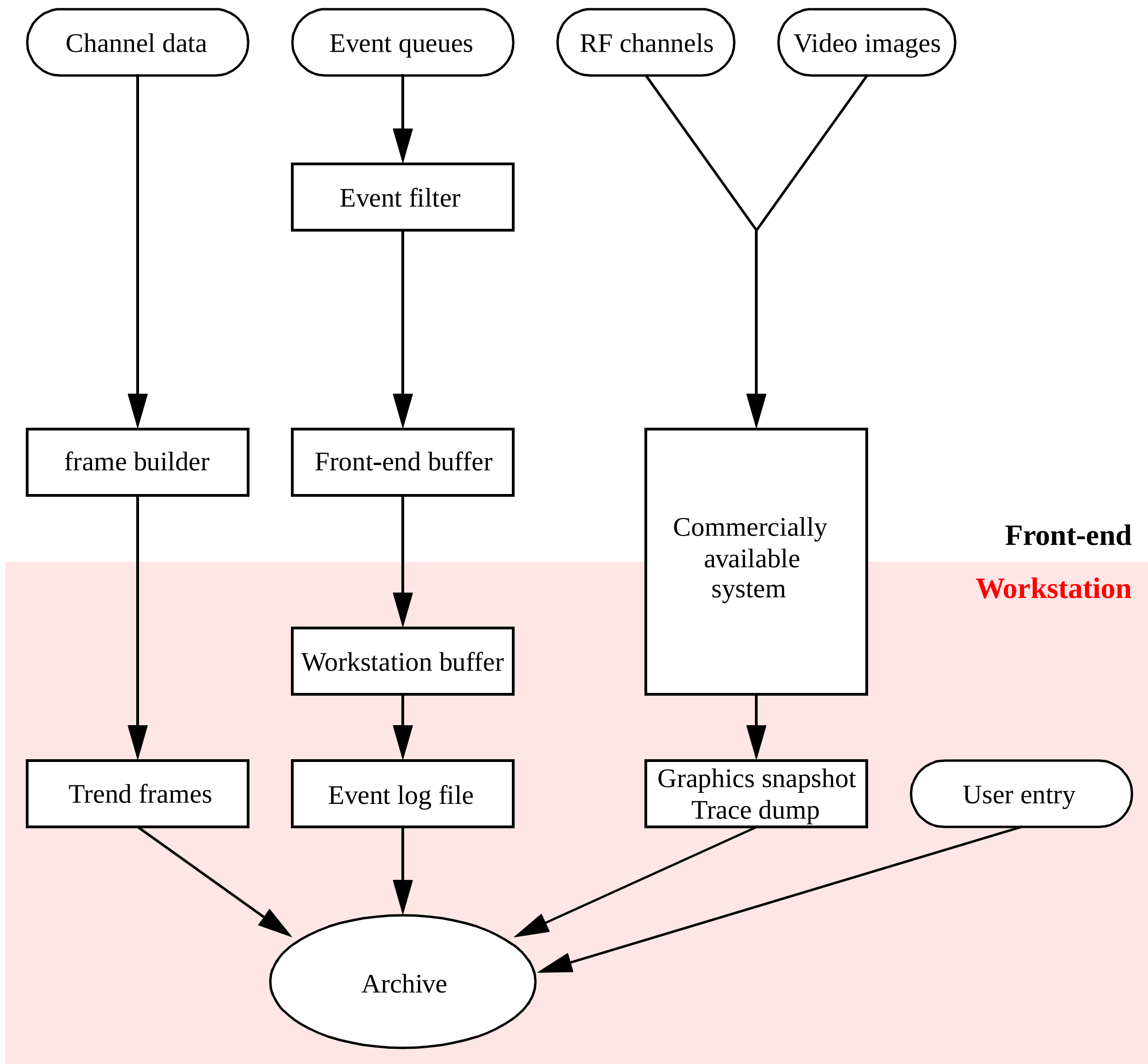


Figure 6: Data flow for diagnostics history recording.

2.4.2.3 Data Conditioning

Data conditioning is possible for channel data. If a data conditioner is used its output corresponds to a new channel; this way more than one data conditioner can be setup for a given data channel. Typical conditioning routines performed are conversion of the numerical ADC values into quantities with physical units, finding the maximum excursion within the recording interval or to apply simple calibration corrections. (The recording interval is the time between two consecutive writes of the recording tool for a given channel.) The classes of conditioning routines are as follows:

Conditioning routine	Description
Conversion	Convert numerical ADC values into physical units
Calibration	Simple calibration corrections
Maximum	Looks for the maximum excursion within a recording interval
Minimum	Looks for the maximum excursion within a recording interval
Mean	Calculates the mean value over a recording interval
Standard Deviation	Calculates the standard deviation over a recording interval
r.m.s.	Calculates the r.m.s. value over a recording interval within a certain frequency band
Amplitude	Calculates the amplitude of a given frequency over a recording interval
Power peak	Looks for the highest peak within a band of the power spectrum; can write either its amplitude or its frequency

2.4.2.4 Data Storage

Data storage is organized on one-channel-one-file basis for sequential channel data, on a one-set-one-file basis for test data, on a one-snapshot-one-file basis for oscilloscope data and for video images and on a add-to-a-log-file basis for events. Typically, files do not contain data for periods longer than one day. Each channel file has a time stamp and a data rate field at the beginning, whereas the event log file relies on the time stamps of the events and the snapshot files rely on the file creation time stamp.

The file format of data written by the recording tool is a lightweight data format for channel data and event logs. Time traces of oscilloscopes, video images and oscilloscope images are using the data format supported by their software package.

The data written by the recording tool over a full year shall be kept on disk for fast retrieval.

2.4.3 Off-Line Viewing and Analysis Tools

No GDS specific on-line viewing of data passed through the recording tool is foreseen. Instead, one has to read the data from disk into commercially available plotting and analysis packages such as Matlab or Mathematica, or by using the LDAS viewer.

2.4.4 List of History Channels

See Appendix A.2.

3 DIAGNOSTICS TESTS

The diagnostic system serves to characterize the performance of the interferometer by establishing relationships between operating parameters, subsystem performance, and environmental inputs. It does this through passive observation, parametric studies, and through active stimulus-response testing. To put the task in context, it will enable the kind of research on the LIGO interferometers which dominates the effort on the laboratory interferometers (40m and PNI), with the expectation of allowing a more controlled approach with additional information to aid in the interpretation of the instrument.

The top-level requirements are to acquire all data which have a significant effect on the interferometer performance, long or short term; and to provide tools to allow that data to be exploited.

Accordingly, the goal of the proposed diagnostics test subsystem is to assist the operator in characterizing the dynamical behavior of the detector and in diagnosing problems which can not be found with the real-time status information subsystem alone. In most cases a diagnostics test will involve an active stimulus and an analysis algorithm in order to be able to obtain and extract the desired information. The proposed system will support:

- A set of generic test templates which can be readily customized. A good example is a swept sine test which can be specified with a set of parameters describing the stimulus channel, the measured channels, the start and stop frequency, etc.
- A way to apply a stimulus to the test inputs of the instrument.
- A way to obtain and archive the information in the tested channels.
- A set of analysis algorithms and plotting routines to extract and display the test results.
- A diagnostics archive which will keep track of test results and which will allow one to look at historical trends of detector performance.
- A way to fully automate diagnostics tests which have proven to be useful.

3.1 CONCEPTUAL OVERVIEW

Diagnostic tests exist in a wide variety. Any attempt to list them in a complete manner beforehand will undoubtedly fail. It is therefore extremely important that the diagnostics test subsystem be flexible and expandable. These goals are achieved by clearly separating the different tasks of a diagnostics test: the excitation, the data gathering and the analysis. Each diagnostics procedure can then specify how to apply the excitation, where to obtain the data and what do with the data in a way best suited for this test. An excitation can be applied either through predefined parameters or directly by the operator. Data can be acquired through the DAQ storage system, through the recording tool of the real-time status information subsystem and through the front-end processor of the diagnostics test tool itself[1]. Analysis algorithms can be evaluated on-line or off-line, either using a high-level analysis and graphics package (such as Mathematica or Matlab) or using a high-level programming language (such as C or C++). An overview of the data flow in the diagnostics test subsystem is shown in Fig. 7.

1. The data path through the diagnostics test tool is mainly intended for scheduled tests that are of fixed duration and involve only a small number of data channels.

Diagnostic tests can be categorized whether they are best done on-line or off-line, whether they are invasive or not and whether they are restricted to a single subsystem or not.

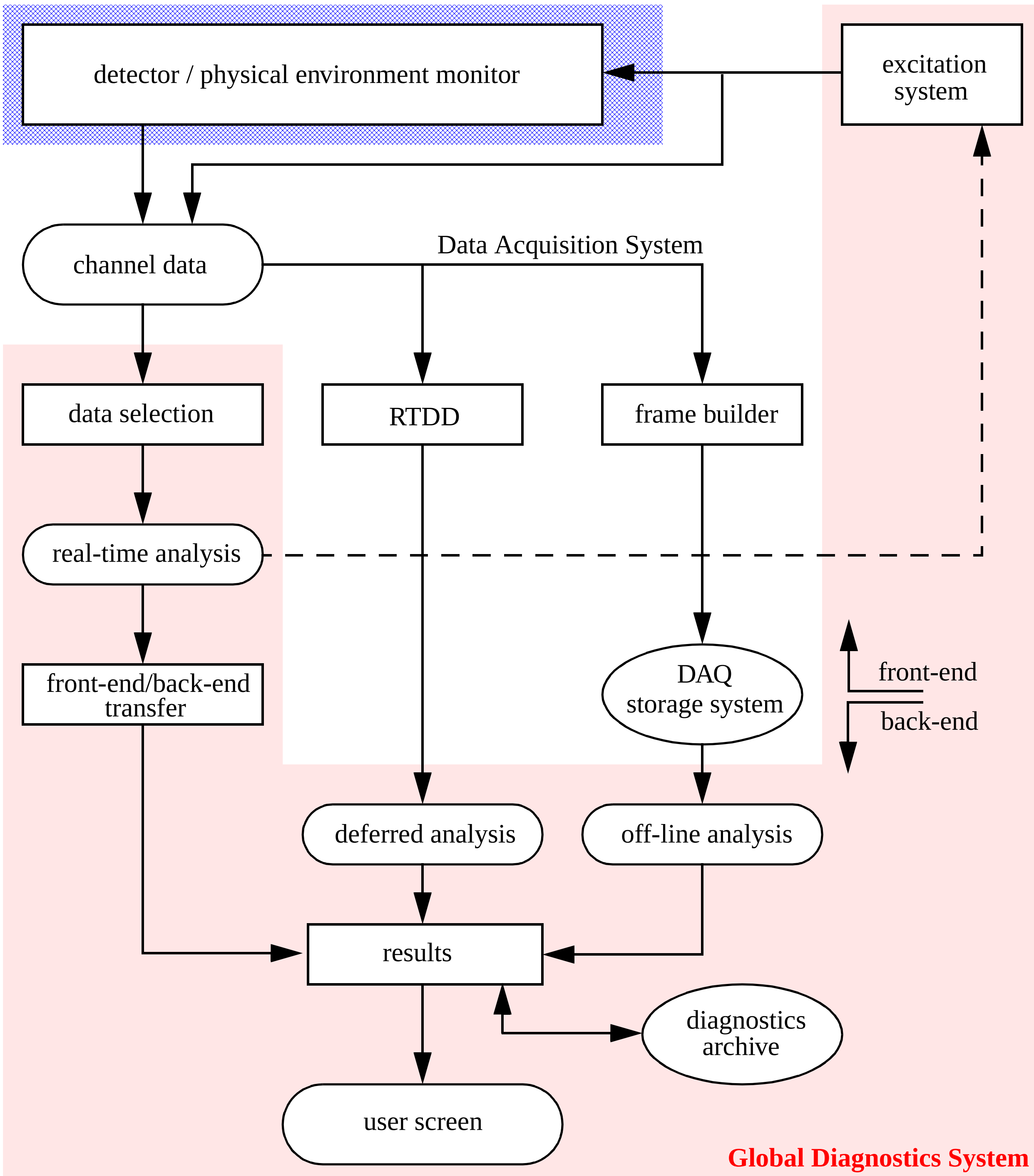


Figure 7: Data flow for diagnostics tests.

3.1.1 On-Line and Off-Line Tests

On-line tests are those which yield results in quasi-real-time, or are invasive (see below). A ring-down measurement of the mirror losses is an example, as is a transfer function from laser frequency modulation to an interferometer output. Clearly, some ‘deferred’ analysis of on-line tests can be performed after the fact, muddying the distinction. A limited number of tests require ‘closed-loop’ control of the test; and example is a stimulus whose level is continuously adjusted to maintain an output of the interferometer below saturation but at a level which allow a good signal-to-noise. The bandwidth of this control is limited to the refresh rate of the reflected memory system and the EPICS bandwidth.

Off-line tests involve the analysis of stored data (by the recording tool to disk, see section 2.4, or as part of the normal data stream from the short- or long-term storage). Study of the correlations of barometric pressure with alignment signals is an example of an off-line test.

It is envisioned that the initial development of new test algorithms is done off-line or deferred using a commercial high-level analysis and graphics package. After establishing a new diagnostics test the analysis algorithm can migrate from off-line to on-line. For on-line analysis we distinguish between real-time analysis which is performed on a back-end workstation and pre-analysis which is performed on the front-end; both tasks are typically written in C or C++.

3.1.2 Invasive and Non-Invasive Tests

Because of the impact on the interferometer availability, we carry a distinction between invasive and non-invasive tests. Non-invasive tests degrade in no way the performance of the interferometer. An example would be to monitor both the accelerometers at the support beams of the seismic isolation system and the motion of the optical components as inferred from interferometer signals and to form the ratio of powers to determine if the isolation system is performing normally.

Invasive tests more or less degrade or make useless the output of the interferometer for GW searches. Clear examples are cavity storage time measurements which involve stopping the beam and looking at the decay of light in the interferometer, or high-level broad-band excitation of the ground to measure the transfer function of the seismic isolation system. It is also possible to perform tests which are only minimally invasive; introducing a single-tone modulation (e.g., calibration signal) or a very low-level pseudo-random noise which could then be identified in the output are examples.

