- lessons from -
LIGO End to End model

- How to make the modeling efforts successful
  - advertisement “why modeling”
- How to make the modeling software successful
  - good objected oriented design
- LIGO End to End simulation package
  - structure and usage

Second LISA Science/Engineering Workshop

October 17, 2001   Pasadena CA

Hiro Yamamoto / LIGO Lab Caltech

LIGO End to End model

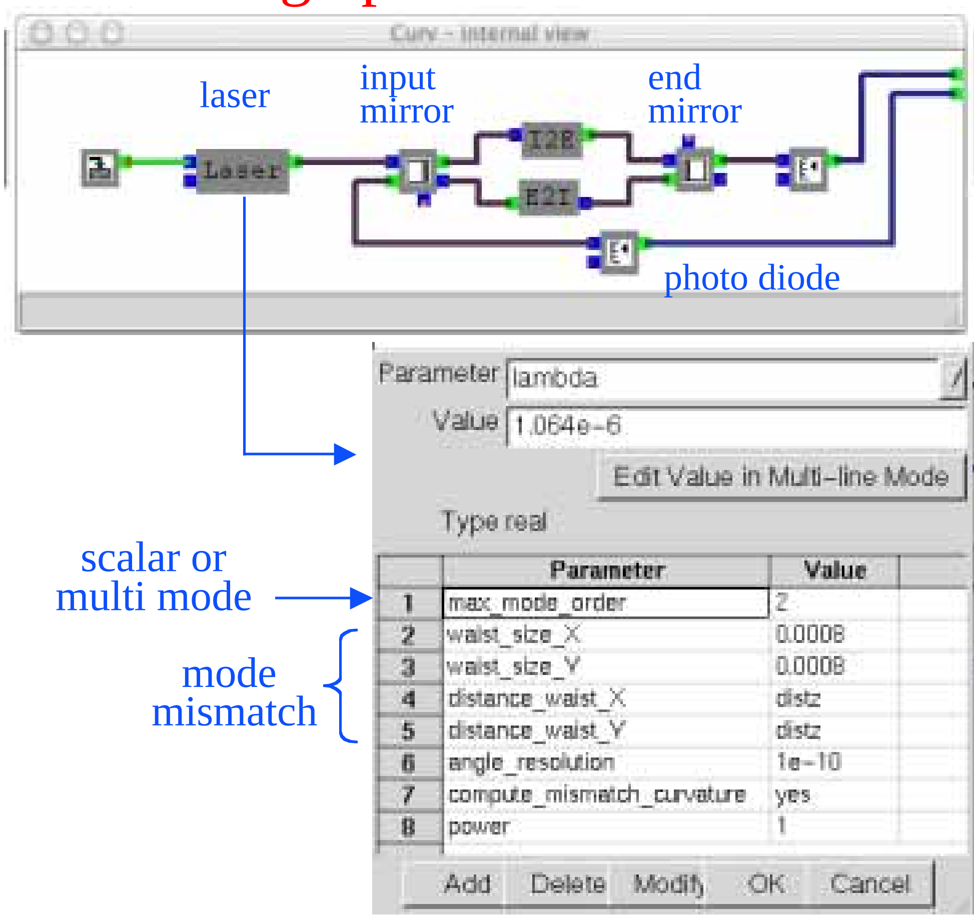
what is it

- General purpose Gravitational Wave Interferometer detector simulation program
  - » Modeling tool like matlab or mathematica
  - » More specifically optimized for interferometer simulation
  - » Easy to simulate a wide range of optical and mechanical configurations without modifying and revalidating codes
- Simulation program
  - » Time domain simulation program written in C++
  - » Physics simulation for LIGO
    - – Optics, mechanics, electronics…
    - – Major noise sources, signals (to be implemented)
  - » Easy to add new phenomena by concentrating on physics, not on programming

