Simulation of LIGO Interferometers

Hiro Yamamoto
LIGO Laboratory / California Institute of Technology

1. End to End simulation
2. Lock acquisition
3. Noise simulation

LIGO-G020255-00-E

E2E overview

- General purpose GW interferometer simulation package
  - Original version designed and developed by M.Evans
  - Generic tool like matlab or mathematica
  - Easy to simulate a wide range of configuration without modifying and revalidating codes
- Simulation program
  - Time domain simulation written in C++
  - Optics, mechanics, servo ...
  - Easy to add new phenomena by concentrating on physics, not on programming

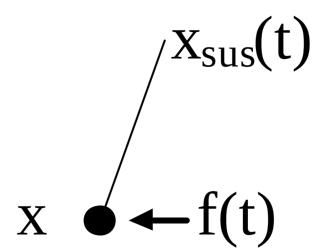
E2E perspective

- e2e development started after LIGO 1 design completed (1997 ~)
- LIGO 1 lock acquisition was redesigned successfully using e2e by M.Evans (2000 ~ 2001)
- Major on going efforts (2001 ~)
  - Realistic noise of the locked state interferometer
  - Effect of the thermal lensing on the lock acquisition
- Integration of the real LIGO length and alignment control system with the simulated interferometer
  - discussion just started
  - not likely to happen due to limited resources

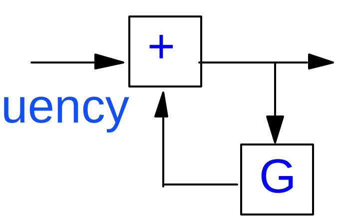
Time domain simulation

- Analog process is simulated by a discritized process with a very small time step (10^{-7}~ 10^{-3} s)
- Linear system response is handled using digital filter
  - » Transfer function -> digital filter
  - » Pendulum motion
  - » Analog electronics

$$x = \frac{1}{s^2 + gs + w_0^2}\left(\frac{f}{m} + w_0^2 x_{sus}\right)$$

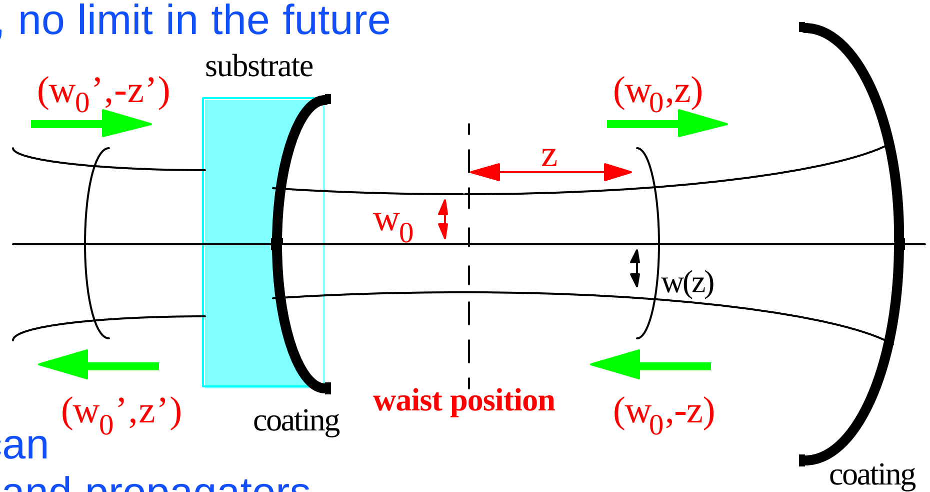


- Easy to include non linear effect
  - » Saturation, e.g.
- A loop should have a delay
  - » Explicit causality needs to be satisfied
  - » Time step is constrained by the unity gain frequency



Fields and optics

- Time domain modal model
  - » field is expanded using Hermite-Gaussian eigen states
  - » number of modes (n+m) <=4 for now, no limit in the future
- Reflection matrix
  - » tilt
  - » vertical shift
  - » curvature mismatch
  - » low mode phase map planned
- Completely modular
  - » Arbitrary planar optics configuration can be constructed by combining mirrors and propagators
- Photo diodes with arbitrary shapes can be attached anywhere
- Adiabatic calculation for short cavities for faster simulation



Mechanics simulation

(1) Seismic motion

» measurements of power spectrum and correlation

» generate time series for each stack which satisfy power spectrum and first order correlation