3.1.3 Subsystem and Global Diagnostic Tests

Subsystem diagnostic tests are designed to determine the performance or otherwise characterize a given subsystem or subsystem component. A measurement of the frequency noise of the laser is an example. Many of these tests rely upon other subsystems to perform the test. Global diagnostic tests are those which are intrinsically cross-subsystem; a measurement of the influence of frequency noise on the interferometer output is an example. This document concentrates on establishing the capability necessary for performing both kinds of tests, without attempting to document completely the subsystem tests (especially those completely internal to a subsystem).

3.1.4 Diagnostics Test Templates and Automatization

Diagnostics tests can be classified (see next section). For instance, the two-tone intermodulation test class can be generally useful for determining non-linearities in the interferometer, and specifically (as an example) in the length sensing and the alignment sensing. What generally distinguishes these tests are where and how the stimulus is applied and what are the measured channels; usually, they share the type of stimulus and the analysis algorithms. It is then only natural to define diagnostics test templates which allow the operator to specify a set of parameters, rather than writing a new piece of software for each new test.

When performing a test for the first time, one typically wants to control it step-by-step to make sure that everything works as intended. Also, it might be initially useful to look at the raw data directly, rather than relying on predefined algorithms which only look at one aspect of the data. For example, it is quite obvious from the raw data when a swept sine test has saturation problems; analyzing it without being aware of this problem probably returns wrong results. However, after thorough testing one will be able to fully automate the test, i.e., one starts the test and only has to wait for the result to show up.

3.2 Classes of Diagnostics Test

There are many types of tests and measurements that will be done in the LIGO to establish that the detector noise is minimized and the interactions of the facilities and the environment with the detector are understood. Typical types of tests are to:

i) optimize operating points of the interferometer adjustable parameters,
ii) determine the transfer functions of one output to another,
iii) determine the transfer function of a perturbation to the interferometer output,
iv) determine the cross-power spectrum of one output against another,
v) determine the sensitivity to a particular parameter by stimulation,
vi) characterize servo systems,
vii) measure the noise from a particular mechanism,
viii) calibrate the sensitivity of the system,
ix) measure the dynamic range of an output,
x) determine the linear range for a particular output,
xi) determine the intermodulation products due to offsets, saturations and non-linearities.

The test signals come from points in the interferometer and its various sub-systems, the environmental monitoring system and the facility monitoring system. The diagnostic uses of the intercomparison of the full and half length interferometers at the Hanford site and the correlation of the information from the Livingston and Hanford sites are not yet considered.

Some of the tests extend a time-honored technique used in the development of all the gravitational wave detector prototypes to diagnose the sources of the limiting noise to the LIGO. The method has four elements:

i) The system is stimulated by either periodic or random excitation at a control point that simulates the effect of a particular noise source. The level of the excitation is chosen to be in the linear range of the system but sufficiently large to override the naturally occurring fluctuations at the gravitational wave output. Example: frequency modulation of the laser.

ii) The transfer function is measured of the excitation to the point in the system most sensitive to the excitation and minimally sensitive to other sources of noise. Example: the light reflected from the interferometer appropriately demodulated.

iii) A measurement is made of the transfer function of the excitation to the gravitational wave output of the system. Example: transfer function of frequency modulation to the GW output.

iv) With the stimulation off, a measurement is made of the noise at the most sensitive point (e.g., reflected light) and the cross-correlation (cross power spectrum) to the gravitational wave output to determine the contribution of the specific noise source to the noise budget.

The method assumes a separation of the individual noise sources can actually be made. Often this is the case. Another method we will want to use is to sample several different points where noise enters the system at the same time and to use the outputs at various points in the system to solve simultaneously for a superposition of several noise sources at the gravitational wave output. It is then not necessary to make the approximation of a “diagonal” representation of the noise sources and it should be possible from the matrix of cross power spectra or cross correlation functions to solve for the contribution of each noise source and to develop the error and covariance matrices. A principal value decomposition of the multiple-input—multiple-output data will be useful.

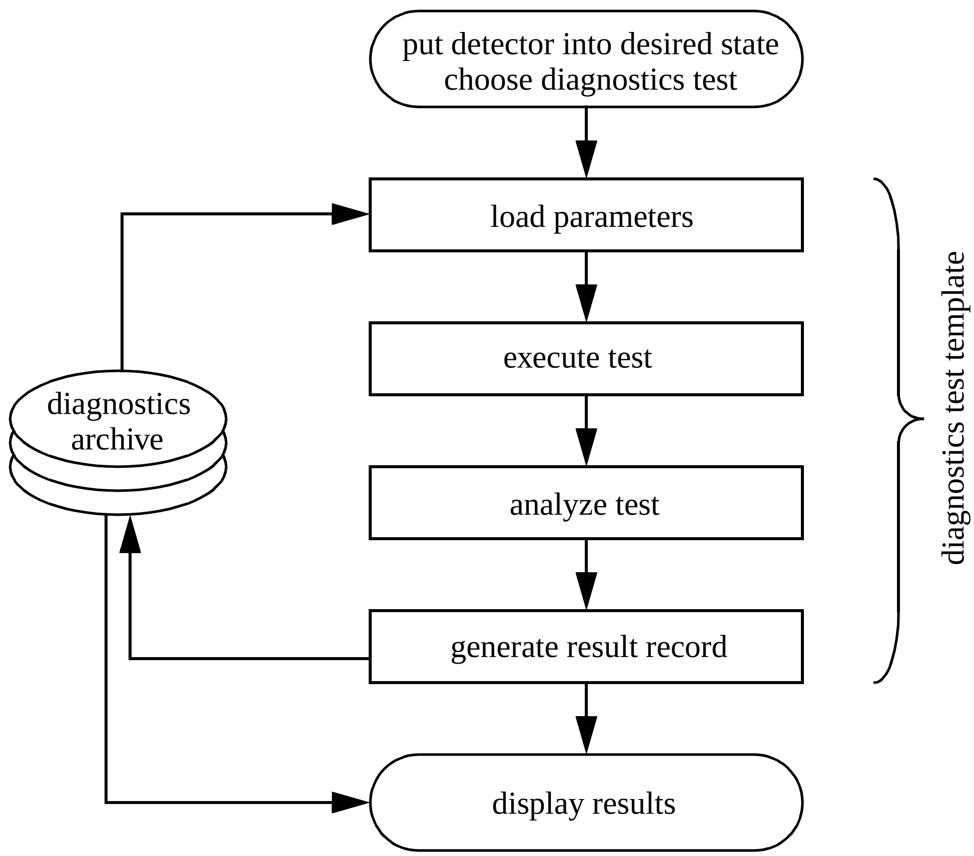


Figure 8: The function of diagnostics test templates.

Another class of tests are those used in optimizing the operating parameters of the interferometer. These tests search for stationary points in the range of an operating parameter. Often the stationary points are the places where the noise is minimized or the system sensitivity to drifts or gain changes is smallest. The stationary points are found by actively varying the parameter through periodic dithering coupled to a systematic variation in the parameter to find the extremum. The technique requires a detection system synchronous with the dither - a suppressed carrier modulation system. Many of these tests are focussed on one subsystem and can best be described in the context of the diagnostic tests of a given subsystem.

Standard techniques are described below which are referred to in the listing of tests in Appendix C.

3.2.1 Common Properties

Each diagnostics test has a template and a list of parameters associated with it. The list of parameters is stored in a separate file. This parameter file lists the used template and the values for each individual parameter. A test template then serves as a generic description for each test class (see Fig. 8); it consists of:

i) a description of the necessary parameters,
ii) a piece of code which executes the test,
iii) algorithms to analyze the test and obtain the test results and
iv) a definition of the result record.

Since the code, the algorithms and the structure of the result record is common to all tests belonging to the same class, it is straight forward to create a new test from an existing test template by making a new parameter file. The result record is divided into the following sections:

Result record section	Description
Header	header information describing the test (name, class, date, time, etc.)
Detector state vector	describes the state of the detector at the time the test was taken
Test parameters	lists all the parameters used for the test
Raw data	section for storing the measured raw data
Analyzed Data	section to store the test results; divided into:
intermediate steps	subsection to store results from intermediate steps
final result	subsection to store the final results

Depending on the test some sections might stay empty. The following parameters are used to define what is written to the result record:

Parameter	Description
StoreState	yes (default) or no; for storing the detector state vector
StoreParameters	yes (default) or no; for saving the test parameters with the result
StoreRawData	yes or no (default); for storing the measured raw data

Parameter	Description
StoreIntermediate	yes, no, Automatic (default; erases results of intermediate steps after receiving the final result)
StoreFinal	yes (default) or no; for storing the final result

Several parameters which are common to all tests govern its execution. These parameters can be both specified in the parameter file or at execution time where they overwrite the values given in the parameter file:

Parameter	Description
WaitForStart	yes or no (default); if yes, the test waits with its execution until it gets a signal from the top level controls[1] that the interferometer is in the proper state
WaitAtEachStep	yes or no (default); if yes, the test waits with its execution at each step until it gets a signal from the top level controls that the interferometer is in the proper state
SignalEndOfStep	yes or no (default); if yes, sends a signal to the top level controls whenever a step of the test is finished
SignalEnd	yes or no (default); if yes, sends a signal to the top level controls when the test is finished
AllowCancel	yes (default) or no; determines whether the test can be canceled by the operator during execution
ExecuteInForeground	yes (default) or no; diagnostics test are separated into foreground and background tests; only one foreground test can be executed at any given time, whereas multiple background tests can run simultaneously
NoStimulus	yes or no (default); if yes, overwrites the standard stimulus behavior of the test (only useful if the stimulus is provided by other means)
NoAnalysis	yes or no (default); if yes, disables the default analysis method
NoResult	yes or no (default); if yes, no result record is stored
DoNotChangeState	yes (default) or no; if yes, the test will make sure that the detector state is the same after the test than before; most tests want this to be yes, however, when optimizing the performance by varying a parameter one might want to keep the new optimized detector state
DisplayIntermediate	yes (default) or no; determines whether results of intermediate steps are display during the execution of the test
DisplayFinal	yes (default) or no; determines whether the final result is automatically displayed when the test is finished

1. The top level controls are meant to be the main control screen of the detector which is used to issue ‘on-off-type’ commands to individual subsystems.

3.2.2 Sinusoidal Stimulus/Power Spectrum Readout

Sine waves at selected frequencies and amplitudes are applied to one or more test points. The power spectra of output signals are monitored; by observing the relationship of the peak height to the noise background and knowing the size of the injected signals, one can immediately determine the SNR with respect to the injection point without separately measuring all the gain factors or interaction strengths (if the coupling mechanism is not well characterized or stable). Averaging of successive measurements is necessary to refine the power spectrum accuracy. Measurements of harmonics, two-tone intermodulation, and sideband amplitudes are also common.

3.2.2.1 (Multiple) Sine Response

A sine response test applies a single sine wave stimulus to a test point and analyzes the response of the system at one or more points. A multiple sine response test applies multiple stimuli of different frequency at different test inputs simultaneously. Typically, one will also look at a number of measurement channels to test how the different stimuli are showing up at different points in the detector. For example, a measurement of the alignment sensing matrix wiggles all mirrors at different frequencies and analyzes the wavefront sensor error signals for each frequency component to determine all the alignment sensitivity matrix elements at once. The parameters for the (multiple) sine response test are as follows:

Parameter (Unit)	**Description**
TestClass	SineResponse
TestName	unique identifier
TestDuration (s)	time of measurement; default is longest of 100/StimulusFrequencyM
StimulusChannelM	channel name or number of the Mth test input; M = 1, 2, 3,... 20
StimulusFrequencyM (Hz)	frequency of the Mth stimulus
StimulusAmplitudeM (V)	amplitude of the Mth stimulus
StimulusOffsetM (V)	offset of the Mth stimulus
StimulusWait (s)	time which is waited after applying the stimulus until the test starts default is 10/StimulusFrequency
MeasurementChannelN	name or number of the Nth measurement channel; N = 1, 2, 3,... 20
TestAnalysis	analyzes the measurement channels and calculates both the amplitude and the phase of the signal at each of the applied frequencies
TestResults	returns the power spectra and/or the signal amplitude/phase matrix

3.2.2.2 Harmonic Distortion

A harmonic distortion test is similar to a sine response with the difference that one not only looks at the fundamental frequency of the stimulus, but also at its harmonics. This test is both useful to look for non-linearities and saturation effects. The parameters for the harmonic distortion test are as follows:

Parameter (Unit)	Description
TestClass	HarmonicDistortion
TestName	unique identifier
TestDuration (s)	time of measurement; default is 100/StimulusFrequency
StimulusChannel	channel name or number of the test input
StimulusFrequency (Hz)	frequency of the stimulus
StimulusAmplitude (V)	amplitude of the stimulus
StimulusWait (s)	time which is waited after applying the stimulus until the test starts default is 10/StimulusFrequency
MeasurementChannelN	name or number of the Nth measurement channel; N = 1, 2, 3,... 20
TestAnalysis	analyzes the measurement channels and calculates the strength of each harmonic up to the Nyquist frequency of the data
TestResults	returns the power spectrum and/or the amplitude of the harmonics

3.2.2.3 Two-Tone Intermodulation

A two-tone intermodulation test is best suited to investigate non-linearities in the system. Two stimulus frequencies are applied simultaneously (most of the time at the same test input) and the measured channels are analyzed for both the sum and the difference of the two frequencies. Since the LIGO sensitivity has a strong frequency dependence (steep fall towards lower frequencies), non-linearities can easily up-convert a significant amount of low frequency noise into the gravitational-wave band. To test the strength of this up-conversion one would like to apply two stimuli — one in the gravitational wave band and one at the seismic peak — and look for 'sidebands' on the higher frequency signal. The parameters for the two-tone intermodulation test are as follows:

Parameter (Unit)	Description
TestClass	TwoToneIntermodulation
TestName	unique identifier
TestDuration (s)	time of measurement; default is 100/StimulusFrequency
StimulusChannel1	channel name or number of the first test input
StimulusFrequency1 (Hz)	frequency of the first stimulus
StimulusAmplitude1 (V)	amplitude of the first stimulus
StimulusChannel2	channel name or number of the second test input; default is StimulusChannel1
StimulusFrequency2 (Hz)	frequency of the second stimulus
StimulusAmplitude2 (V)	amplitude of the second stimulus
StimulusWait (s)	time which is waited after applying the stimulus until the test starts default is the longer of 10/StimulusFrequencyM

Parameter (Unit)	Description
MeasurementChannelN	name or number of the Nth measurement channel; N = 1, 2, 3,... 20
TestAnalysis	analyzes the measurement channels and calculates the amplitude of the signals at the sum and difference of the applied frequencies
TestResults	returns the power spectrum and/or the signal amplitude(s)

3.2.3 Pseudorandom Stimulus/Power Spectrum Readout

This is functionally similar, but for nonlinear interactions[1] a sine wave test tends to underestimate the coupling. Using a noise-like test signal is often necessary to get the true interaction strength (i.e. that which applies for the “real” noise). A Gaussian random probability distribution is best for most purposes. Ability to bandlimit the random noise sharply (e.g. a “box” bandpass) will help to get around a serious problem: to get a measurable interaction, broadband excitation may introduce so much additional signal power that the system goes *really* nonlinear. Analyzers often can synchronize the pseudorandom sequence to the sampling and frame rates; this can improve the measurement dynamic range and variance by allowing coherent (vector) averaging of successive FFT measurements.