Simulation setup

graphical user interface

graphical user interface



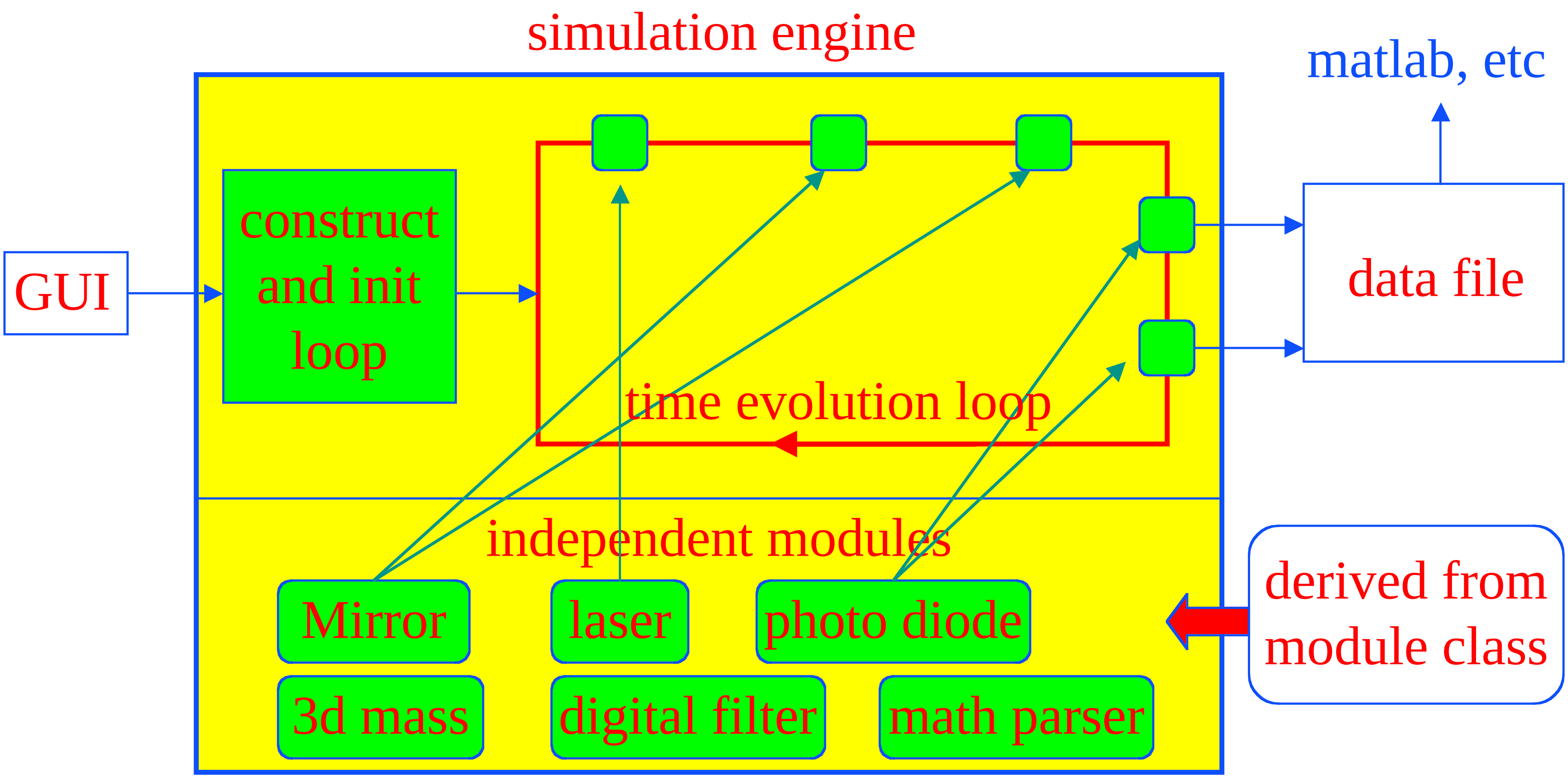
GUI output

```
Add_Submodules
{
  field_gen Laser;
  mirror2 ITM;  mirror2 ETM;
  propagator I2E;  propagator E2I;
  data_in Power; ...
}
Settings Laser
{
  max_mode_order = 2
  distance_waist_X = distz
  compute_mismatch_curvature = yes; ...
}
Settings I2E { length = CavLength }
Settings E2I { length = CavLength }
Settings Power {init = 1; type = vector_real}
...
Add_Connections
{
  Power 0 -> Laser power; Laser 0 -> ITM Bin;
  ITM Aout -> I2E 0;   I2E 0 -> ETM Ain
  ETM Aout -> E2I 0;   E2I 0 -> ITM Ain
  ETM Bout -> TransPower 0;
  TransPower 0 -> this Tran0 ...
}
```