(2) Parameterized HYTEC stack

» 6x6 for HAM and BSC camber

(3) Simple single suspended mirror

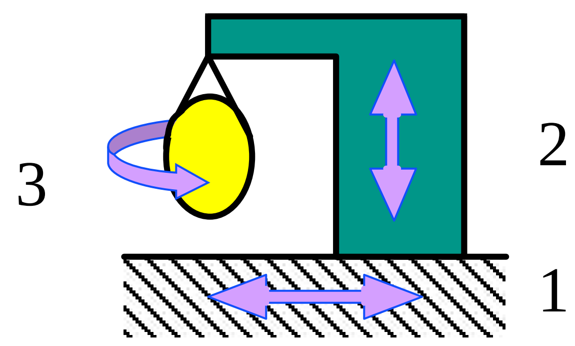
» simple 1dimension mass object

– used with scalar fields for fast simulation

» 3 dimension model with modal model

– 4 sensors and actuators

– couple between LSC and ASC

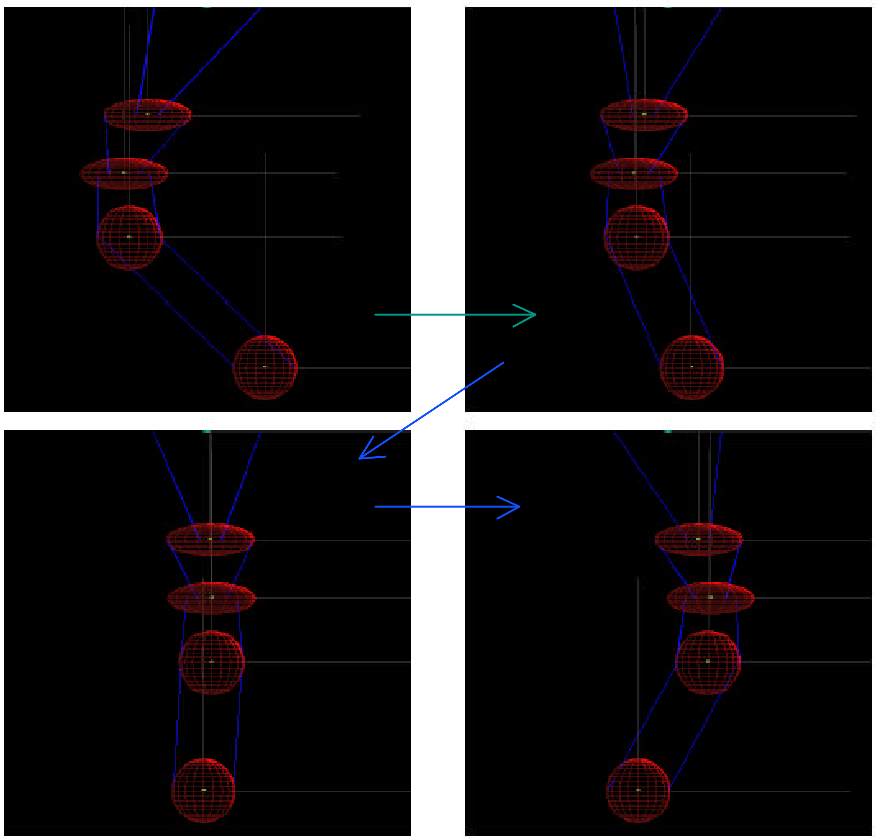


MSE :
Mechanical Simulation Engine

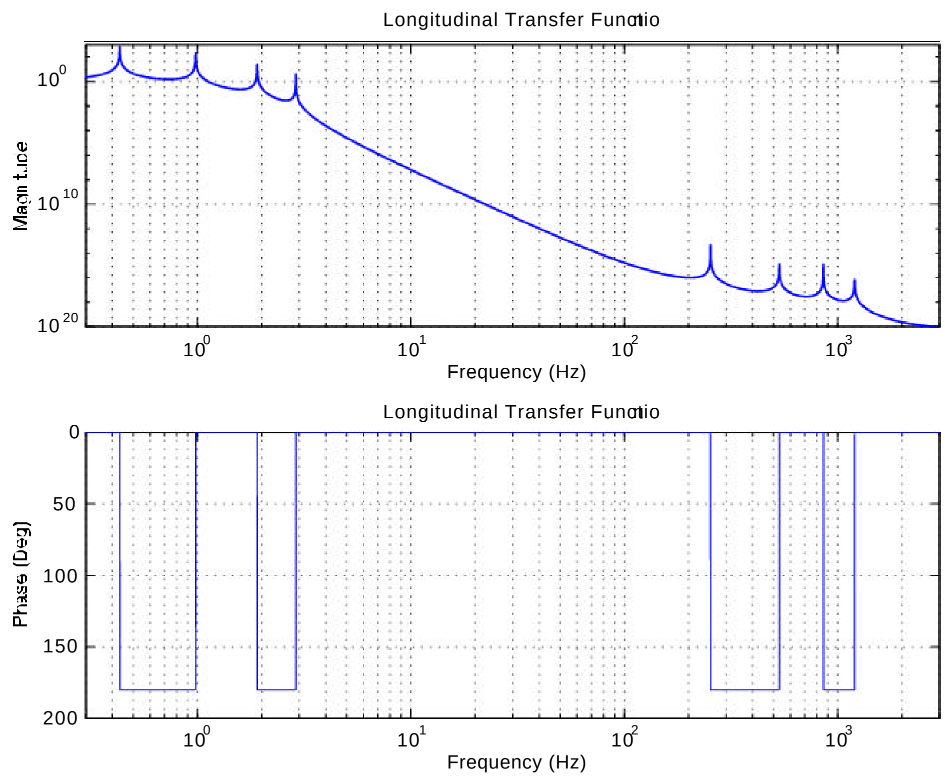
- C++ Library to simulate a fully three-dimensional mechanical system, developed by G. Cella
  - » modular environments
  - » automatic search for working points
  - » thermal noise and realistic damping simulation
  - » system asymmetries properly propagated
- Stand alone simulation package, with interface to e2e taken into account
  - » frequency and time domain
  - » build and debug a model and integrate to e2e by placing wrapper
  - » integration with other mechanical simulation
    - – For adv. LIGO, there are several sub-system level modeling efforts are already doing on, and MSE can interface to those models.