Another advantage of the pseudorandom excitation is the possibility of it allowing low-level stimulus/response measurements without disturbing on-line operation significantly. Its disadvantage is generally long measurement times.

3.2.3.1 Pseudorandom Response

The parameters for a pseudorandom response test are as follows:

Parameter (Unit)	Description
TestClass	PseudorandomResponse
TestName	unique identifier
TestDuration (s)	time of measurement
StimulusChannel	channel name or number of the test input
StimulusSource	Random (default)
StimulusRMS (V)	r.m.s. value of the stimulus
StimulusFilter	optional filter which is applied to the noise source before being applied to the test input; default none
StimulusWait (s)	time which is waited after applying the stimulus until the test starts; default is 0.
MeasurementChannelN	name or number of the Nth measurement channel; N = 1, 2, 3,... 20
MeasurementAverages	number of averages; default is 10.

1. For example, coupling between mirror alignment and cavity length (quadratic) or amplifier saturation (limiting).

Parameter (Unit)	Description
TestAnalysis	calculates the power spectra and averages over consecutive measurements
AnalysisWindow	Hanning, Flat-top or Uniform
TestResults	returns the averaged power spectra after each step

3.2.4 Pseudorandom Stimulus/Cross Spectrum Frequency Response

This kind of test is principally meaningful for linear transfer functions. Stimulating a test point with a pseudorandom noise voltage, one collects two or more response waveforms and does a cross power spectrum for each pair. One of the response waveforms may be the stimulus itself, but frequency responses between any two affected points in the system may be needed. To insure that the cross power spectra measured are attributed to the same stimulus (and not naturally occurring noise on one input, for example) the quality of this kind of measurement needs to be monitored by averaging successive measurements and computing the coherence function as a measure of how much power at each displayed frequency actually came from the stimulus. The magnitude and phase of the transfer function are read-out on a Bode plot or similar. One will combine (multiply/divide) such transfer functions with power spectra from various sources, including ones with different numbers of points or nonuniform point spacing. Math operations on transfer functions and power spectra are also commonly used (like “divide by frequency squared” to turn force into displacement).

3.2.4.1 Pseudorandom Cross Spectrum

The parameters for a pseudorandom cross spectrum test are as follows:

Parameter (Unit)	Description
TestClass	PseudorandomCrossSpectrum
TestName	unique identifier
TestDuration (s)	time of measurement
StimulusChannel	channel name or number of the test input
StimulusSource	Random (default)
StimulusRMS (V)	r.m.s. value of the stimulus
StimulusFilter	optional filter which is applied to the noise source before being applied to the test input; default none
StimulusWait (s)	time which is waited after applying the stimulus until the test starts; default is 0.
MeasurementChannelAN	name or number of the Nth measurement A channel; N = 1, 2, 3,... 20
MeasurementChannelBN	name or number of the Nth measurement B channel; N = 1, 2, 3,... 20
MeasurementAverages	number of averages; default is 10.
TestAnalysis	calculates the cross spectra (B/A) and averages over consecutive measurements

Parameter (Unit)	Description
AnalysisWindow	Hanning, Flat-top or Uniform
TestResults	returns the averaged power spectra after each step

3.2.5 Swept Sine Frequency Response

This method for measuring linear transfer functions just applies sine waves at successive frequencies[1] and computes the relative magnitude and phase of the response waveform. At each frequency the measurement may be repeated to reduce the effect of noise; one may also compute the coherence function to determine the quality of the result at each point, as above. It is obviously more time-intensive than FFT methods, but because the stimulus amplitude can be varied at each frequency, it can achieve much higher dynamic range (one can measure a transfer function that varies by 120dB over the selected frequency range desired, virtually impossible with FFT-based methods). Also, since only a narrow band is excited at any given time, this measurement usually adds negligible signal power to the system under test so it is much less perturbative. In addition to the cross spectrum capabilities mentioned above, swept signal analyzers generally include handy features like:

- active interpolation of additional frequency points when adjacent measurements differ markedly, so narrow features and transitions get accurately represented,
- auto source level adjustment, to keep the signal at a manageable amplitude in one of the channels (and requiring the low-bandwidth closed-loop control noted above),
- active dwell time, to be sure the measurement uses a reasonable number of cycles (or enough averages to get an acceptable variance) at each frequency.

3.2.5.1 Swept Sine Response

One of the most used diagnostics tests is the swept sine measurement which determines a transfer function between two measurement points by applying a sine stimulus and sweeping it through frequency. The parameters of the swept sine test are as follows:

Parameter (Unit)	Description
TestClass	SweptSine
TestName	unique identifier
StimulusChannel	channel name or number of the test input
StartFrequency (Hz)	start frequency of the swept sine (for linear and logarithmic sweeps)
StopFrequency (Hz)	stop frequency of the swept sine (for linear and logarithmic sweeps)
StimulusFrequency (Hz)	list of frequencies (for user sweep)
SweptSineType	linear (default), logarithmic or user
SweptSineDirection	upwards or downwards (default)

1. Most useful if the amplitude for each frequency can be selected in advance to avoid overdriving resonances, for example; since the stimulus is also measured, this variation is divided out

Parameter (Unit)	Description
StimulusAmplitude (V)	amplitude of stimulus; can be a list of (frequency, amplitude) points which is used as an interpolation table (linear interpolation in log-log scale); can be Automatic for auto source level adjustment
StimulusWait (cycles)	time which is waited after applying the stimulus until the test starts default is 0
NumberOfPoints	number of measurement points
MeasurementChannelAN	name or number of the Nth measurement A channel; N = 1, 2, 3,... 20
MeasurementChannelBN	name or number of the Nth measurement B channel; N = 1, 2, 3,... 20
MeasurmentTime (cycles)	time for each measurement point; default is 10 cycles; can be Automatic
MeasurementAverages	number of averages; default is 1.
TestAnalysis	calculates the transfer functions (B/A)
TestResults	returns each measurement point of the transfer function; returns the transfer function at the end of the measurement; returns the coherence spectrum

3.2.6 Triggered Pulse Response

This kind of measurement simply injects pulses of a desired shape into selected test points and triggers an oscilloscope readout to capture one or more responses using multiple display traces (possibly including the stimulus itself). Pulse response waveforms give good indications of propagation delays and non-linearities (especially rate limiting, which is hard to find otherwise). To an experienced operator it also gives a quick, intuitive summary of simple frequency responses and the performance of servo systems.

3.2.6.1 Single Trigger Response

We distinguish between the response to a single trigger and a periodic trigger. A single trigger is useful when the effect of the trigger pulse is so large that the system changes into another state. A typically example is a cavity ring-down measurement. Hence, only one trigger (pulse) is applied whilst the measurement is being taken. The parameters for the single trigger response test are as follows:

Parameter (Unit)	Description
TestClass	SingleTriggerResponse
TestName	unique identifier
TestDuration (s)	time of measurement
TestPreTriggerTime	measurement time before the trigger is applied; default is 20% of the total measurement time
StimulusChannel	channel name or number of the trigger input
StimulusType	impulse, step or ramp

Parameter (Unit)	Description
StimulusAmplitude (V)	amplitude of the stimulus
MeasurementChannelN	name or number of the Nth measurement channel; N = 1, 2, 3,... 20
TestAnalysis	calculates the decay time of the response
TestResults	returns the time series of the measured channels and the decay constants

3.2.6.2 Periodic Trigger Response

When the effect of the trigger doesn’t change the state of the system, a periodic trigger can be applied to improve the signal-to-noise in the resulting time traces by averaging. A typically example is the step response measurement of a servo loop. The parameters for the periodic trigger response test are as follows:

Parameter (Unit)	Description
TestClass	PeriodicTriggerResponse
TestName	unique identifier
TestDuration (s)	time of measurement (trigger to trigger)
TestPreTriggerTime (s)	measurement time before the trigger is applied; default is 20% of the total measurement time
NumberOfTriggers	the number of periodic triggers which are applied to the stimulus channel
StimulusChannel	channel name or number of the trigger input
StimulusType	impulse, step, ramp or triangle
StimulusAmplitude (V)	amplitude of the stimulus
MeasurementChannelN	name or number of the Nth measurement channel; N = 1, 2, 3,... 20
TestAnalysis	averages over repeated measurements; calculates the decay time of the response
TestResults	returns the (averaged) time series of the measured channels and the decay constants

3.2.7 Statistical Analysis of Noise Sources

Data taken during normal operation is useful for characterization of detector noise properties. For example, spectral analysis of length sensing error signals will provide the noise floor of the photodetectors; and histograms of measured physical quantities can be used to characterize non-Gaussian noise. Other statistical methods may be used to both diagnose detector behavior and to create a reduced data stream e.g. principal component analysis.

Occasionally environmental events will stimulate the detector in a way not possible with built-in excitation mechanisms. Understanding both the sources of these events and the resulting detector behavior is a natural job for the on-line diagnostics. Specifically, these events will be used to determine couplings between the environment, environmental monitors and interferometer lengths. The diagnostics system will trigger on these events and capture relevant data or extract it from the LDAS on-line archive for detailed analysis. Standard analysis procedures will be developed for recurring environmental events. These events will also be parameterized and catalogued.

3.2.7.1 Time Series Measurement

The time series measurements takes the time series of a couple of channels and stores it to file. The parameters for the time series measurement are as follows:

Parameter (Unit)	Description
TestClass	TimeSeries
TestName	unique identifier
TestDuration (s)	time of measurement
MeasurementChannelN	name or number of the Nth measurement channel; N = 1, 2, 3,... 20
TestAnalysis	none
TestResults	returns the time series of the measured channels

3.2.7.2 Power Spectrum Measurement

The power spectrum measurements takes the time series of a couple of channels and calculates their power spectra. The parameters for the power spectrum measurement are as follows:

Parameter (Unit)	Description
TestClass	PowerSpectrum
TestName	unique identifier
TestDuration (s)	time of each measurement; default is 10/bandwidth
FrequencySpan (Hz)	highest frequency; default is 8192 Hz
Bandwidth (Hz)	bandwidth; default is 1 Hz
MeasurementChannelN	name or number of the Nth measurement channel; N = 1, 2, 3,... 20
TestAnalysis	calculates the power spectra and averages over consecutive measurements
AnalysisWindow	Hanning, Flat-top or Uniform
TestResults	returns the averaged power spectra after each step

3.2.8 Parameter Dependencies and Optimizations

The section does not introduce new diagnostics tests, but rather extensions to existing tests, so that they can be used to investigate the dependencies of their test results on other parameters, or to find optimum values in respect to the properties of some other parameters.

3.2.8.1 Parameter Sweep

Most of the diagnostics test can be performed repeatedly while sweeping the frequency, the amplitude or the offset of a signal. The following parameters can be added to a diagnostics test to perform a sweep:

Parameter (Unit)	Description
Sweep	must be yes for enabling a sweep; default is no
SweepType	Linear (default), Logarithmic or User
SweepParameter	Frequency, Amplitude or Offset (default)
SweepChannelN[1]	name or number of the Nth sweep channel; N = 1, 2,... 20
SweepFrequencyN	used only when SweepParameter is not equal Frequency (default 0)
SweepAmplitudeN	used only when SweepParameter is not equal Amplitude (default 0)
SweepOffsetN	used only when SweepParameter is not equal Offset (default 0)
SweepStartN	start value of the Nth sweep parameter (linear and logarithmic)
SweepStopN	stop value of the Nth sweep parameter (linear and logarithmic)
SweepPointsN	number of points of the Nth sweep parameter (linear and logarithmic)
SweepDirectionN	upwards (default) or downwards
SweepValuesN	list of values used for the sweep (user)
SweepWait (s)	(settling) time to wait before each measurement
TestResults	returns the default test results for each step of the sweep, together with the used sweep parameter(s)

1. A sweep channel can be identical to one of the normal stimulus channel used in the test. In this case the sweep parameters for frequency, amplitude and offset will default to the ones given for the stimulus channels. In any case the ones specified for the sweep will override the ones for the stimulus channels.

3.2.8.2 Parameter Optimization

Sometimes one wants to minimize a noise coupling by adjusting another parameter such as a servo loop offset. The optimization option can be used to do exactly this. For instance, one might want to optimize the gravitational wave sensitivity as function of the angular offset adjustment. This can be done by repeatly performing a sine response test between the antisymmetric ouptut and a differential arm cavity length drive while changing the dc offset for the angular control loops and picking the best value for each angular degree-of-freedom. This concept can readily be generalized for any test which returns a scalar value function. The optimization option requires that a parameter sweep is specified, since it uses the sweep parameters to determine the parameter space to look for the optimum. The parameters of the optimization option are as follows:

Parameter (Unit)	Description
Optimization	must be yes for enabling an optimization; default is no
FindType	Minimum (sums automatically over a value function which returns a vector), Maximum, Zero, Value, MinimumAbs (minimizes sum of the absolute values), MaximumAbs, MinimumSqr (minimizes the square sum), MaximumSqr
FindValue	specifies the value to be found (if FindFunction = Value)
FindFunction	value function which is used to evaluate the value to be optimized; default is: CoherentRatio for the sine response; HarmonicRatio for the harmonic distortion test, SidebandRatio for the two tone intermodulation test, SpectralDensity for the pseudo-random response, the pseudo-random cross-correlation, the swept sine and the power spectrum measurement TimeSeries for the time series measurement
FindMethod	Scan (default); this method scans through the parameter channels, one after the other, and keeps the best found value for proceeding steps
OptimizationAdjustment	Best for choosing the best parameters at the end of the optimization test (default) or None for setting the parameters back to the values they had before the optimization.
TestResults	returns the default test results at each step of the optimization, together with the used scan parameter(s) and the value function. returns the best test result, together with the best scan parameter(s) and the best value function at the end of the optimization.