Simulation engine

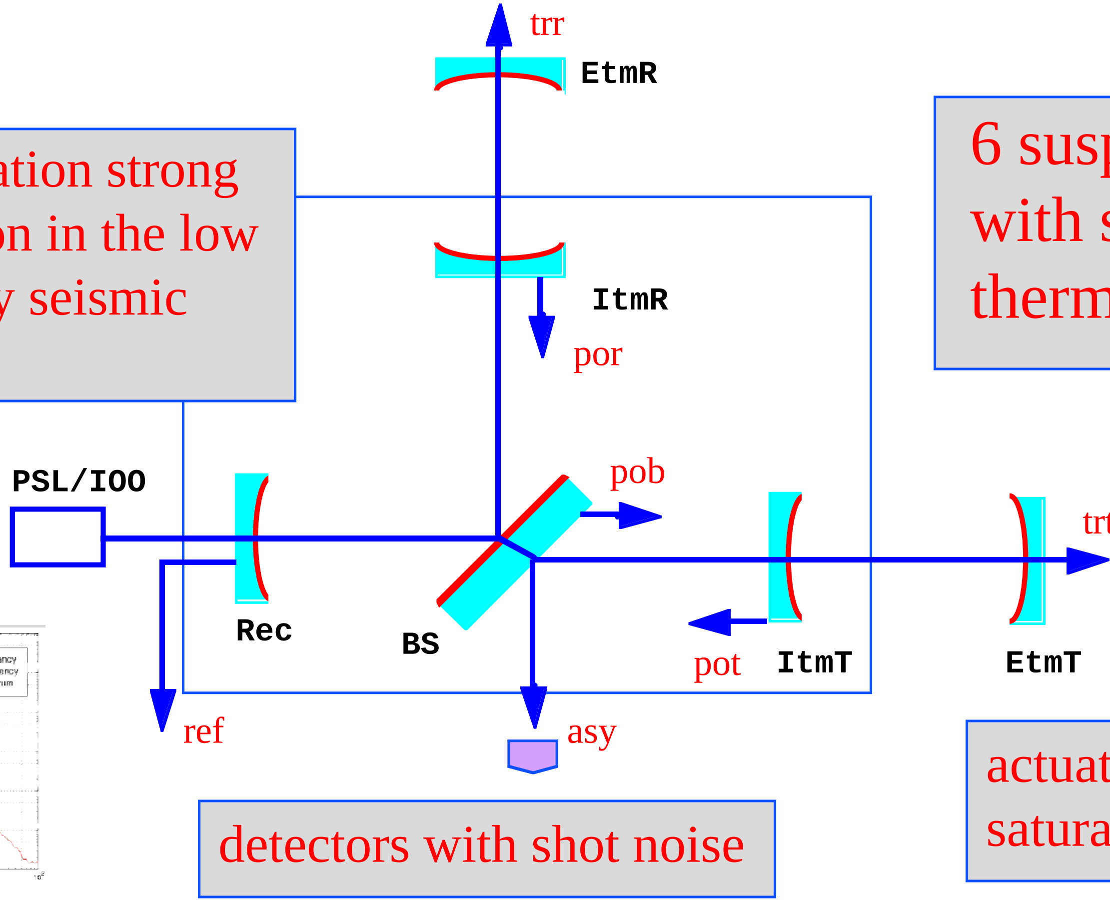
developed by M.Evans

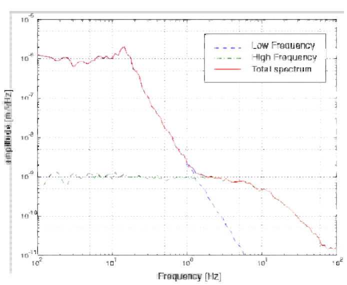
time loop and independent modules



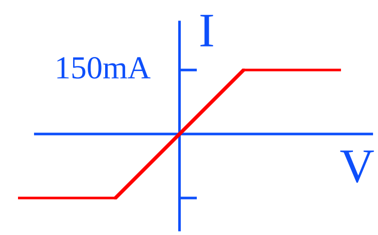
LIGO simulation setup