MSE example (V.Sannibale) Adv. LIGO suspension

motion visualization

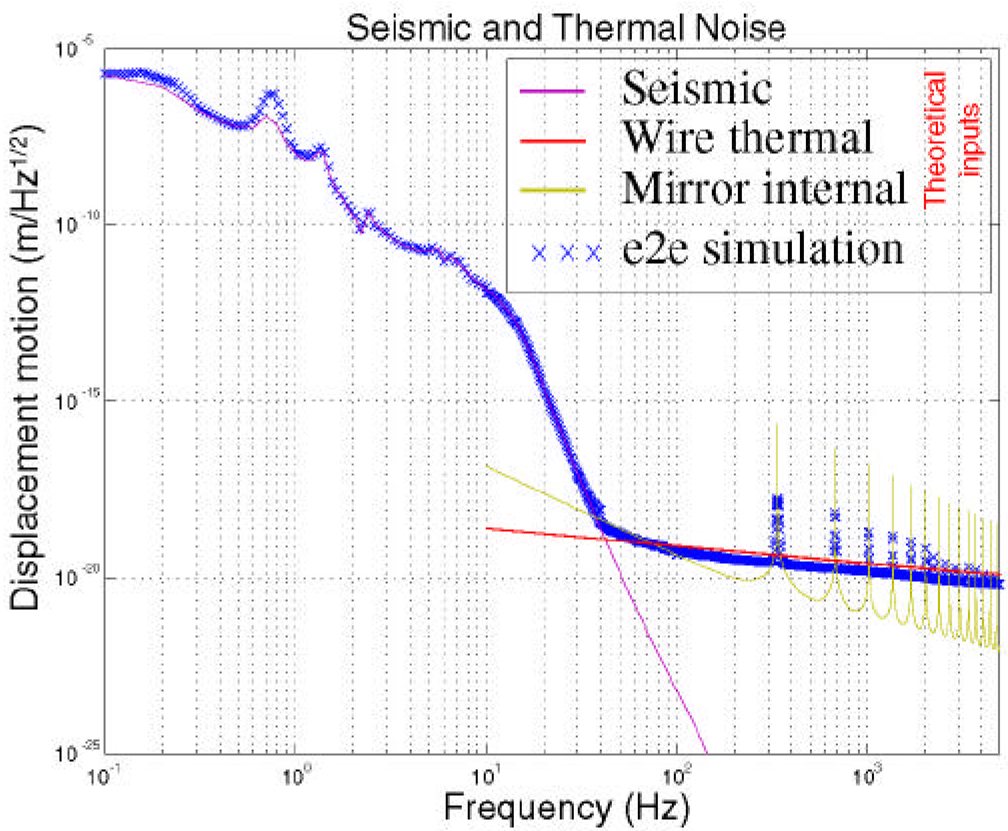


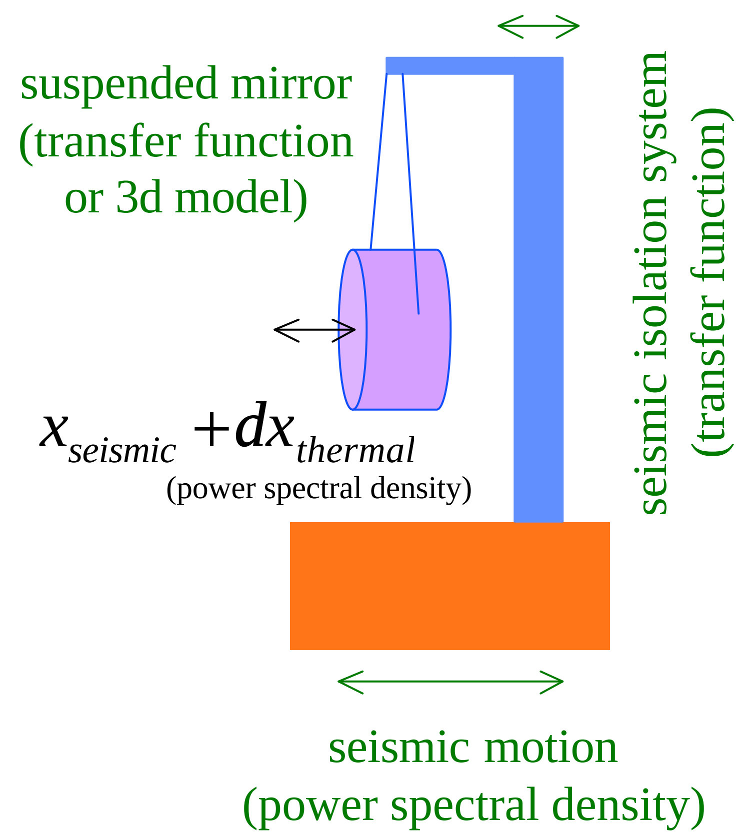
longitudinal transfer function



Primary noise - 1

seismic & thermal noise





Primary noise - 2

shot noise for an arbitrary input

Average number of photons

$$n_0(t) = \frac{h \cdot P(t) \cdot \Delta t}{h \cdot n}$$

Actual integer number of photons

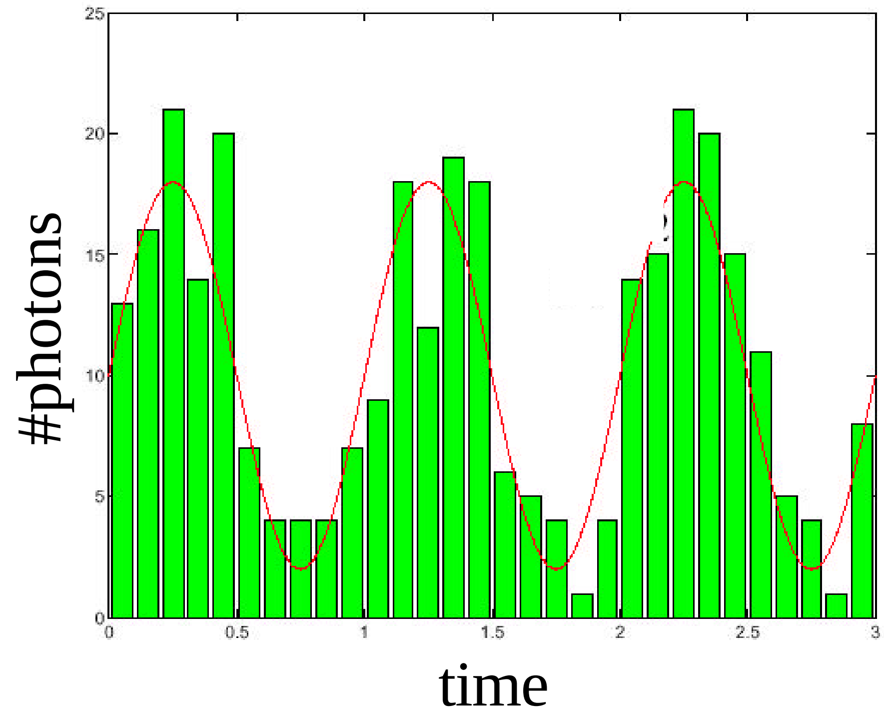
$$n(t) = Poisson(n_0(t))$$

Simulation option

Shot noise can be turned on or off for each photo diode separately.