3.3 Test Scheduling

Non-invasive tests, once set up, have minimal impact on interferometer operations, and will only need scheduling to coordinate (avoid competition for) resources. The installation of sensing equipment, e.g., accelerometers on the seismic isolation support piers, may interfere with interferometer operation and will require coordination with other planned down-times.

Invasive tests render the interferometer output nominally useless, and thus directly impact the availability of the interferometer. A central coordination of the time allocated to invasive tests is needed, and the amount of time will clearly depend on the usefulness of the data which would otherwise be collected, the way in which the test will help interferometer performance, and the otherwise determined interferometer availability (as limited by failures or adverse environmental conditions).

We distinguish between foreground and background diagnostics tests: foreground tests need to run exclusively; whereas background tests can run concurrently with other tests. Only one foreground test is allowed at any given time; on the other hand as many background tests as useful and possible can run simultaneously.

Since a fair number of the diagnostics tests require the detector to be in a specified state — e.g., the interferometer has to be locked in order to measure the coupling of frequency noise into the gravitational wave band — coordination between the top level controls of the detector and the GDS is important. We require that diagnostic tests can be synchronized using the top level controls, e.g., they have the ability to wait for a start signal (see also section 3.2.1).

3.4 On-Line Diagnostics Operation

On-line diagnostics procedures will, in general, be accomplished with stand-alone programs controlled via EPICS and/or parameter files. On-line procedures are those that involve triggering, data capture or signal injection. Our long term emphasis will be to evolve these procedures toward a minimum of operator interaction. On-line diagnostics analysis techniques will be developed and refined through deferred and off-line diagnostics analyses described below and later incorporated into real-time programs. Those diagnostics analyses not well suited to automation will remain deferred or off-line procedures (see Fig. 7).

3.4.1 On-Line Programming tools

On-line programming will be done in C++ or C.

3.4.2 Supervisory Control

A typical diagnostics procedure will rely on a real-time supervisory program to coordinate excitation signals and acquisition of relevant interferometer channels. The supervisory program shall have the capability for (slow) closed-loop control of excitation signals. The supervisory control program may also initiate predefined real-time diagnostics analyses and a subsequent plotting program to display the results.

3.4.3 Diagnostics Excitation

In general, the subsystems rather than the GDS are responsible for applying the physical excitations. However, it is the GDS which controls the excitations waveforms. In case a subsystem needs an analog signal at its excitation input, the GDS provides instructions to a subsystem remote-controlled (arbitrary) function generator which can apply sine, square, ramp and noise output in the range from 1 mHz to 1 MHz. To support the diagnostics analysis each function generator output is read-back as an additional channel by the DAQ system. On the other hand, for digital subsystems GDS will implement function generators in software which will download waveforms to the subsystem using a digital link. The digital subsystems are then responsible to add the excitation values to the appropriate internal (servo) test points. These software function generators can output arbitrary waveforms in the frequency range from 0.01 Hz to the Nyquist frequency of the sampling rate, and are also read-back by the DAQ system (through a digital link).

Excitations are usually controlled from the diagnostics test tool. However, there is also the possibility that they can be controlled by the operator directly.

3.4.4 Diagnostics Parameter Files

The necessary parameters of a diagnostics test are stored in a parameter file. These files will be a lightweight format. Specifically, this must be a human readable and easily edited format, e.g., ASCII. There will be a mechanism to associate test parameters with individual test results either by directly storing test parameters with analysis results or via a time-tagged parameter database.

3.4.5 Raw Data

Diagnostics procedures which do not complete analysis on-line will produce files or streams of raw data in a lightweight format. Typically these data will be redundant with the LDAS archive and will therefore be temporary files maintained only until deferred analysis is complete. In the case where raw diagnostics data is not redundant it will be backed-up to tape.

3.4.6 Diagnostics Test Results

Results from diagnostics tests whether on-line, deferred or off-line will be stored in a lightweight file format. Their standard contents is described in section 3.2.1. These files will be stored on disk for one year (minimum) and will be organized for easy access. The data format of test results will be kept as standardized as possible, i.e., transfer functions are always stored the same way. Physical quantities will use SI units. We will adhere to the following conventions:

Data type	Format
channel name	null-terminated string
channel number	long (32 bit)
names, file references	null-terminated string
data, time	TBD
natural number	long
real scalar quantity	double precision floating point
complex scalar quantity	pair of double precision floating points (real and imaginary part)
demodulated value	will be stored as a complex number (phase = argument)
real vector or matrix	(multi) dimensional array of real scalar quantities arrays are saved with a header describing their length(s)
complex vector or matrix	(multi) dimensional array of complex scalar quantities arrays are saved with a header describing their length(s)
FFT of a time series	will be stored as two complex vectors (frequency points, FFT values) FFT amplitudes are complex numbers; no ‘negative’ frequencies are store when the original data is real
power spectrum	will be stored as two complex vectors (frequency points, power)
transfer functions	will be stored as two complex vectors (frequency points, ratio) ratio is a complex number
parameter values	ASCII; same convention as parameter files

Diagnostics results are presented using a high-level analysis and graphics package (Mathematica and Matlab). The GDS will support functions such as automatic loading of data, displaying of numerical values, plotting of results and generating reports. The following plots will be standard:

Data type	Format
time series	tY plot
multiple time series	tY plot with multiple functions
FFT of a time series	fY plot (showing two functions for real and imaginary part)
power spectrum	fY plot
transfer function	Bode plot

All plots will support an option for using an interpolation function between the data points. Due to the flexibility of the used graphics packages more advanced plots such as histograms can easily added when needed.

3.5 Deferred and Off-Line Diagnostics Analysis

3.5.1 Deferred Diagnostics Analysis

Diagnostics procedures which are not well defined, are under development or require interactive operation may be performed manually as deferred analyses. In such cases, the diagnostics test tool must provide raw channel data as described above. Deferred analysis will be performed in a supported analysis environment or with stand-alone user programs. Deferred diagnostics analysis programs will be transportable to off-line diagnostics analyses described below. This may require that the data input functions be decoupled from the analysis procedures.

3.5.2 Off-Line Diagnostics Analysis

Off-line diagnostics analysis will proceed identically to deferred analysis described above with the exception of the input raw data which in this case will be derived from frame data in either the on-line or off-line LDAS archive.

3.5.3 Supported Analysis Packages

Supported interactive analysis environments will be Unix/C/C++, Mathematica and Matlab.

3.5.4 Diagnostics Test Results

Deferred and off-line diagnostics analyses will produce data files of results identical to those described for on-line diagnostics analyses. They will also use identical plotting software to present the results.

3.5.5 Updating Interferometer Parameters

As a result of a diagnostics test it might be determined that some of the present interferometer parameters, such as a servo loop gain, are no longer optimal and need to be updated. Currently, no fully automatic interferometer parameter update mechanism is foreseen; operator intervention will be required to make sure that new parameters are propagated and logged correctly.

3.5.6 Diagnostics History and Trend Analysis

Diagnostics analysis result files will be stored on-line for one year (minimum) and will be organized for quick access.

Long-term trend analysis will be supported on either trend data stored in the LDAS frame archives or in the diagnostics history database. Trend analysis differs from off-line analysis only in its access to the diagnostics history.

3.6 List of Diagnostics Tests

See Appendix C.

4 System Design

GDS software complies with the software standards described in LIGO-T970211-00- and LIGO-T960004-A. The GDS hardware design is compatible with the CDS control and monitoring system, LIGO-T970171-00, and the LIGO data acquisition system, LIGO-T960009-A and LIGO-T970136-00.

4.1 Overview and Basic Concept

4.1.1 Implementation Issues

An overview of the components of the GDS and their interaction with the DAQ system is shown in Fig. 9. It is divided into a front-end part (VME based) and a back-end part (workstation based). The data acquisition system is responsible for collecting real-time data through DAQ owned ADCs, through links to the digital subsystems and the excitation system and by reading back pre-

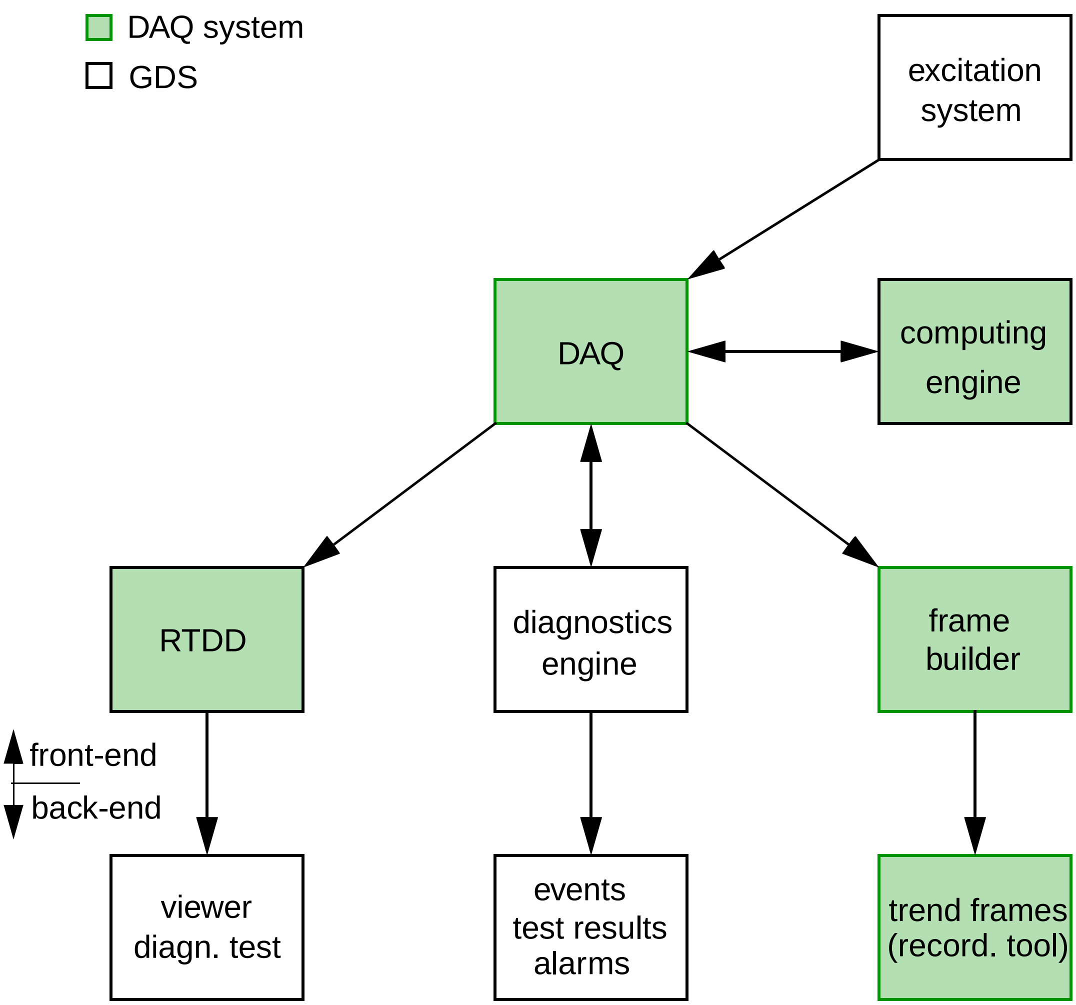


Figure 9: Main data paths of the global diagnostics system.

analyzed data from the computing engine and the diagnostics engine. It also serves as a distribution center for data. The real-time data distribution (RTDD) system implements a client-server architecture which is able to push data in (near) real-time to ‘customers’ such as the viewing tool and the deferred analysis task of the diagnostics test tool. The on-line search tool and the real-time analysis part of the diagnostics test tool are collectively called the diagnostics engine which will have access to the DAQ data at the front-end through shared memory (i.e. reflected memory modules). They will send test results, alarms and events to the operator consoles in the control room using the CDS fast ethernet/ATM network. The DAQ system also implements a frame builder which archives the data. Trend frames are written once every second summarizing channel information (average, highest/lowest, r.m.s., etc.); thus, implementing the recording tool. The computing engine will be responsible for the on-line FFTs calculations.

The global diagnostics system is implemented only once per site. However, one excitation system is provided per detector.

4.1.2 Task Scheduling

Data gathered by the DAQ system will be available every 1/16 sec. This timing is maintained by a GPS clock which will also be available to synchronize the GDS back-end workstations. Diagnostics software components running on the front-end (excitation system, RTDD, diagnostics engine and computing engine) will then be scheduled to restart at the beginning of each time interval, to process the data of the corresponding time interval and to terminate, before the next interval starts. Time intervals are numbered from 0 to 15 beginning at each second boundary. Typically, the frame builder will read channel data from the GDS only once every second. In any case it will not wait longer than 2/16 sec.

Software components residing on workstations will mainly act as data clients. Their main duty will be receiving data and results from their corresponding front-end tasks and display, store or analyze it, respectively.

4.1.3 Supervisory Control

Supervisory control is provided through EPICS. Users will have the capability to start a new diagnostics test, view a number of channels simultaneously, look at the event log of the search tool, reconfigure the trigger bank and the recording tool among others. Supervisory tasks are located in front-end CPUs, but will use GUIs running on back-end workstations.

4.1.4 Configuration of Diagnostics Components

The behavior of diagnostics software components will be determined by parameter files which can be changed by the operator. For instance, triggers are organized into separate trigger banks, each of them having its own parameter file listing the parameters for each trigger. Similarly, each diagnostics test will have its own parameter file describing the exact test procedure. Parameters files are organized as simple ASCII files. They are read by the supervisory control and ‘uploaded’ to their corresponding diagnostics component for initialization and on-the-fly reconfiguration.