consists of 1000 modules

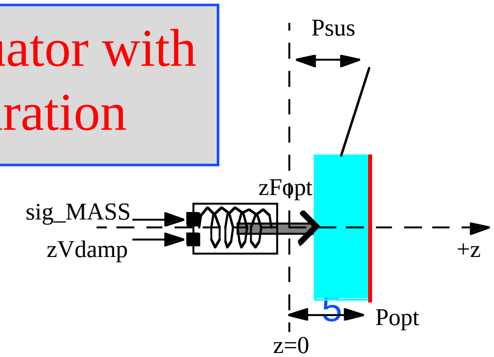




6 suspended mirrors with seismic and thermal noise

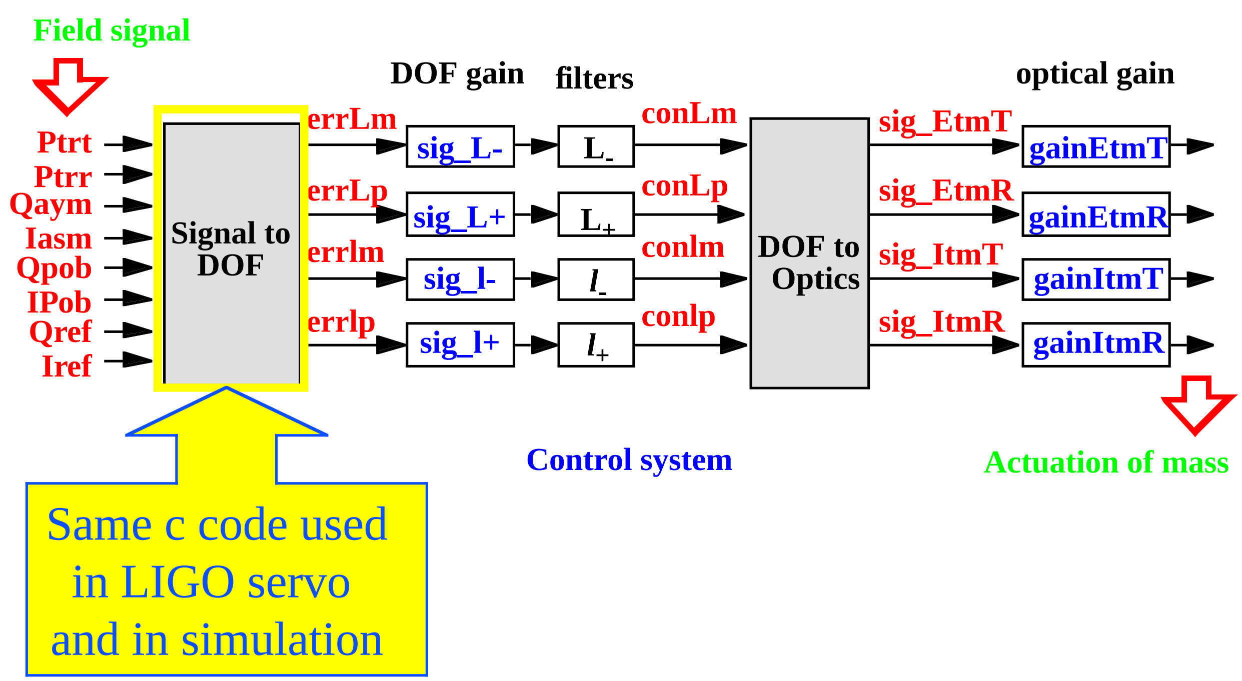


actuator with saturation



Automated Control Matrix System