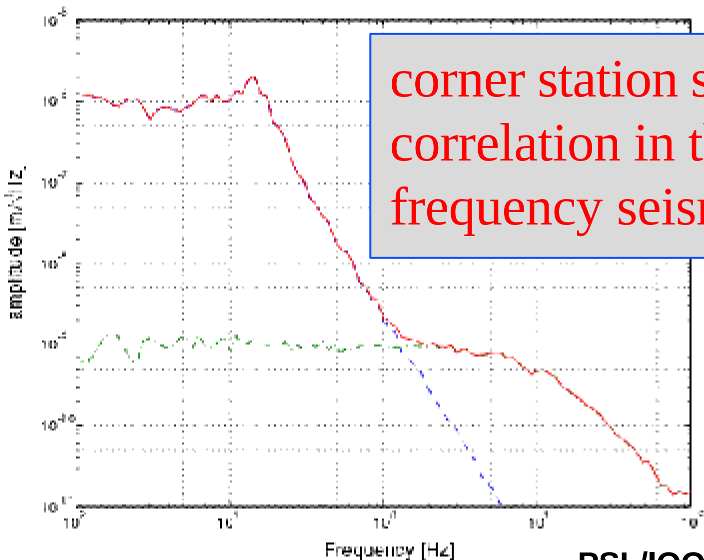
Average number of photons by the input power of arbitrary time dependence

Actual number of photons which the detector senses.