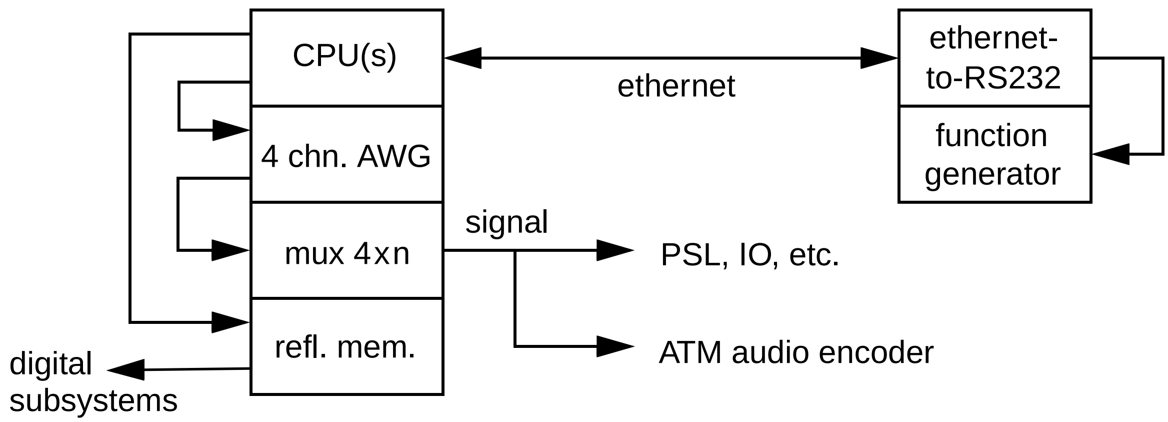


Figure 10: Diagnostics Excitation System

4.1.5 Excitation System

The excitation system uses function generators — either remote-controlled hardware units or software simulations — to provide the required excitation waveforms. Both implementation use the same supervisory control interface (based on EPICS) to communicate with the diagnostics test tool. Digital waveforms are generated by the diagnostics excitation system and then transferred to the digital subsystems using reflected memory. The diagnostics excitation system also uses a 4 channel VME based arbitrary waveform generator followed by an analog multiplexer to supply test signals to analog subsystems (see Fig. 10). Additionally, stand-alone function generators which can be carried around will be supported through ethernet-to-RS232 converters.

Those function generator are usually controlled from the diagnostics test tool. However, there is also the possibility that they are controlled by the operator directly. The arbitrary waveform generators will serve double duty as audio synthesizers and send audio channels to the ATM video/audio encoders.

4.1.6 Front-End

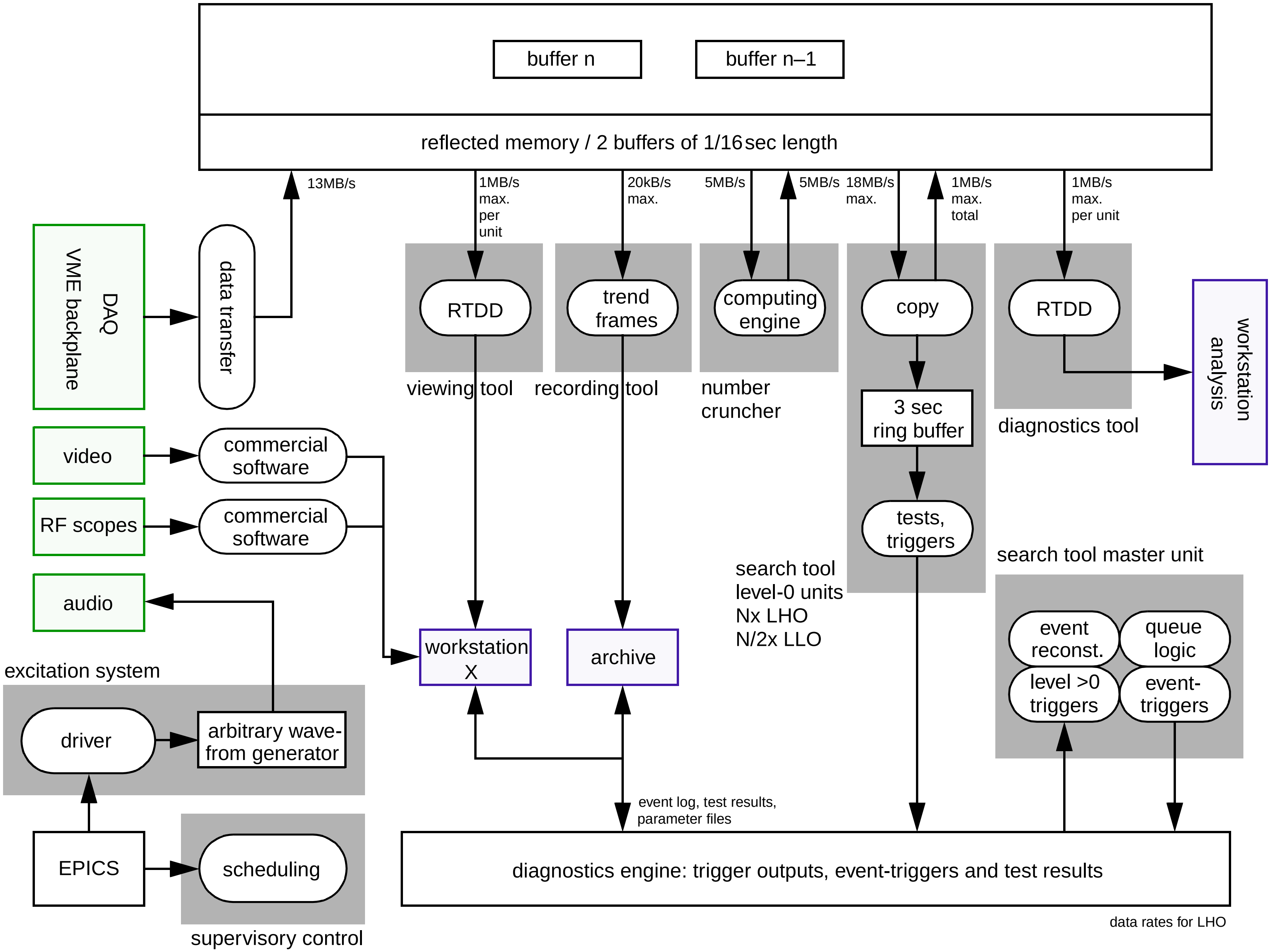
An overview of the data flow and the software tasks running on VME front-end processors is shown on Page 62.

4.1.6.1 Reflected Memory (RM)

4.1.6.1.1 Data Flow and Rates

The reflected memory serves as a buffer for channel data. Raw data from the detector is copied into the RM by the DAQ system at a rate of approximately 13MB/s (LHO) and 6.5MB/s (LLO), respectively. The RM also stores real-time FFTs of the DAQ channels and the flag channels generated by the search tool. The RM is read by RTDD, the computing engine, the diagnostics engine and the trend frame builder. A summary of the corresponding data rates is presented below:

Software front-end



Description	Data rate		Unit	Type
	LHO	LLO		
Data transfer from DAQ system (typical)	13	6.5	MB/s	write
Computing engine[1] (typical)	5 / 5	2.5 / 2.5	MB/s	read / write
Viewing tools[2] (typical)	2	1	MB/s	read
Recording tool[3] (typical)	0.02	0.01	MB/s	read
Search tools[4] (typical)	18 / 1	9 / 0.5	MB/s	read / write
Diagnostics tests tool[2] (typical)	2	1	MB/s	read
Total (maximum)	27 / 19	13.5 / 9.5	MB/s	read / write

1. based on the assumption that one FFT is performed on half the channels which are not excitation read-backs.
2. based on an estimate of looking at 30 fast channels simultaneously.
3. based on an estimate of storing 20 bytes of information per second for every channel.
4. based on an estimate that each channels is processed exactly once.

4.1.6.1.2 Organization

The RM is divided into 2 buffers, each covering the data of one time interval of 1/16 sec. It is organized as a ring buffer representing the current data and the data from the previous time interval. The current buffer is used by the data transfer task which copies the data from the DAQ, whereas the previous buffer is used by diagnostics tools to access the data.

4.1.6.1.3 Memory Requirements

With a data rate of 18 MB/s (LHO) and 9 MB/s (LLO) for all channel data the minimum required size of the RM is 8.2 MB and 4.1 MB, respectively (FFTs are calculated only once a second).

4.1.6.2 Platforms

4.1.6.2.1 Processing Requirements

Front-end tasks will run on VME hosted CPUs which are linked both to the RM. A crude estimate of the needed processing power is presented below:

Diagnostics task	Assumption	CPU Power		Unit
		LHO	LLO	
Number cruncher	FFT on half the channels, typical length 16k points	400	200	MFLOPS
Viewing tool	8th-order filter on 50 channels simultaneously	30	30	MFLOPS
Search tool	2000/1000 triggers, 2 operations per point	70	35	MFLOPS
Diagnostics tool	e.g. down-conversion on 10 fast channels[1]	35	35	MFLOPS

1. Assumes sine and cosine functions are 100 times more expensive than multiplications.

4.1.6.2.2 Memory Requirements

The computing engines has to buffer data before performing the FFTs, because the data will only be available for a short time in the RM. The diagnostics engine has to buffer at least 2 sec worth of data and, therefore requires the implementation of a 3 sec ring buffer with a total size of 57 MB (LHO) and 29 MB (LLO), respectively. Of course, if multiple CPUs are used to implement the diagnostics engine, the memory can be distributed. On the other hand if individual triggers make use of large chunks of private memory the requirement could be higher.

4.1.7 Front-End/Back-End Communication

4.1.7.1 Low Speed and Supervisory

The supervisory control communication and the transfer of configuration data and will rely on EPICS channel access running over the CDS net. Additionally, the diagnostics archive will be accessible through an NFS mounted file system, so that parameter files can be directly loaded from file.

4.1.7.2 High Speed

The transfer of high speed, real-time channel data to the diagnostics back-end workstations will be done by the RTDD system; it will use the CDS fast ethernet/ATM network. The transfer of asynchronous data from the GDS diagnostics engine to the archive and display will also be routed over the CDS fast ethernet/ATM network.

4.1.7.2.1 Throughput Requirements

An estimate of the maximum required throughput:

Front-end	Assumption	CPU Power		Unit
		LHO	LLO	
Viewing tool	50 channels simultaneously	1600	1600	kB/s
Diagnostics tool	20 channels at full speed	640	640	kB/s

4.1.7.2.2 Network Protocol

For the software interface between front-end processors and back-end workstations we propose to use direct calls to TCP/IP sockets.

4.1.8 Back-End

4.1.8.1 Software Tasks

The viewing tool, the deferred analysis of diagnostics tests and the event reconstruction will run on back-end workstations.

4.1.8.2 Platforms

4.1.8.2.1 Processing Requirements

Workstations running UNIX will be used as diagnostics consoles and as data clients displaying plots, analyzing diagnostics tests and reconstructing events. CPU requirements for the diagnostics console and the diagnostics archiver are modest.

4.1.8.2.2 Memory Requirements

Memory requirements on the diagnostics console and the diagnostics archiver are modest.

4.1.8.2.3 Storage Requirements

The diagnostics archive must be able to store one year of recorded data, one year of diagnostics test results and one month of raw channel data associated with diagnostics tests; next to current versions of the parameter files and software executables. The diagnostics archive is periodically backed-up to tape.

Storage	Description	Storage space		Unit
		LHO	LLO	
Diagnostics archive	on-line disk storage speed of disk access not relevant	100	50	GB

4.1.8.3 User Interfaces

A list of user interfaces accessible by an user or an operator is presented below. In general, these interfaces are available in the control room; or by remote-login into a computer located in the control room. Only permanent diagnostics screens showing detector overviews are automatically exported to the LIGO web page and are periodically updated every 10 min.

GUI	Description
Main diagnostics	Supervisory control Configuration control of diagnostics engine starts diagnostics engine
Alarm handler	shows alarm raised by triggers
Viewing tool	displays plots of data and reconstructed events
Audio	audio-jack to connect a headset
Video	displays images of video cameras
Permanent diagnostics screens	displays permanent diagnostics screens exports permanent diagnostics screen to the web
Diagnostics tests	starts diagnostics tests
Diagnostics analysis & test results	analyzes diagnostics tests and displays results
Parameters editor	Change parameter files
Excitation system	remote-control of a function generator

4.2 Software Design

4.2.1 Channel Access API

4.2.1.1 Data Transfer from the DAQ System

The diagnostics engine and the computing engine have access to the reflected memory of the DAQ system. Block transfers are used to copy channel data into local buffers.

4.2.1.2 Look-up Table for Channel Addresses

A dictionary describing channel information will be stored in the DAQ reflected memory. This dictionary is described in the CDS DAQ documentation and contains a channel description. The API for channel access will be provided by the DAQ system.

4.2.1.3 Channel Read/Write Library

This library is intended to be used by the diagnostics engine to access the internal ring buffer. The following functions are provided to retrieve channel information, to read data from a channel and to write to a channel:

Function	Arguments	Action
GetChannelInfo	channel number return: channel info	returns the channel information associated with the specified channel (channel name, the channel address in the RM, the sampling rate, the data type (int16 or double), the time when this info was last updated etc.)
ReadChannel	channel info local buffer info RM buffer time return: error code	copies a channel from the RM into a local buffer the local buffer info contains information about the local buffer (buffer address, data type, decimation factor, etc.)
WriteChannel	channel info local buffer info RM buffer time return: error code	copies a channel from the local buffer into the RM error codes: unknown channel time interval passed

Channel addresses and sampling rates might change on-the-fly. This will be reflected in the dictionary. To hide a reconfiguration from the user as much as possible the channel information structure carries a time stamp when its channel information was last changed. Both ReadChannel and WriteChannel check before each data transfer whether the configuration has changed in the mean time and if yes, automatically call GetChannelInfo to update the information. They also check whether the data are still valid in the specified RM buffer by comparing the given time with the current time.

4.2.2 Data Archive API

4.2.2.1 Archive Index

The primary organization of the GDS archive index will be by diagnostics data type with a time-stamp further ordering data within a given type. This organization will provide fast access to time windows of diagnostics data as well as good random access. Access to the index will be via functions which create filenames and through utility programs calling these index functions.