LIGO document T000105 by Matt Evans



Lock acquisition

real and simulated

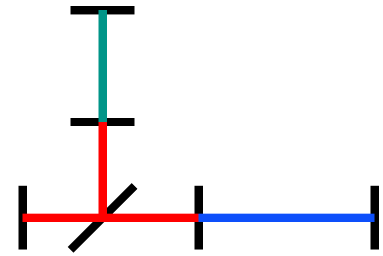
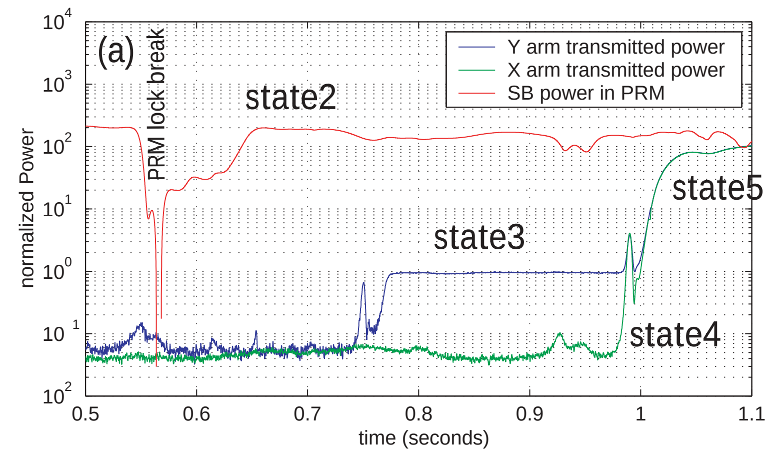
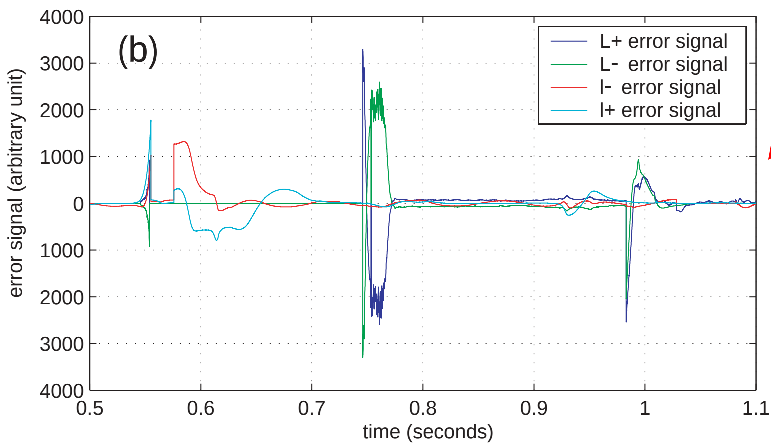