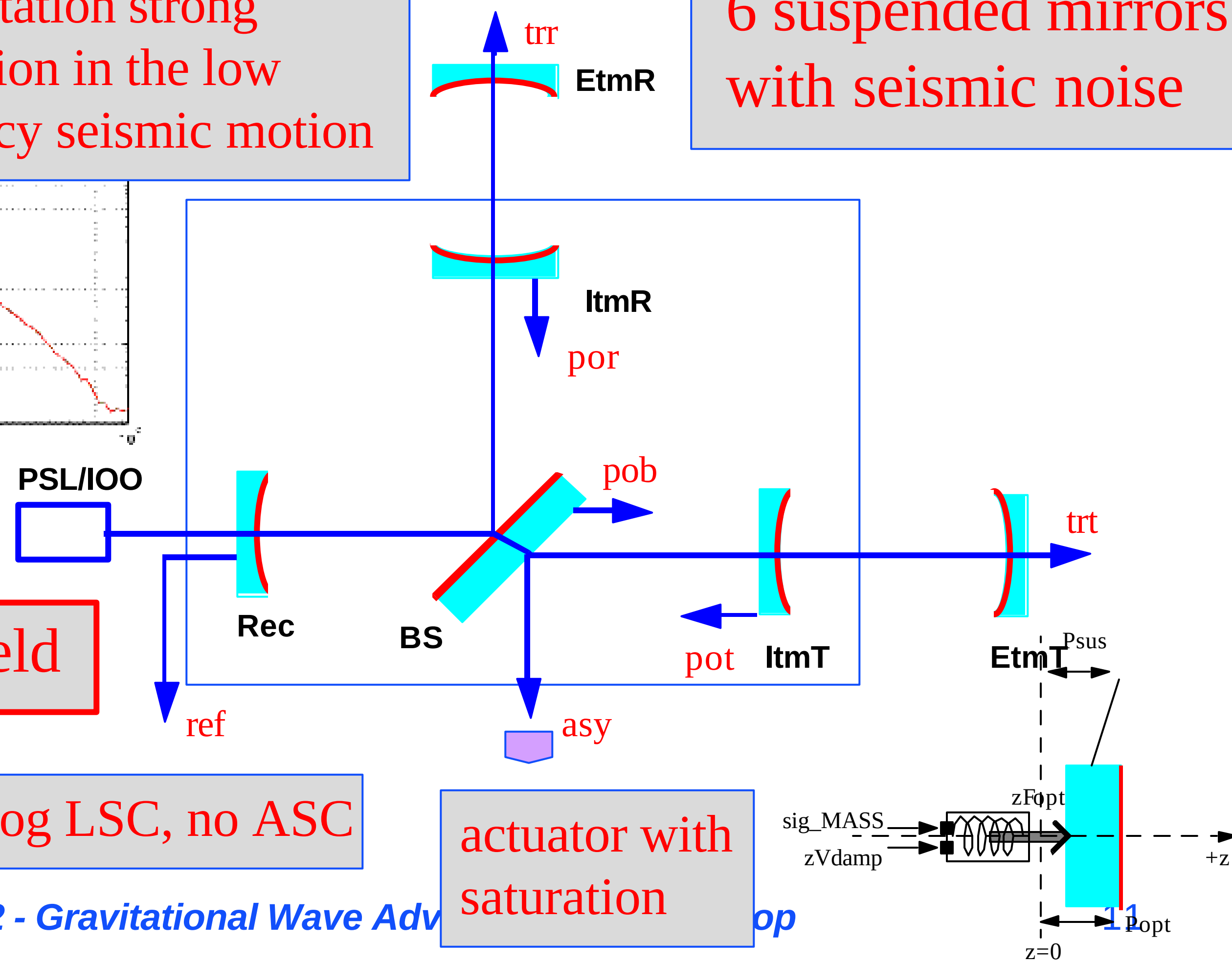
Lock acquisition design

Han2k



corner station strong correlation in the low frequency seismic motion

6 suspended mirrors with seismic noise



scalar field

analog LSC, no ASC

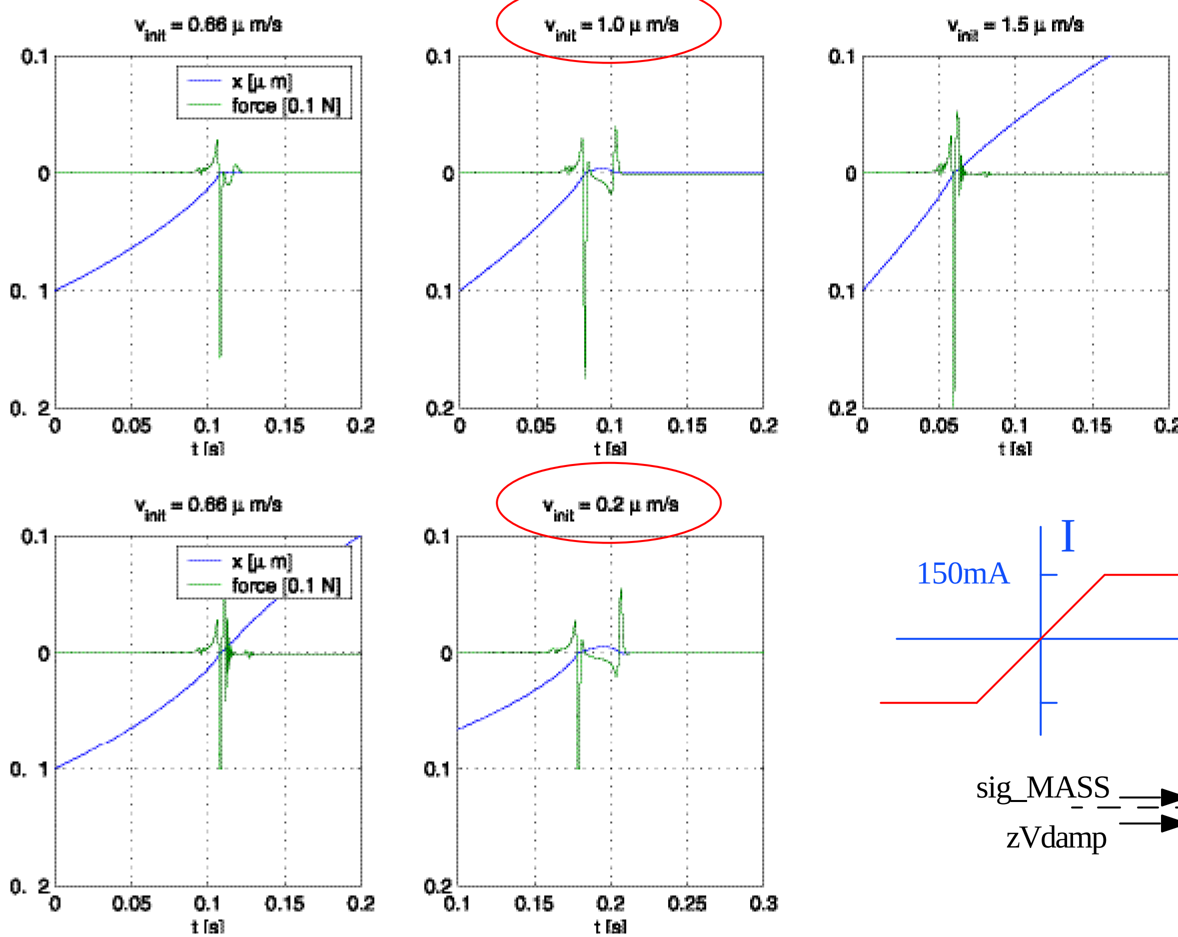
actuator with saturation

Fabry-Perot

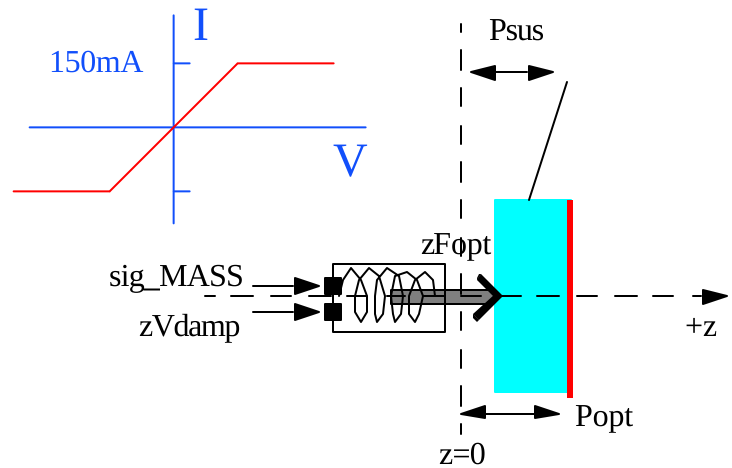
ideal vs realistic

ideal

realistic

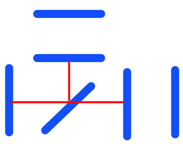
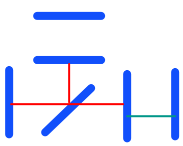
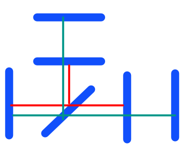
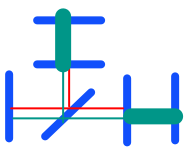


Linear Controllers: Realistic actuation modeling plays a critical role in control design.