4.2.2.2 Archive Data Formats

Lightweight data will be either straight ASCII or preferably conform to the netCDF format as described in “A lightweight data format for LIGO, version 1.01” (Tom Prince). Straight ASCII files may be either formatted (e.g., parameter files) or free-form (notes and log files)

High level access to the archive will be provided through the archiver and parameter manager servers described in Section 4.2.3. Unix based analysis processes may utilize the archive servers or the archive C++ API. Access to data in the GDS archive will typically be via the index interface.

4.2.2.2.1 Parameter Files

Parameter files will be human readable, preferably formatted ASCII. Major groups of diagnostics tasks may share parameter files with parameters for individual tasks stored in localized sections. A history mechanism will allow access to old parameter values. The implementation of this latter function is TBD.

4.2.2.2.2 Raw & Fourier Transformed Data

Raw data files will contain an ASCII header specifying at a minimum the channel names, acquisition time, duration and sampling rate or frequency bin size followed by an ASCII or binary data block. Multiple channel data may be stored in the same file provided it is of the same epoch, type and sampling rate. Multiple channel files will have additional header parameters specifying additional channel names.

4.2.2.2.3 Diagnostics Results

Scalar diagnostic results (e.g., operating parameters) will be time tagged within a files of like diagnostic data. Vector and vector array results will be stored in individual netCDF files. Result file headers will contain at a minimum diagnostics task descriptor, key parameter values and epoch (see also Section 3.2.1).

4.2.2.2.4 Event Trigger

Event trigger files will contain at a minimum an ASCII header containing a trigger descriptor which uniquely identifies the channels involved, the value which produced the trigger condition, the epoch of the trigger and its priority. Additionally, the file may contain binary raw or processed data.

4.2.2.2.5 Video & RF Capture

When archived, image and video data will be in standard file formats accessible via free or commercially available Unix software. Data traces will be stored as netCDF files.

4.2.3 Network Transfer API

4.2.3.1 Real-Time Data Distribution (RTDD)

The RTDD interface and implementation will be provided by the DAQ system. It will be used by both the viewing tool and the deferred analysis task of the diagnostics test tool.

4.2.3.2 Parameter Files

Since the diagnostics archive will be mounted as an NSF file system, parameter files can be directly loaded from file. The following functions are supported by the parameter file API:

Function	Type	Arguments	Action
ReadParam	called by client blocking	service name diagnostic type parameter section parameter name return: error code parameters	fetches a single parameter from server time-out: try again later error codes: server down or not reachable file does not exist section does not exist
ReadParamSection	called by client blocking	service name diagnostic type parameter section return: error code section	fetches a parameter section from server time-out: try again later error codes: server down or not reachable file does not exist section does not exist
ReadParamFile	called by client blocking	service name diagnostic type return: error code file	fetches a parameter file from server time-out: try again later error codes: server down or not reachable file does not exist

4.2.3.3 Trend Files

Trend files will be generated by the frame builder. Frame file access will be provided by DAQ/DAS.

4.2.3.4 Test Results and Events

Test results and events are asynchronous data. They are transferred to the back-end workstations using a message-type format. Each message will have a destination address, a data type descriptor and a data field. Transfers over the CDS network are routed through TCP/IP sockets.

Test results and event lists are save to the diagnostics archive using the NFS mounted file system.

An event queue manager will run on the back-end and will offer two basic services: submitting events and event-triggers to the tail of the queue and requesting triggers from the head of the queue. For submission, the event manager will parse the incoming data in order to determine its type and priority in the event queue and whether reconstruction is required. For event requests, the event manager return the next event or events and will optionally screen for event-triggers of a given type or priority. The event queue manager will also be responsible to perform the event reconstruction. The following functions are supported by the API of the event manager:

Function	Type	Arguments	Action
PutEvent	called by client blocking	service name return: error code	places an event on event queue time-out: try again later error codes: server unreachable
PutEventList	called by client blocking	service name return: error code	places an event list on event queue time-out: try again later error codes: server unreachable
GetEvent	called by client blocking	service name type | priority return: error code file	fetches next event from queue time-out: try again later error codes: server unreachable
GetEventList	called by client blocking	service name type | priority return: error code file	fetches qualifying events from queue time-out: try again later error codes: server unreachable

4.2.3.5 Error Messages

One system console in the control room will be devoted to collected all error, warnings and debug messages. A scheme of passing messages over TCP/IP sockets will be used. A naming convention and a color scheme will be used to distinguish different message origin and importance. The following functions are supported by the system console message API:

Function	Type	Arguments	Action
ConsoleMessage	called by client not blocking	service name subsystem name message string return: none	Writes a message to the system console
DebugMessage	called by client not blocking	service name subsystem name message string return: none	Writes a debug message to the system console
ErrorMessage	called by client not blocking	service name subsystem name message string return: none	Writes an error message to the system console
WarningMessage	called by client not blocking	service name subsystem name message string return: none	Writes a warning to the system console

4.2.4 Supervisory Control

Control and configuration of real-time diagnostics tools will be via parameters broadcast on EPICS channels. This control is primarily through high-level commands, both binary on/off (e.g. StartTaskN) and configuration commands (e.g. ActiveShakerID). The supervisory control program is a simple back-end process that provides a user interface to EPICS channel values. MEDM may prove adequate for this task. Front-end processes that monitor the EPICS supervisory channels will spawn appropriate diagnostics processes when requested. These processes will auto-configure on start-up via the parameter manager.

4.2.5 Recording Tool

The recording tool will be implemented by the DAQ system using trend frames.

4.2.6 Search Tool

4.2.6.1 Organization and Supervisory Control

The search tool is implemented as front-end tasks running on the diagnostics engine. The search tool is auto-configured on start-up via parameter files whereas high-level supervisory control is provided through the following EPICS command channels.

EPICS Channel	Type	Contents	Description
ST_Submit	Binary	Command	Notify search tool of new control
ST_Disable	Binary	Command	Disable trigger operations
ST_Engage	Long	Trigger Bank ID	Enables specified trigger operation.
ST_Halt	Long	Trigger Bank ID	Terminates specified trigger operation.
ST_Reconfigure	Long	Trigger Bank ID	Command to reconfigure trigger bank from parameter server.

Event triggers are generated when conditions specified by configuration parameters are met. The search tool passes these event triggers to its back-end server, the event queue manager.

4.2.6.2 Trigger Operation and Hierarchy

Triggers are organized into functional banks which group like or related channels or operations. Trigger banks are organized into a hierarchy of levels. The lowest level trigger processes operate on individual channels. In general, higher level triggers operate on the conditional output of lower level trigger processes (see Section 2.3.2).

Usually triggers have access to 2 sec worth of data buffered by the diagnostics engine. Some triggers operate on longer time series of filtered data and require a private memory buffer to maintain adequate look-back time. The memory requirements for these buffers are modest (of order 100 MB). In some cases event triggers are bundled with time series or spectral data. These data blocks are maintained in ring buffers by individual trigger processes.

The initial CPU requirements of the search tool are specified in Section 4.1.6.2. Most likely this will require the implementation of a multiple CPU system.

4.2.6.3 Flag Channels

Output from selected triggers are organized as flag channels. These flag channels are written back to RM by the trigger bank processes where they can be accessed by the RTDD and the frame builder. This feature allows the implementation of multi-bit flag registers which summarize the output of a set of triggers, the implementation of sophisticated trigger algorithms for the viewing tool and the implementation of channel preprocessing in real-time outside the computing engine.

4.2.7 Event Queue Manager

The EQM is a back-end server process running on a GDS workstation serving the search tool. Event triggers are received from the search tool and evaluated by the EQM. Events of sufficient priority are posted to the operator. Reconstruction processes are spawned for events requiring further analysis. In all cases, the event triggers are posted to the archive server for long-term storage.

4.2.8 Excitation System

The excitation system is interfaced via EPICS. It will either program a remote analog function generator or calculate the required waveform and send it to the arbitrary waveform generator or a digital subsystem. The excitation system is connected to the DAQ system through reflected memory, so that excitation channels can be read back.

The excitations system will also provide the audio channels by reading channel data directly from the DAQ reflected memory and send it to the arbitrary waveform generator. The analog audio signal will then be fed into the ATM video/audio encoder. Listening to an audio channel will be using a commercial available software to decode the audio stream on the back-end workstations or using the ATM video/audio decoder with a loud speaker in the control room.

4.2.9 Diagnostics Test Tool

The diagnostics test tool is a front-end process responsible for real-time synchronization of diagnostic procedures as well as capture and/or analysis of DAQ channel data. The algorithmic content of these procedures is covered in Chapter 3 and Appendix C.

4.2.9.1 Supervisory Control

The diagnostics tool comprises a master process which spawns specific tasks when requested. High-level control of the diagnostics tool is through the following EPICS channels:

EPICS Channel	Type	Contents	Description
DT_Submit	Binary	Command	Notify the diagnostic tool of new control
DT_Disable	Binary	Command	Disable all diagnostics tasks
DT_Shutdown	Long	Task descriptor	Halt specified diagnostics jobs

EPICS Channel	Type	Contents	Description
DT_Task	Long	Task descriptor	Identify current task
DT_ParamN	Analog	Task parameter N	Configuration parameter for current task

In general, test parameters are specified with parameter files; however provision is made for “tuning” parameters, which change from run to run, to be specified through EPICS channels.

4.2.9.2 Data Flow and Processing Requirements

The diagnostics tool has the capability to acquire a substantial fraction of the IFO data for limited periods of time.

Deferred analysis tasks will run on back-end workstations receiving their data from the RTDD system. They will use the memory in the workstations to buffer the data. On-line analysis tasks will run on the diagnostics engine following the trigger concept, so that they can be integrated into the search tool architecture. After finishing the analysis they will raise a trigger and send the result data to the back-end workstation for display as a special type of event. Some diagnostic tasks operate on long time series of filtered data and require a memory buffer to maintain adequate look-back time. The memory requirements for this buffer are modest (of order 64 MB).

The initial CPU requirements are listed in Section 4.1.6.2. As algorithms are refined, substantial (~100 MFLOPS) computing resources may be required for the diagnostics tool.

4.2.9.3 Excitation Control

The diagnostics test tool supervisory task issues excitation commands to the excitation subsystem via EPICS on the CDS network. The excitations system then decides whether it has to invoke a remote analog function generator, the arbitrary waveform generator or send waveforms directly to digital subsystems. The coordination of excitation and data collection is the task of the diagnostics test tool supervisory task and may be accomplished open loop (specifying delay times) or closed loop (monitoring of data channels).

4.2.10 Viewing Tool

The viewing tool will be using the RTDD system to obtain its data. The back-end component is responsible to read the needed channels from the RTDD, to do all the necessary preprocessing (see section 2.2.2.1), to make plots (see section 2.2.3.2) and to display this information to the operator.

4.2.11 Graphical User Interface

4.2.11.1 Supervisory Control

Since EPICS is the used for the supervisory control of the global diagnostics system, its graphical user interface will be compatible with the CDS control and monitoring system, i.e. MEDM.

4.2.11.2 Real-Time Viewing

The viewing tool has to display great amount of channel data fast. The possible plots and their options are predefined (see section 2.2.3.2). The display package which is required to implement the viewing tool is TBD.

4.2.11.3 Permanent Diagnostics Screens

Permanent diagnostics screens are build up from the same display package used in implementing the viewing tool. Periodically, these screens are saved to disk as bitmap files. To enable monitoring the performance of the instrument from anywhere in the world, web browsers will be able to access these bitmap images.

4.2.11.4 Parameter Update

At this time parameter files are updated by using a text editor.

4.2.11.5 Off-Line Viewing

For viewing results of diagnostics tests or results of an off-line diagnostics analysis commercially available high-level graphics and analysis packages will be supported. Presently, we intend to support Mathematica and Matlab.

4.2.12 Computing Engine

The computing engine will be implemented as a stand-alone unit which requires no or only very few interactions with other diagnostics tasks. The computing engine will directly access the DAQ reflected memory. The program logic of the number cruncher is straight forward, see Fig. 11. It will be implemented by the DAQ system.

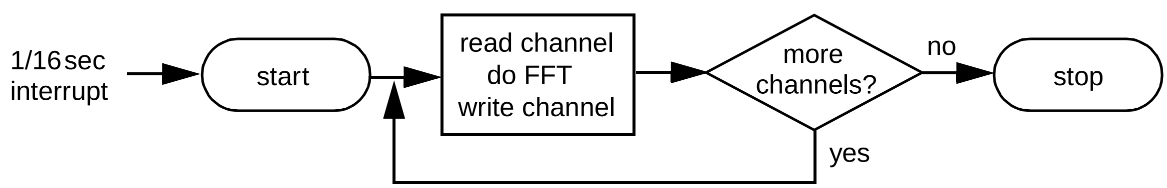


Figure 11: Program logic of the number cruncher.

4.3 Hardware Layout

4.3.1 Overview

An overview of the Hanford GDS hardware layout is shown in Fig. 12. The GDS consists of three VME crates, one with a split backplane which contains the processors for the search tool (Search Engine 1 & 2), and one Excitation Engine crate for each interferometer.

4.3.2 Interfaces

Interface to the Data Acquisition System (DAQS) is via reflected memory. The size of this memory is 4MBytes, with the DAQS collecting and sending data in the top 2MBytes. The remaining 2MBytes are for use by the GDS. The DAQS sees all 4MBytes, thereby allowing the GDS to send data back to the DAQS for data storage, display and distribution to other systems. The Interferometer Sensing and Control (ISC) processors also see the GDS reflected memory, thereby allowing the GDS to feed excitation signals to these controllers. This reflected memory network is further described in LIGO-T980017-00-C DAQS Reflected Memory Network Design.

The processors and DSP units also connect, via ethernet, to the CDS networks. This is used for initial booting of the system and operator control of the GDS.

4.3.3 Search Engines

The search engines run the search tool software described previously. The conceptual design for this system is a VME crate with the following modules:

- Heurikon MIPS 200E processor, running VxWorks real-time operating system.
- Reflected memory module (4MBytes)
- Multiple C40 DSP module
- GPS module (time stamping / data clocking)

In this configuration, the MIPS processor primarily provides for synchronization and data movement between the reflected memory, DSP and the DAQS. Since much of the data analysis requires FFT calculations, an expandable VME DSP module is chosen to perform this function. This DSP unit is a Spectrum 123S, which can carry up to six Texas Instrument C40 DSP. These particular units are presently in house for testing as part of ISC controls and will be evaluated for this use.