Figure 1. LHO 2k IFO data

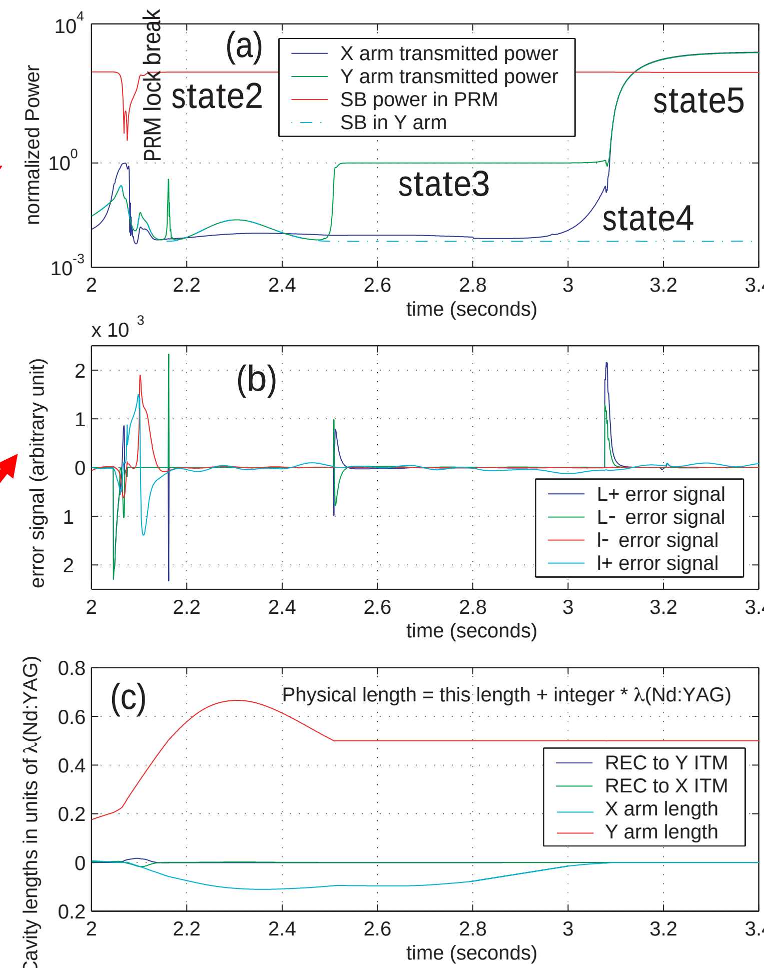


Arm powers are normalized by the power when one arm is locked.
SB power is normalized by the input SB power.



observable

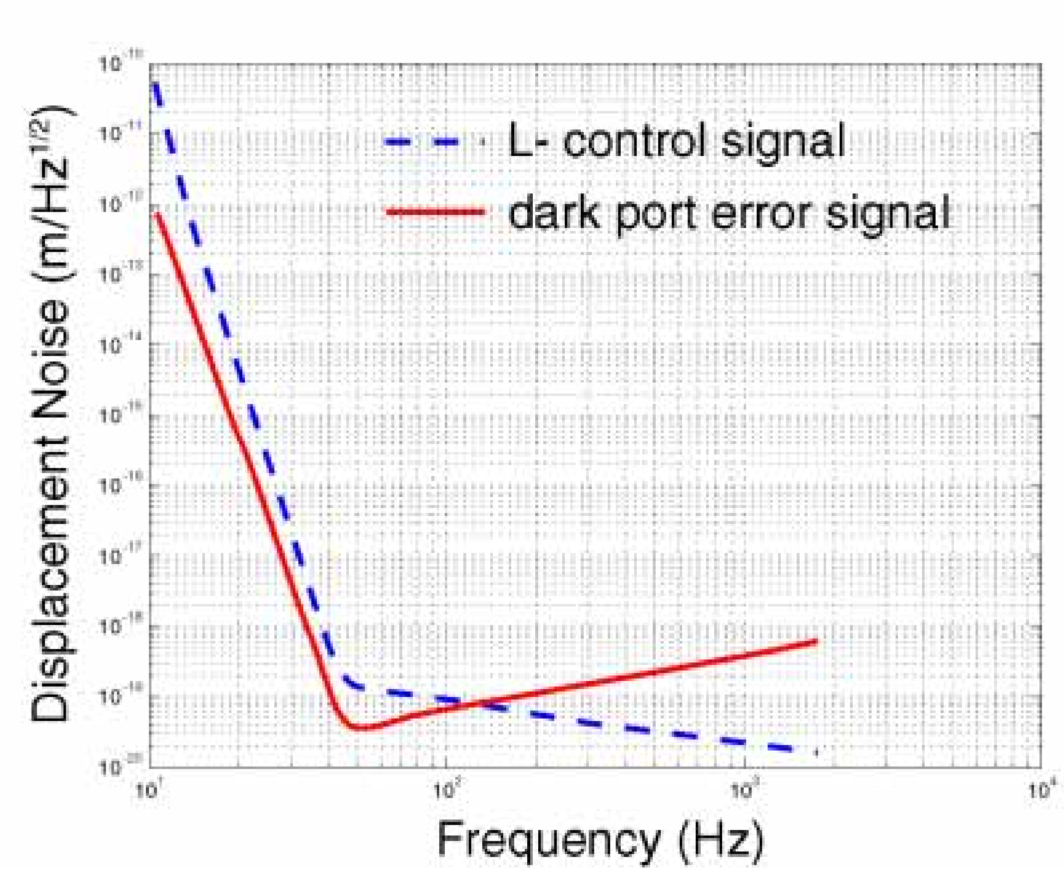
Figure 2. Simulated signal



Not experimentally observable

Simulated sensitivity curve

baseline fundamental noise by realistic simulation



Simulation can include

- Interferometer
- Mechanics
- Sensor-actuator
- Servo electronics
- Signal extraction
- Noises
  - Mechanical
  - Sensor
  - Laser
  - Mode Cleaner
  - Electronics

Simulation of subsystems with different time scales

- LIGO Core optics components ~ 10 μ seconds
  - » simulation time step is determined by this
- subsystems with different time scales
  - » Michelson cavity ~ 0.01 μ seconds
    - – module using adiabatic approximation, originally formulated by Dave Redding
  - » Pre Stabilized Laser, Mode Cleaner ~ 0.1 μ seconds
    - – parameterize the subsystems when used to study the core optics response
  - » Mechanical structure ~ 100 μ seconds
    - – use smooth extrapolation, in stead of simulating at each time step
- When characteristics of each subsystem is studied, run at its own time step.

Different cultures in HEP and LIGO

High Energy Physics	LIGO
people know of simulation	very few people know of simulation
simulation and hardware developments go side-by-side, stimulating each other	simulation is used only when there is no other way, and a model is developed only for that problem
simulation is used to understand the integrated system	overall performance is maintained by noise budgets assigned to each subsystem
hardware people intend to use simulation to understand problems	hardware people try to address problems by dealing with hardware
use of simulation for data analysis is a well known procedure	simulation for data analysis is non existent

Advertisement to whole community

- what is the modeling / simulation of the system
- what it is for, what it can do and cannot do
- how it can be used, and why it is useful
- establish common sense
- identify the scope and requirement
- identify customers
- allocate enough resource
- stimulate the communication between hardware and software groups

make simulation ready for hardware design and development

- make the minimum code ready asap so that subsystem modeling can start
- simulate subsystems using the same program for the integrated system
- minimize duplicated efforts
  - » avoid making “only for this problem” special model
- maximize sharing of codes in simulation and in real device
- hardware people will learn what the simulation is and where it can be used
- model will grow up by inputs from hardware group

Modeling facility for wider community

- modular design of the software
  - » main program + completely independent modules
  - » design module API to hide the detail structure of the main program
  - » concentrate on physics, not programming
  - » independent development and validation of new modules and subsystems
  - » easy integration of all modules on larger and fast computers
- good user interface to understand the configuration
  - » easy adoption by new users
  - » easy adoption of codes developed by other group
  - » easy maintenance and development
- LIGO Data Analysis System (LDAS)
  - » analyzing huge data
  - » server-client model

Code design

- object oriented framework
  - » need time and talent to design a very good framework
  - » easy to expand, integrate and maintain
  - » easy to develop in modular and incremental way
- flexibility is more important than speed
  - » easy to develop complex system
  - » easy to apply the model to wider problems by wider users
- generic modeling environment
  - » can model small subsystems to all integrated system
  - » easy customization by the end-user
- Good front-end to develop the simulation setup and to setup various settings and parameters