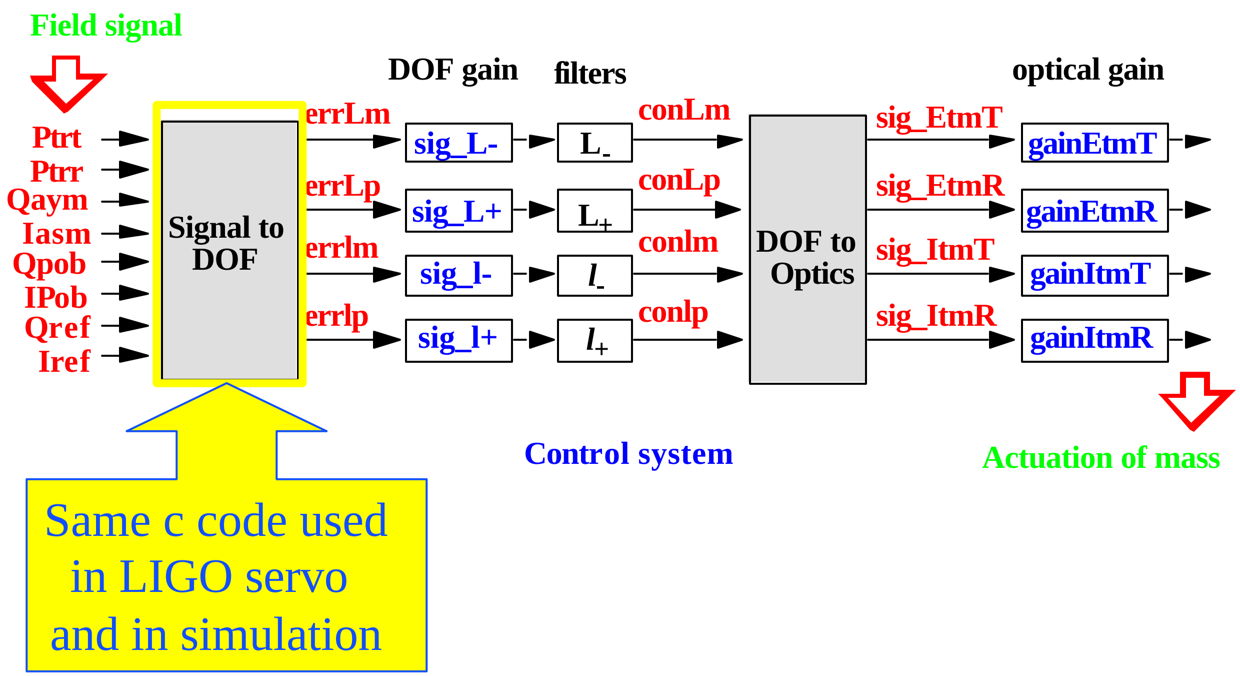
Multi step locking

change of significance and magnitude of field signals

	State 1 : Nothing is controlled. This is the starting point for lock acquisition.
	State 2 : The power recycling cavity is held on a carrier anti-resonance. In this state the sidebands resonate in the recycling cavity.
	State 3 : One of the ETMs is controlled and the carrier resonates in the controlled arm.
	State 4 : The remaining ETM is controlled and the carrier resonates in both arms and the recycling cavity.
	State 5 : The power in the IFO has stabilized at its operating level. This is the ending point for lock acquisition.

Automated Control Matrix System

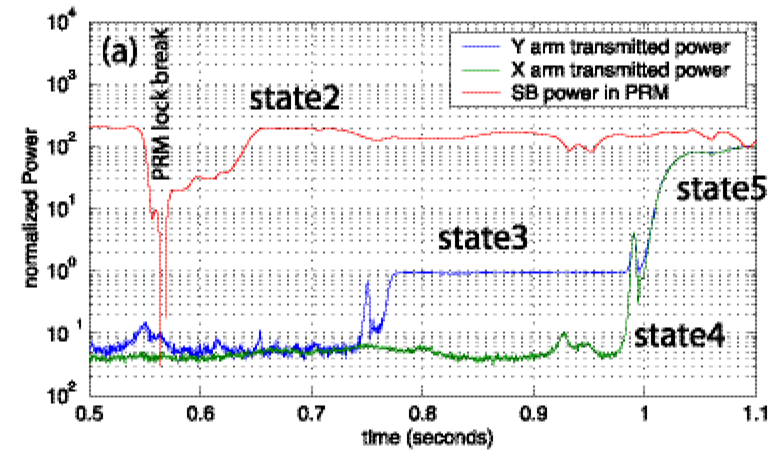
LIGO T000105 Matt Evans



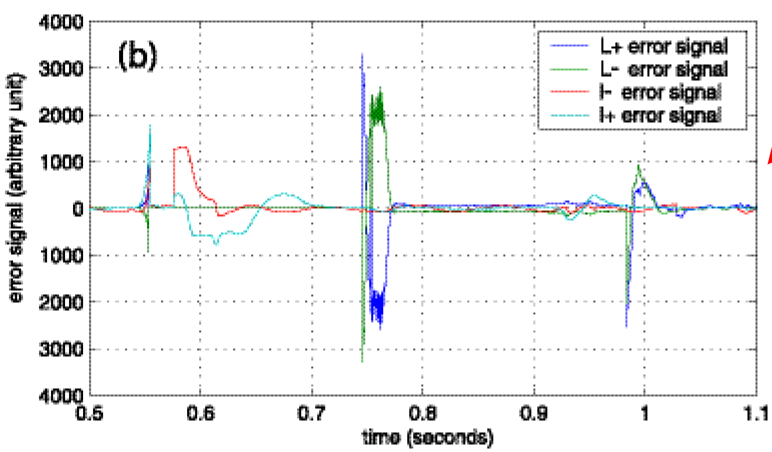
Lock acquisition

real and simulated

Figure 1. LHO 2k IFO data

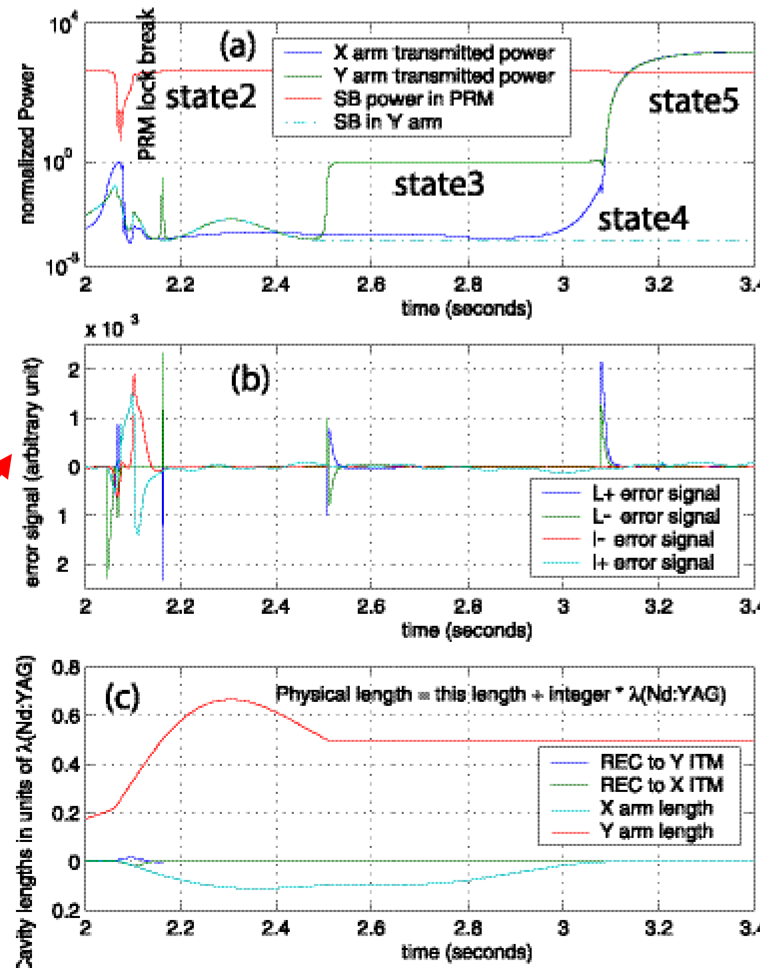


Arm powers are normalized by the power when one arm is locked.
SB power is normalized by the input SB power.



observable

Figure 2. Simulated signal



Not experimentally observable

Locked state noise simulation SimLIGO

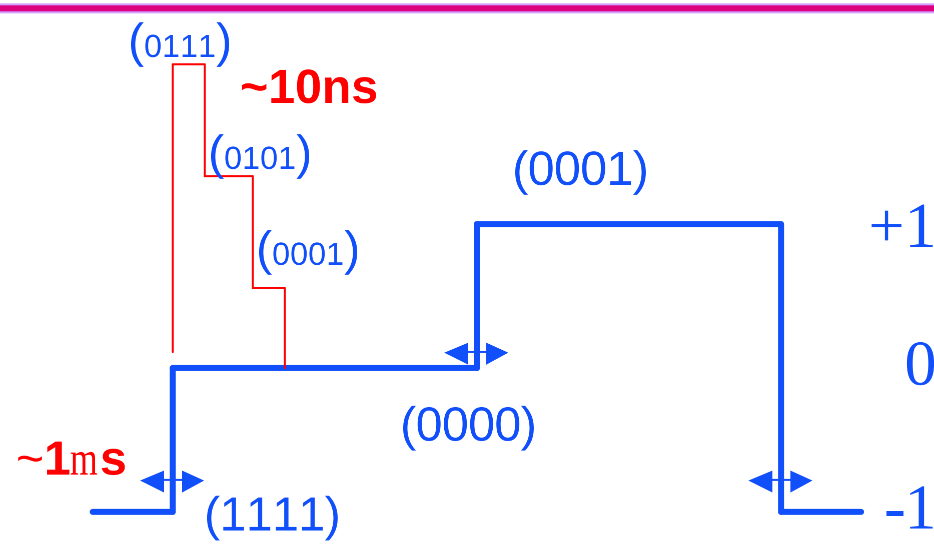
- Purpose
  - » Quantitative noise estimation of as-built LIGO
  - » Assist noise hunting and noise reduction in the commissioning phase
- Simulation includes
  - » As-built optical and mechanical parameters
  - » Seismic motion correlations among chambers
  - » 3D mirror with 4 sensors and actuators
  - » Digital LSC
  - » ASC via WFS and/or optical lever
  - » Digital suspension controller
  - » Common mode servo
  - » Mode cleaner with suspended small optics with ISC
  - » …

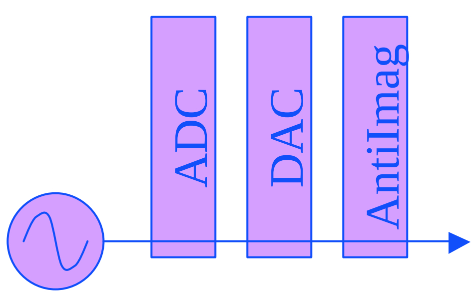
Noise sources in SimLIGO

- IFO
  - » optical asymmetries (R, T, L, ROC)
  - » non-horizontal geometry (wedge angles, earth's curvature)
  - » phase maps
  - » scattered light
- Mechanics
  - » wire resonances, test mass internal modes
- Sensor
  - » photo-detector, whitening filters, anti-aliasing
- Digital system
  - » ADC, digital TF, DAC
- Actuation
  - » anti-imaging, dewhitening, coil drivers

DAC noise
an example of simulated noise

- **Observation**
  - » glitch when sign flips
  - » strong input frequency and voltage dependence
- **Assumption**
  - » bit flipping takes time
  - » digitization timing fluctuation

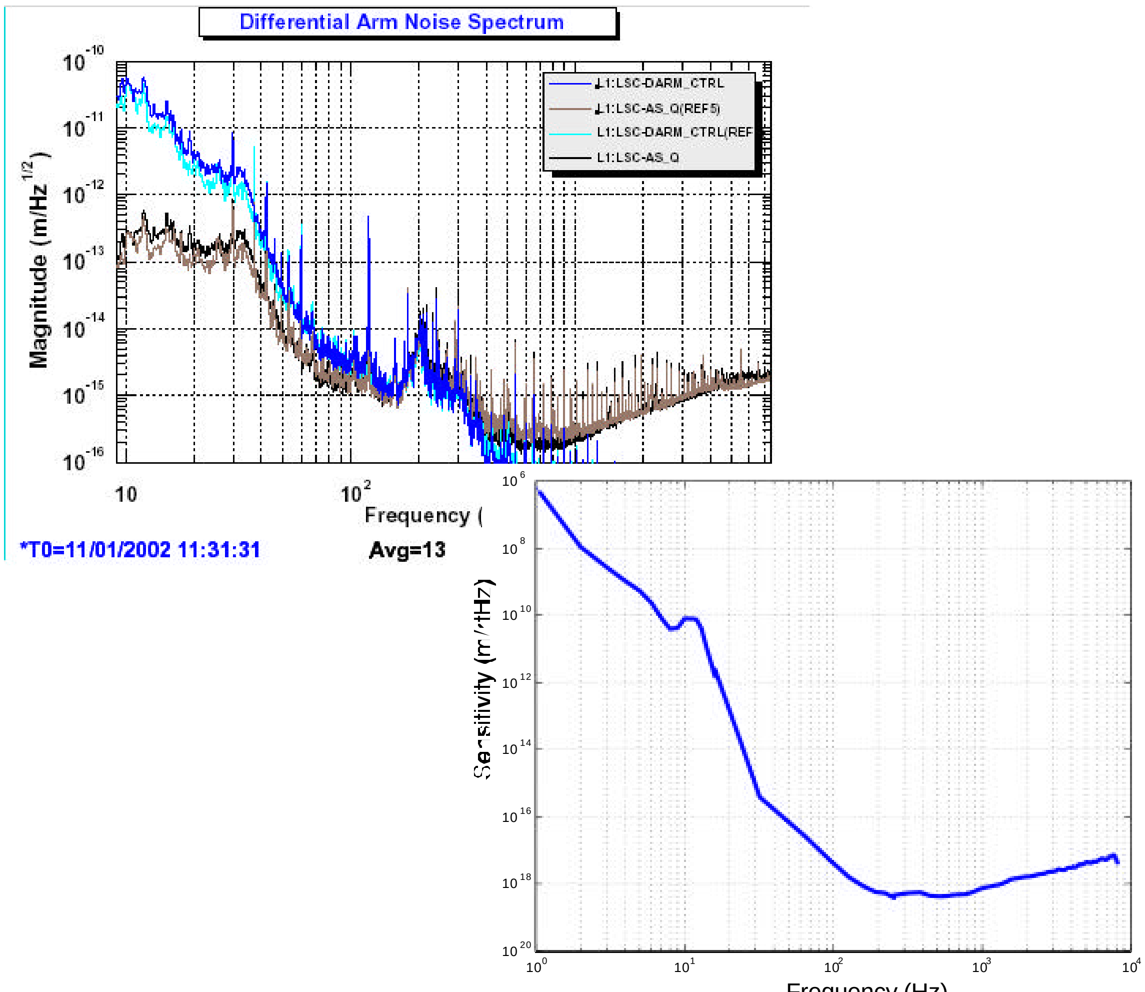




Frequency (Hz)	Amplitude (V-pk)	noise floor (µV/vHz) (**data**)	noise floor (µV/vHz) (**simulated**)
	0	**0.5**	**0.5 (input)**
19	1	**1.5**	**1.4**
199	0.12	**2.0**	**2.7**
	1	**10**	**11.6**

Simulation of sensitivity curve

noise by realistic simulation

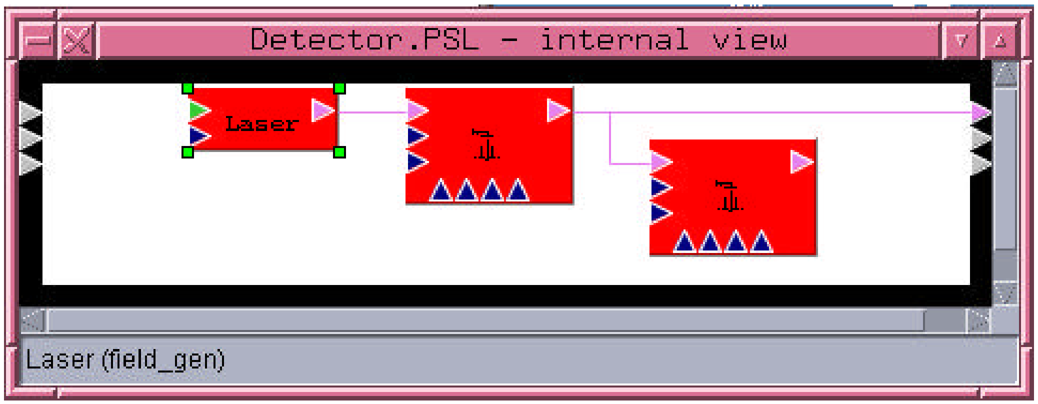


Simulation includes

- Locked interferometer
- Mechanics
- Sensor-actuator
- Servo electronics
- Signal extraction
- Simulated and measured noises
- …

Graphical User Interface

easy development and maintenance



	Current Value	Type	Data Type	notes re current v		
lambda	DEFAULT	parameter	real			
waist_size_X	DEFAULT	parameter	real			
waist_size_Y	DEFAULT	parameter	real			
distance_waist_X	DEFAULT	parameter	real			
distance_waist_Y	DEFAULT	parameter	real			
angle_resolution	DEFAULT	parameter	real		1e-9	set in PSL.Laser
compute_mismatch_curvature	DEFAULT	parameter	bool		no	set in PSL.Laser
max_mode_order	DEFAULT	parameter	size_t		-1	set in PSL.Laser
polarization	DEFAULT	parameter	size_t		1	set in field_gen
compute_option	DEFAULT	parameter	size_t		1	set in field_gen
<file>	DEFAULT					

Equations -

File Edit

```
/* naming inputs and outputs */
#inputs in gain speed offset amp freq phase noiseAmp
#outputs out
/* definition of digital filters */
velocity_integrator(v,r) = digital_filter( 1, {}, {0} );
angle_integrator(v,r) = digital_filter( 2*PI, {}, {0} );
lowPass(v,r) = digital_filter( 2*PI, {}, {-2*PI} );
/* total motion */
out = in*gain
+ offset + velocity_integrator( speed, reset )
+ amp * sin( angle_integrator( freq, reset ) + phase )
+ lowPass( white_noise( noiseAmp ), reset );
```

SimLIGO

more realistic LIGO simulation