4.3.4 Excitation Engines

The excitation engines are used to run diagnostic tests on the interferometer systems. Therefore, they must provide direct analog stimulus to various systems and digital stimulus (via reflected memory) to ISC systems.

These units then consist of:

- Heurikon MIPS 200E processor, running VxWorks real-time operating system.
- Reflected memory module (4MBytes)
- Arbitrary Waveform Generator (AWG)
- GPS module (time stamping / synchronized data clocking)

Several options are being explored for the AWG. One option is a commercial VME AWG module, with models available from several vendors. If these are found not to meet LIGO requirements, this unit will be a multiple channel DAC (such as the ICS-115), with output controlled by the CPU. Clocking of the AWG would be provided by the GPS module.

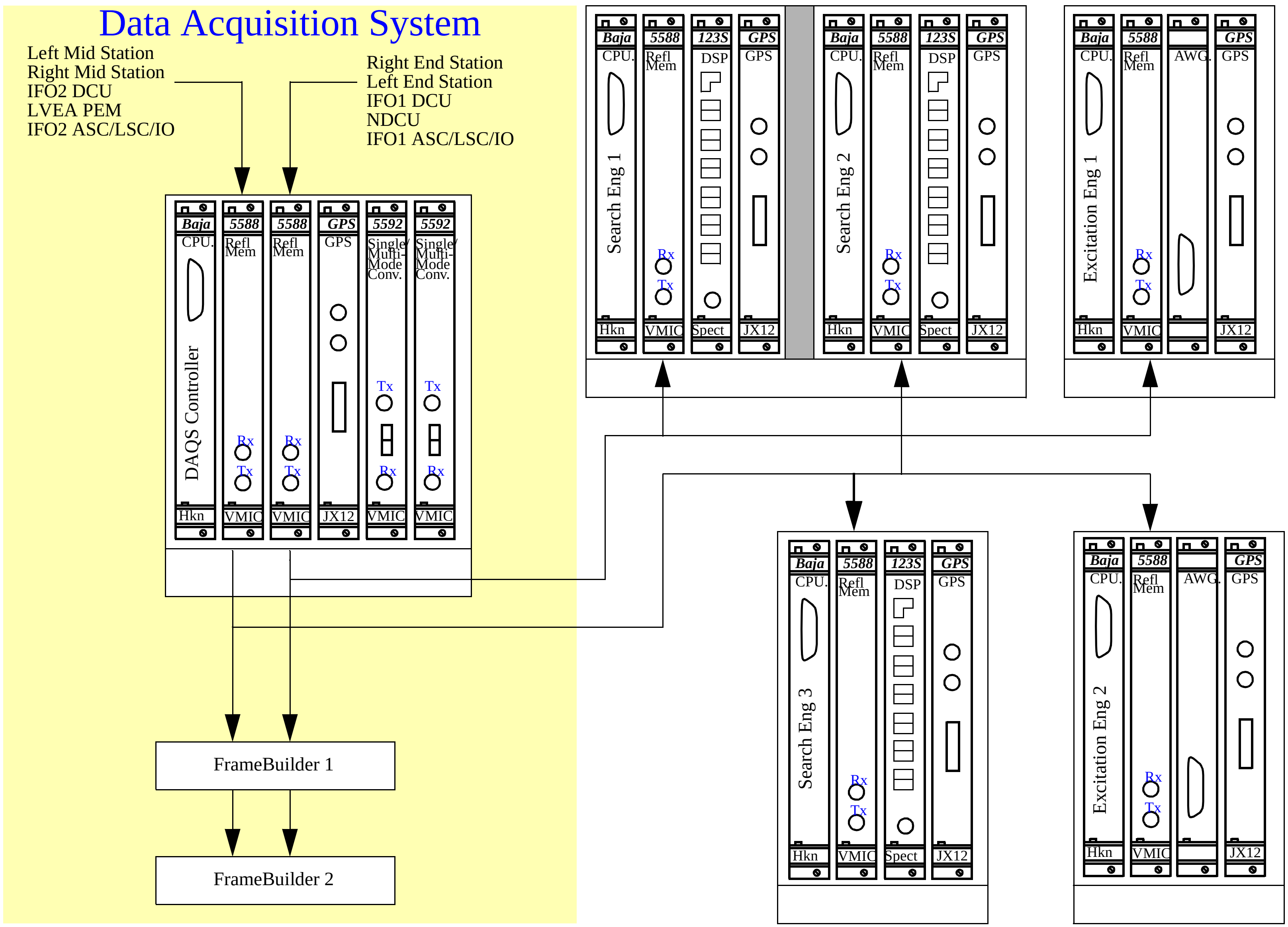


Figure 12: GDS Hardware Layout / Interfaces

APPENDIX A LIST OF CHANNELS

A.1 AUDIO-FREQUENCY CHANNELS

See supporting document T980004-00-D.

A.2 RECORDING CHANNELS

See description of trend frames in T960009-A-D.

APPENDIX B LIST OF TRIGGERS AND EVENTS

B.1 TRIGGERS

TBD.

B.2 EVENTS

TBD.

Appendix C List of Diagnostics Tests

This list is intended to be a 'living document' which will require significant refinement and amplification to establish sufficient information to code tests. Its initial use is to establish the test templates, magnitude of the task, and required hardware. The initial source of the information is LIGO-T970078-00, *"Interferometer Diagnostics Tests and Tools"*.

C.1 Sensing Noise Tests

C.1.1 Laser Frequency Noise in the Gravitational Wave Band

Test Description: The sensitivity to laser frequency fluctuations of the gravitational wave output at the antisymmetric port of the beamsplitter is measured by impressing a swept sine and/or a wide band random noise at the controller or summing junction of the laser frequency control loop. The transfer function of the demodulated interferometer output at the antisymmetric port to frequency fluctuation stimulus as measured at the demodulated recycling mirror reflection port is determined. The noise budget attributable to frequency noise is determined by measuring the cross power spectrum (covariance) of the interferometer output at the antisymmetric port with that at the reflection port of the recycling mirror with the stimulus off.

- technique: transfer function from laser power variation to LSC sensing ports, followed by cross power spectrum measurement of ambient power fluctuations and selected LSC port
- stimulus: sinusoidal variation of the laser power, 10^{-7}-0.01 Hz/$\sqrt{\mathrm{Hz}}$, 1 Hz to 10 KHz
- response: each LSC sensing port
- analysis: standard transfer function tools
- visualization: log-log transfer function, log-log contribution to strain spectrum, coherence or other measure of confidence
- recurrence: once per week or per change in hardware
- duration:60 sec unavailability
- contingent tests: input beam angular deviations, input beam amplitude fluctuations

C.1.2 Laser Amplitude Noise in the Gravitational Wave Band

Test Description: The sensitivity of the gravitational wave output at the antisymmetric port of the beamsplitter to amplitude fluctuations of the input light is measured by impressing a swept sine and/or a wide band random noise at the summing junction of the amplitude control loop. The transfer function of the demodulated interferometer output at the antisymmetric port to the signal at the common mode Michelson port is measured. The noise budget attributable to amplitude fluctuations is determined by measuring the cross power spectrum (covariance) of the interferometer output at the antisymmetric port with that at the Michelson common mode port with the stimulus off.

- technique: transfer function from laser power variation to LSC sensing ports, followed by cross power spectrum measurement of ambient power fluctuations and selected LSC port
- stimulus: sinusoidal variation of the laser power
  - $10^{-8} - 10^{-3}$ modulation depth
  - 1 Hz - 10 kHz

- response: each LSC sensing port
- analysis: standard transfer function tools
- visualization: log-log transfer function, log-log contribution to strain spectrum, coherence or other measure of confidence
- recurrence: once per week or per change in hardware
- duration:60 sec unavailability
- contingent tests:

C.1.3 Amplitude Noise at the Sideband Frequency

- technique: transfer function from modulation to LSC sensing ports, followed by cross power spectrum measurement of ambient modulation fluctuations and selected LSC port
- stimulus: sinusoidal variation of the modulation index for each modulation frequency
  - 1% of nominal modulation
  - 1 Hz - 10 kHz
- response: each LSC sensing port
- analysis: standard transfer function tools
- visualization: log-log transfer function, log-log contribution to strain spectrum, coherence or other measure of confidence
- recurrence: once per month or per change in hardware
- duration:60 sec unavailability
- contingent tests:

C.1.4 Amplitude Noise due to Unintended Interferometers

- technique:
  - (1) determine effective amplitude of unintended interferometers, via modulation of the laser frequency either in a sweep or with sinusoidal modulation; study of interferometer spectrum for spectral ‘cliff’ (up-conversion)
  - (2) determine ambient variations in critical paths (identified above) through...addition of sinusoidal modulation of the position of one of the optics in question with a ‘shaker’ and synchronous demodulation to find baseband motion?
- stimulus (1): sinusoidal variation of the modulation index for each modulation frequency
  - 1% of nominal modulation
  - 1 Hz - 10 kHz
- stimulus (2): PEM shaker driven by CDS
- response: each LSC LSC sensing port
- analysis: Inference of straylight amplitude; inference of physical motion of scatterers
- visualization: log-log transfer function, log-log contribution to strain spectrum, coherence or other measure of confidence
- recurrence: once per month or per change in hardware
- duration: 10 min
- contingent tests:

C.1.5 Noise due to Input Beam Position and Angle Fluctuations

- technique: transfer function from modulation from IO last mirror angle to LSC length sensing ports, followed by cross power spectrum measurement of ambient motion fluctuations (NEED SENSOR) and selected LSC port
- stimulus: single frequency sinusoidal drive to last IO mirror angle in tilt and twist
  - $10^{-12} – 10^{-7}$ rad r.m.s.
  - 1 Hz - 1 kHz
- response: each LSC sensing port
- analysis: study of strain spectrum; search for 1ω and 2ω
- visualization: log-log transfer function, log-log contribution to strain spectrum, coherence or other measure of confidence
- recurrence: once per month or per change in hardware
- duration:60 sec unavailability
- contingent tests:

C.1.6 Intermodulation Products due to Offsets and Large Amplitude Deviations

- technique: Introduction of intentional offset from the dark fringe (or resonance condition for cavities; addition of signals; analysis of output to determine linearity of the system
- stimulus: offset from nominal operating point of optical system
  - static offset from 0.1x to 10x required operational offset, OR sinusoidal modulation, 0.1x to 10x required operational offset, 0.1 Hz - 100 Hz
  - modulation in laser frequency 10^{-7} -0.01 Hz/$\sqrt{\mathrm{Hz}}$, 1 Hz to 10 KHz
  - modulation in lengths (individual or CM/DM basis) $10^{-20} – 10^{-15}$ m r.m.s., 1 Hz - 1 kHz
- response: all PSL/IO/LSC ports (strain, MI, RC, Frequency)
- analysis: study of sum and difference frequencies as function of variables; extraction of peak heights
- visualization: plot of distortion vs. parameters, fit to curve; projection of net effect on quiet strain spectrum
- recurrence: once per month or per change in hardware
- duration: 10 min
- contingent tests:

C.1.7 Phase Noise Limits due to Scattering in the Beam Tube

- technique: transfer function from modulation from beam tube intentional excitation to LSC length sensing ports, followed by cross power spectrum measurement of ambient motion fluctuations and selected LSC port
- stimulus: PEM cart shaker attached to beam tube at point of interest
  - 1 Hz - 30 Hz for up-conversion effects
  - 30 Hz - 1 kHz for direct effects
- response: PEM BT accelerometers, LSC length sensing ports
- analysis: differences of strain spectra with and without excitation for the 1-30 Hz band; signals at excitation frequency for 30 Hz - 1 kHz

- visualization: power in strain spectrum due to BT natural motion
- recurrence: once per month or per change in hardware
- duration:10 min of unavailability; 1 day setup for placement of PEM shaker etc.
- contingent tests:

C.2 Optimization of Optical Phase Sensitivity

C.2.1 Signal-to-Noise Optimization of the RF Modulation Index

- technique: variation of the RF modulation depths to search for best performance in shot noise limited region of the spectrum
- stimulus:
  - fixed-frequency calibration peak (strain modulation), 10 Hz - 10 kHz, S/N 10 - 100 in sample long enough to get 1% uncertainty in shot-noise region
  - variation (stepped in 1% to 10% increments with stationary points for strain measurement) of modulation depth, 0.1 nominal to 2x nominal (may be limited by interferometer servos to a smaller range). An intelligent search for a maximum cuts duration.
  - 1 Hz - 10 kHz
- response: measure of peak height of calibration in strain spectrum as a function of modulation depth; measure of shot-noise limited performance in regions without interfering features (peaks, etc.)
- analysis: fit to shot-noise limited level, fit to peak height, search for optimum
- visualization: strain spectrum for each step; plot of S/N vs. modulation depth with peak indicated
- recurrence: once per week or per change in hardware
- duration:60 sec
- contingent tests:

C.2.2 Mode Matching into Interferometer

- technique: stepping of input mode parameters, measurement of circulating power in interferometer as measure of matching
- stimulus: stepping of the translation stage which varies the IO telescope mirrors, thus changing the matching. Magnitude of motion: TBD, according to initial state of matching. Each step will require reinjection of the beam, realignment, and interferometer tuning, and possibly a wait for thermal equilibrium
- response:
  - circulating power in the arm cavities (recycling cavity high-order modes are easily excited?), as monitored by photodiodes looking at ETM transmitted beam
  - S/N in shot noise limited region
- alternative response: analysis of CCD camera looking at RM for first circularly symmetric mode; amplitude of mode as function of matching will allow faster convergence maybe
- analysis: fit to curve of power and S/N as function of matching, identification of desired matching
- visualization: plot of power and S/N as a function of matching with optimum point identified
- recurrence: once per month or per change in hardware

- duration:1 hour unavailability, TBD on translation and realignment realities for IO telescope
- contingent tests:

C.2.3 Mirror Absorption through Change in Mode Matching

- technique: Varying input power to the interferometer; looking at signals characteristic of mode matching to see the changes; use of a model to regress back to the change in mirror curvature and thus absorption
- stimulus: laser power stepped in factors of three, from 60 mW to 6 W net input power at RC
- response: light reflected from RC captured with a CCD and stored in frames for post-analysis
- analysis: calculation of the amount of TEM_{01} as a function of input power; from this, calculation of an average of absorption in TMs.
- visualization: TEM_{01} as a function of input power; absorption number
- recurrence: once per month or per change in hardware
- duration:10 minutes unavailability
- contingent tests:

C.2.4 Higher Order Arm Cavity Mode Scan

- technique: with other cavities misaligned, drive the ETM of the cavity of interest at uniform velocity through several λ. Measure the intensity vs. time on the ETM transmitted light.
- stimulus: 0.1 Hz ramp to the z input of the ETM suspension controller. Best if this is done with a closed loop control using the suspension sensors to reduce variations due to seismic input.
- response: photodiode measurement ETM transmitted light
- analysis: Plot of intensity vs. position, normalized to give free spectral range. Superposition of multiple traces to show variability; averaging to improve estimate. Identification of cavity modes by comparison with cavity model; analysis of the nature of the mismatch (simple misalignment suppressed) by comparison with the model
- visualization: Classic ‘optical spectrum analyzer’ (intensity vs. cavity length modulo $\lambda/2$)
- recurrence: once per month or per change in hardware
- duration:10 min unavailability
- contingent tests:

C.2.5 Arm Cavity Loss Measurement by Reflection

- technique: with other cavities misaligned, the cavity of interest is locked and then unlocked. The difference in the reflected light allows an inference of the losses, once matching (test C.2.4) is known.
- stimulus: locking and unlocking of a single arm cavity. This might be performed using a mechanical dither of the ETM at ~100 Hz with a synchronous demodulation of the ETM transmitted light; or a subset of LSC RF locking electronics.
- response: photodiode measurement of the reflected light PO from the ITM
- analysis: application of the Fabry-Perot formula to infer losses based on known TM properties (transmissions) and light reflected.
- visualization: numbers on a screen
- recurrence: once per month or per change in hardware

- duration:10 min unavailability
- contingent tests:

C.2.6 Arm Cavity Loss Measurement by Ring-down

- technique: With other optical cavities misaligned, the cavity under test is locked. Small step changes in the input light intensity are made, and the exponential drop-off in stored power is measured.
- stimulus: 1-10% step modulation of the laser intensity (<1 μsec rise/fall times), 1-30 Hz rep rate
- response: photodiode measurement of the ETM transmitted light
- analysis: fit to exponential decay/rise in the ETM transmitted light, with synchronous averaging of the data. Inference of storage time, comparison with design storage time and previous measurements. Inference of additional loss.
- visualization: storage time and loss vs. time, vs. integrated light intensity; fits to models for decay as a function of time, products of time and intensity, etc.
- recurrence: once per week or per change in hardware
- duration:10 min non-availability
- contingent tests:

C.2.7 Recycling Cavity Loss Measurement

- technique: ETMs misaligned, Michelson locked, recycling cavity locked. Small step changes in the input light intensity are made, and the exponential drop-off in stored power is measured.
- stimulus: 1-10% step modulation of the laser intensity (<1 μsec rise/fall times), 1-30 Hz rep rate
- response: photodiode measurement of the BS PO light intensity
- analysis: fit to exponential decay/rise in the PO intensity, with synchronous averaging of the data. Inference of storage time, comparison with design storage time and previous measurements. Inference of additional loss.
- visualization: storage time and loss vs. time, vs. integrated light intensity; fits to models for decay as a function of time, products of time and intensity, etc.
- recurrence: once per month or per change in hardware
- duration:10 min non-availability

C.2.8 ETM Suspension Actuator Axial Calibration

- technique: Form a Michelson interferometer using an auxiliary laser, with one arm using the ETM surface (as viewed through the optical relay system to bring out the ETM transmitted light) and the other arm a reference arm. All components except for the ETM COS components are outside of the vacuum. A slow sweep of the ETM is made to calibrate the fringe height. Then a small modulation of the mirror position is made and the r.m.s. motion inferred from the auxiliary interferometer output.
- stimulus:
  - For auxiliary interferometer calibration: 0.1 Hz ramp to the z input of the ETM suspension controller. Best if this is done with a closed loop control using the suspension sensors to reduce variations due to seismic input.

  - For suspension actuator calibration: fixed-frequency and sweep sinusoidal excitation of the z input to the suspension actuator at levels corresponding to $\sim 10^{-10}$ to 10^{-15} m r.m.s., 30 Hz - 1 kHz.
- response: Auxiliary Michelson antisymmetric photodiode output
- analysis:
  - measurement of the peak-peak output of the auxiliary interferometer while sweeping ETM. From this, the 'volts/λ' for the interferometer is measured. Fits made to peak and peak, multiple averages taken
  - measurement of the coil currents (at suspension controller test points) and the auxiliary interferometer photodiode output locked at the dark fringe or mid-fringe.
- visualization:
  - sweep of auxiliary interferometer vs. ETM position
  - motion of the ETM (as measured by the calibrated auxiliary interferometer) per ampere of suspension controller current, as a function of frequency (transfer function). Coherence.
- recurrence: once per month or per change in hardware
- duration:10 min unavailability
- contingent tests:

C.2.9 Suspension Actuator Angular Calibration

- technique: use of the optical levers to measure angular motion effected by the suspension actuators. Optical levers are calibrated by making known translations of the quadrant photodiodes and measuring their response in a preliminary measurement.
- stimulus: sinusoidal drive to each COC element suspension controller θ and ϕ input
  - $10^{-7} - 10^{4}$ rad
  - 1 Hz - 1 kHz
- response: calibrated optical lever output.
- analysis: form the transfer function from angle commanded to angle resulting as determined from the optical lever.
- visualization: the transfer function; coherence.
- recurrence: once per month or per change in hardware
- duration:60 sec
- contingent tests:

C.2.10Length Control System Diagonalization and Diagnostics

- technique: Using calibrated suspension actuators, make equal in-phase or counter-phase axial motions of the two ETMs, or two ITMs, or RC. Observe the signals at the LSC and ASC sensors. The interferometer is in the operational state during the measurement.
- stimulus: sinusoidal or pseudo-random excitation to the suspension controllers
  - 0.1 - 100x the ambient control force in the locked case.
  - 1 Hz - 10 kHz
- response: LSC and ASC outputs
- analysis:
  - standard transfer function tools; also, fixed-frequency transfer function vs. other interferometer parameters (like beam centering, suspension controller balance, offsets

  from nominal alignment, etc.)
  - formation of the complete matrix of motions to signals for the LSC
  - formation of the complete matrix of accidental coupling to the ASC
- visualization: standard transfer function tools; matrix
- recurrence: once per month or per change in hardware
- duration:10 min unavailability
- contingent tests:

C.2.11Angle Control System Diagonalization and Diagnostics

- technique: Using calibrated suspension actuators, make equal in-phase or counter-phase angular motions of the two ETMs, or two ITMs, or RC. Observe the signals at the LSC and ASC sensors. The interferometer is in the operational state during the measurement.
- stimulus: sinusoidal or pseudo-random excitation to the suspension controllers
  - 0.1 - 100x the ambient control force in the locked case.
  - 1 Hz - 10 kHz
- response: LSC and ASC outputs
- analysis:
  - standard transfer function tools; also, fixed-frequency transfer function vs. other interferometer parameters (like beam centering, suspension controller balance, offsets from nominal alignment, etc.)
  - formation of the complete matrix of motions to signals for the ASC
  - formation of the complete matrix of accidental coupling to the LSC
- visualization: standard transfer function tools; matrix
- recurrence: once per month or per change in hardware
- duration: 10 min unavailability
- contingent tests:

C.3 Noise due to Random Forces

C.3.1 Suspended Optical Component Seismic Noise Sensitivity

- technique: measurement of the effect on the strain output for both monitored ambient seismic excitation, and with applied excitation
- stimulus: PEM shaker applied to the seismic isolation support beams, in x, y, z; excitation waveform may be sinusoidal, swept sine, impulse, or pseudorandom
- response: PEM accelerometers, strain output
- analysis: transfer functions and ratios of power spectra of strain output over accelerometer output
- visualization: transfer functions
- recurrence: once per month or per change in hardware
- duration:10 min (attaching shaker, shaking, removal of shaker) per mass
- contingent tests:

C.3.2 Suspended Optical Component Acoustic Noise Sensitivity

- technique: measurement of the effect on the strain output for both monitored ambient acoustic excitation, and with applied excitation
- stimulus: PEM loudspeaker placed in the vicinity of VE containing COC element of interest; excitation waveform may be sinusoidal, swept sine, impulse, or pseudorandom
- response: PEM microphones, strain output
- analysis: transfer functions and ratios of power spectra of strain output over microphone output
- visualization: transfer functions
- recurrence: once per month or per change in hardware
- duration:60 sec per test, one test per mass
- contingent tests:

C.3.3 Suspended Optical Component Magnetic Field Sensitivity

- technique: measurement of the effect on the strain output for both monitored ambient magnetic field excitation, and with applied excitation
- stimulus: PEM magnetic field generator placed in vicinity of VE containing COC element of interest; excitation waveform may be sinusoidal, swept sine, impulse, or pseudorandom
- response: PEM magnetometer, strain output
- analysis: transfer functions and ratios of power spectra of strain output over magnetometer output
- visualization: transfer functions
- recurrence: once per month or per change in hardware
- duration:60 sec per test, one test per mass
- contingent tests:

C.3.4 Suspended Optical Component Electric Field Sensitivity

TBD.

C.3.5 Suspended Optical Component Tilt Sensitivity

TBD.

C.3.6 Pendulum Longitudinal Mode Q

TBD.

C.3.7 Pendulum Wire Transverse Mode Q

- technique: excitation of the wire ‘violin string’ resonances using the suspension actuators; measurement of the peak in the strain spectrum, ring-down time.
- stimulus: suspension actuators axial input. Sine excitation at the fundamental and harmonics of the suspension wires; cutoff of excitation once at 10-100x the ambient motion to observe ring-down.
- response: strain output

- analysis: fits to exponential decays
- visualization: transfer functions; Q vs. time (periodic tests over months to observe changes)
- recurrence: once per month or per change in hardware
- duration:10 min per resonance; ~2 resonances per wire; 2 wires per mass; 4 TMs
- contingent tests:

C.3.8 Pendulum Wire Longitudinal Mode Q

- technique: excitation of the vertical motion of the SEI support beams at or around the resonant frequency of vertical motion (~11 Hz); measurement of the transfer function or of the ring-down time of the motion in the strain or control spectrum (coupling due to at least the earth's curvature) or the suspension sensors (nominally zero response, but certainly visible).
- stimulus: PEM shakers placed on the four support beam support points of the VE containing COC element of interest; excitation waveform may be sinusoidal at or swept sine around the (known) resonant frequency
- response: PEM accelerometer, strain output or other point in the LSC control system sensitive at ~11 Hz
- analysis: transfer functions and ratios of power spectra of strain output over accelerometer output; fits to exponential decays (if a ring-down test is made) or fits to line widths in the transfer function (if a sweep around the resonance is used to determine its width and thus losses)
- visualization: transfer functions; Q vs. time (periodic tests over months to observe changes)
- recurrence: once per month or per change in hardware
- duration:10 min
- contingent tests:

C.3.9 Pendulum Vertical to Horizontal Cross Coupling

TBD.

C.4 Optimization to Minimize Noise from Random Forces

C.4.1 Search for Rotation Insensitive Beam Position on Suspended Component

- technique: modulation in angle of the suspended optic; observation of the signal in the strain spectrum; motions of the beam and/or SEI actuators to observe size and sign of change of coupling; inference of point of no response, and placement of the beam/SEI at that point. Commands to the SEI coarse actuator are less desirable, as it will probably cause loss of lock in the interferometer. After locating the best point, SEI actuators may be used to re-center the optic.
- stimulus:
  - angular input to the SUS controller, in θ, ϕ; excitation waveform may be sinusoidal, swept sine, impulse, or pseudorandom
  - changes in the beam position on the mirror by commands to global ASC to change the optical axis affecting the beam on that mirror

- response: strain output
- analysis: transfer functions and ratios of power spectra of strain output over angular
- visualization: transfer functions
- recurrence: once per month or per change in hardware
- duration:60 sec per test, one test per mass
- contingent tests:

C.4.2 Search for Astatic Point in Suspended Component Position Controller

TBD.

C.5 Tests of the Facility-Detector Interface

C.5.1 Correlation of Residual Gas Pressure Fluctuations with Detector Output

- technique: RGA real-time measurements are correlated with some LSC signals to determine if pressure fluctuations are influencing the detector performance
- stimulus: none, or possible use of remote-control-valved leak in BT or VE
- response: PEM RGAs in BT and VE, LSC sensing signals
- analysis: correlation; averaging of time series; statistics of gas pulses; comparison with calculated waveforms
- visualization: correlation function; time series of pressure and time series from detector; histograms
- recurrence: once per month or per change in hardware
- duration: if using gas pulse, 10 min; otherwise, non-invasive
- contingent tests:

C.5.2 Correlation of Technical Power Fluctuations with Detector Output

- technique: Power monitor real-time output is correlated with LSC signals
- stimulus: none
- response: PEM power monitor, LSC sensing signals
- analysis: correlations; statistics of power fluctuations
- visualization: correlations; histograms
- recurrence: once per month or per change in hardware
- duration: non-invasive
- contingent tests:

C.5.3 Correlation of Facility Power Fluctuations with Detector Output

- technique: Power monitor real-time output is correlated with LSC signals
- stimulus: none

- response: PEM power monitor, LSC sensing signals
- analysis: correlations; statistics of power fluctuations
- visualization: correlations; histograms
- recurrence: once per month or per change in hardware
- duration: non-invasive
- contingent tests:

C.5.4 Correlation of Facility Monitors with Detector Output

- technique: Each facility monitor with real-time output is correlated with the LSC and ASC signals to find the influence on the interferometer performance
- stimulus: none; or Facility equipment can be power cycled, change in operating parameters (speed, temperature, etc.)
- response: Facility status flags and sensors, LSC/ASC sensors
- analysis: correlations; statistics; comparison of waveshapes, harmonic structure
- visualization: correlation, histograms, time series
- recurrence: once per month or per change in hardware
- duration: non-invasive
- contingent tests: