Update of E2E

Hiro Yamamoto LIGO Lab/CIT

- LIGO End to End simulation package
- Physics update
- Architecture update
- GUI update
- Summary

Simulation group

- H. Yamamoto (1 FTE) : Manager, Salesman, Science programmer
- M. Evans (1 FTE) : Lead Scientist for e2e application
- B. Bhawal (1 FTE), V. Sannibale (1/3 FTE) : Scientist
- B. Sears (1 FTE), M. Araya (1 FTE) : User Interface programmer

LIGO End to End simulation overview

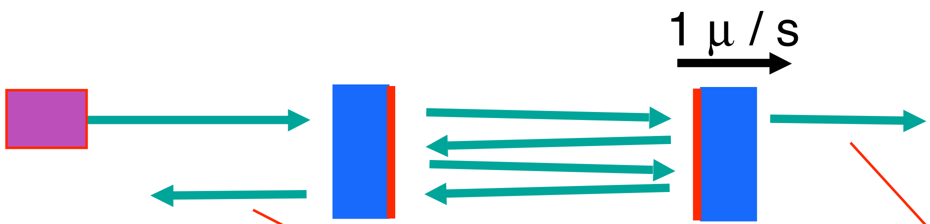
- Time domain simulation written in C++
- Like MATLAB with Interferometer toolbox
- Major physics components and tools relevant for LIGO
  - » fields & optics, mechanics, digital and analog electronics, measured noise, state space model using ABCD matrix, etc
- Flexible to apply for wide varieties of systems
  - » from a simple pendulum to full LIGO I to adv.LIGO
  - » from fast prototyping of subsystems to entire interferometer simulation
- Easy development and maintenance
  - » use of graphical front end for e2e programming
  - » object orient design for easy addition of new physics

LIGO e2e usage

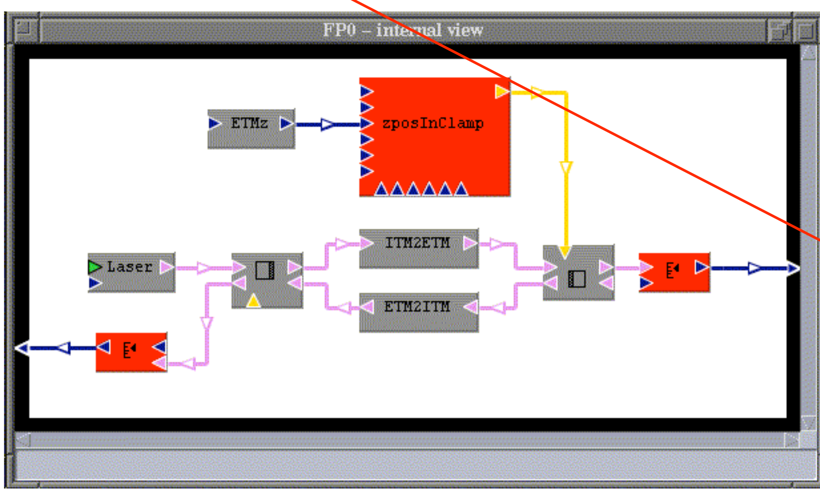
- LIGO I
  - » Lock acquisition design, original and improvements
  - » Robust alignment control design
  - » Cross check with other calculation
    - – ASC matrix miscalculation found
    - – LIGO I 4k Schnupp asymmetry mis-design found
  - » Effect of seismic noise on lock and sensitivity at LLO
  - » Detailed study of input beam (mode cleaner and mode matching telescope)
- Adv.LIGO
  - » Lock acquisition
  - » Effect of noise of input beam
  - » Radiation pressure and alignment control

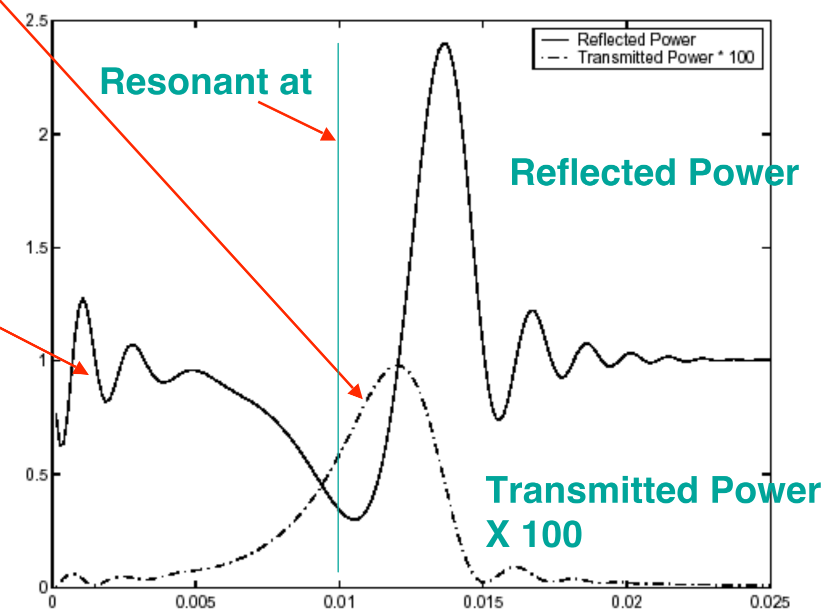
e2e example

Fabry-Perot cavity dynamics



ETMz = $-10^{-8} + 10^{-6}$ t

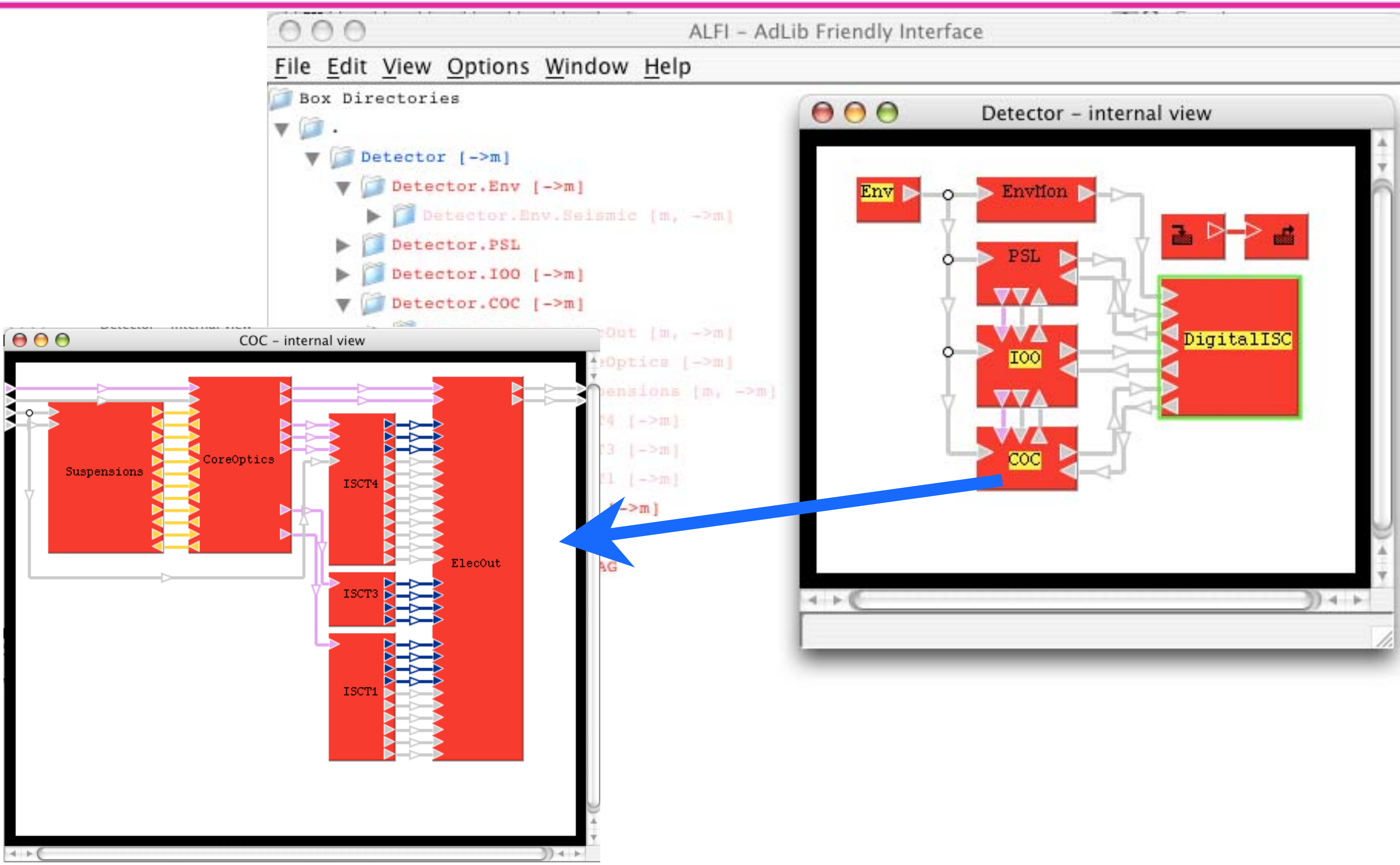




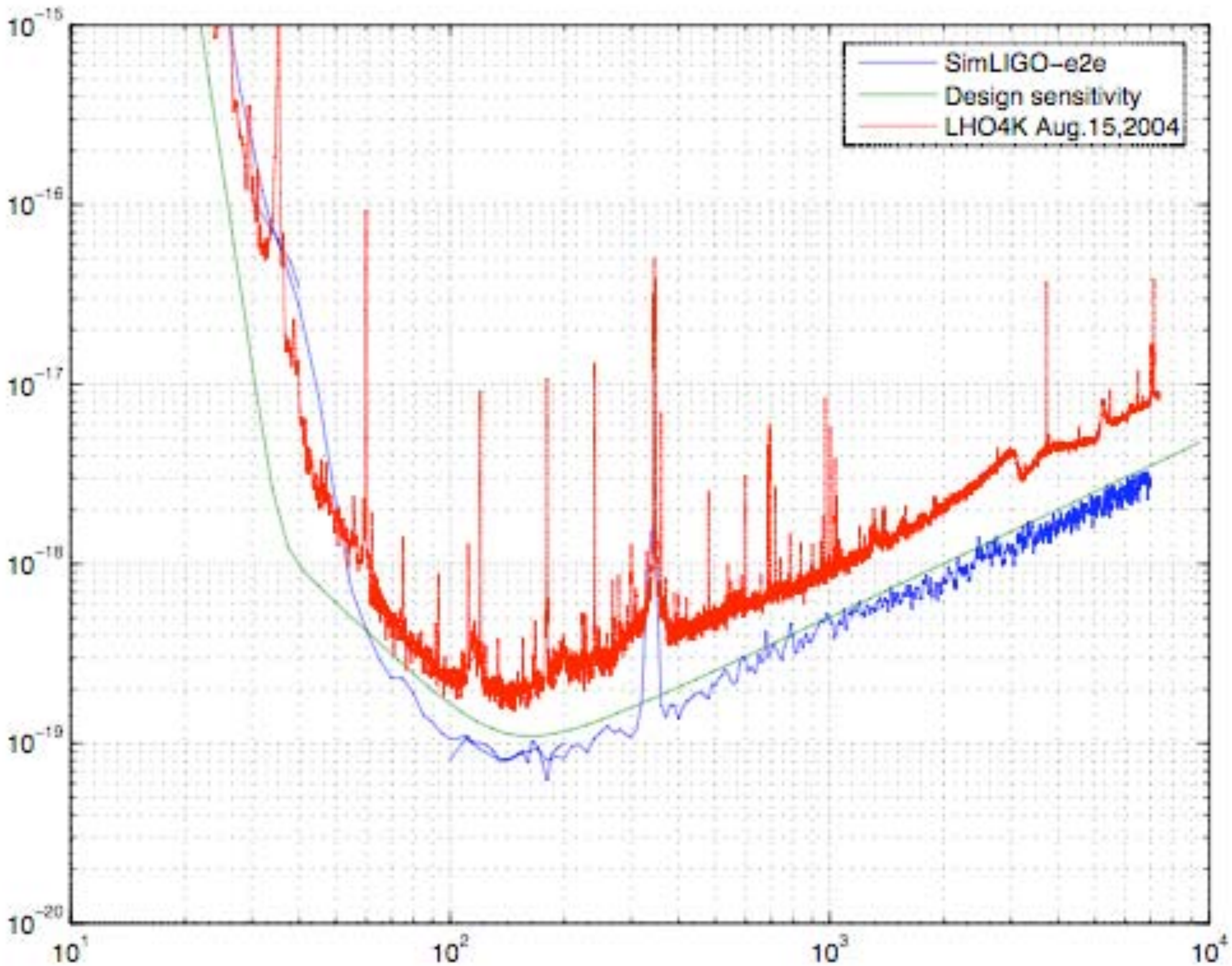
Power = 1 W, T_{ITM}=0.03, T_{ETM}=100ppm, L_{cavity} = 4000m

SimLIGO

full LIGO I simulation



Sensitivity curve

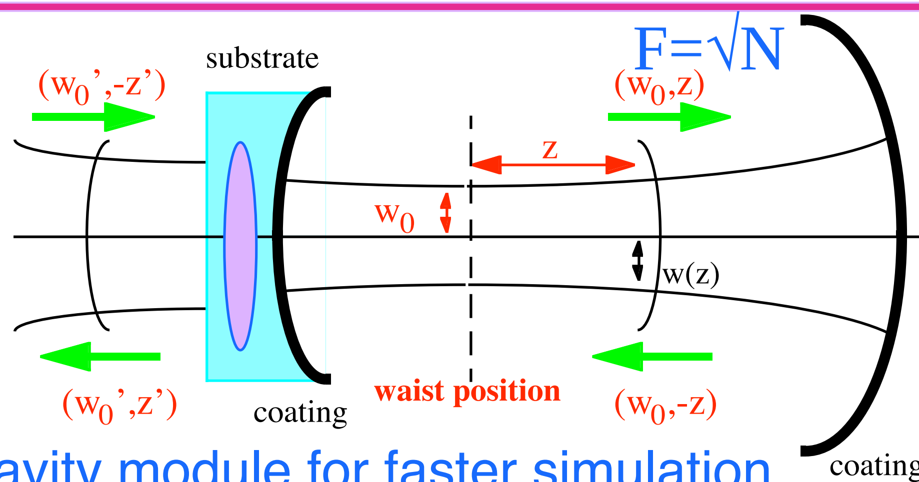


e2e physics (1)
basic

- Analog process is simulated by a discretized process with a very small time step (10^{-7}~ 10^{-3} s)
  - » Time step should be shorted than the 1/bandwidth of subsystems
  - » Not practical to simulate Core Optics and full Mode Cleaner together
- Linear system response is handled using digital filter and State Space
- Easy to include non linear effect
  - » Saturation, e.g.
- Mechanical simulation
  - » Stack - linear system using digital filter (LIGO I) and state space (adv.LIGO)
  - » Suspension - single pendulum (LIGO I) and state space and MSE (adv.LIGO)
  - » Ad-hoc thermal noise - psd matches with theory, no dynamics

e2e physics (2)

Fields and optics

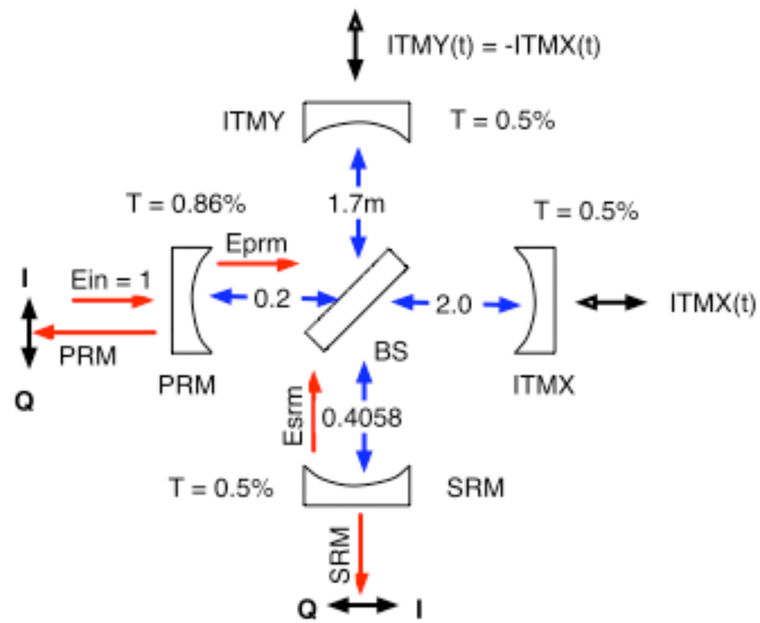


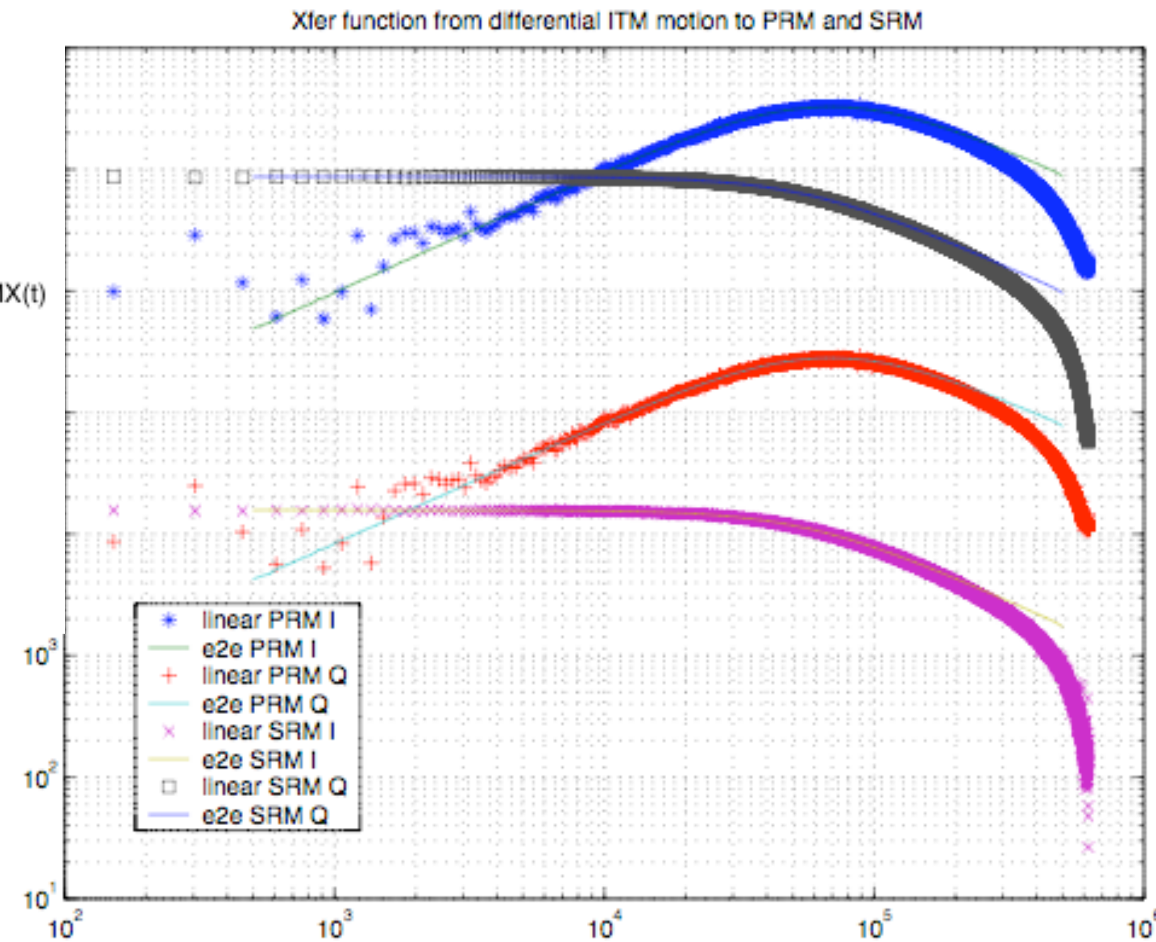
- Time domain modal model
- Reflection matrix
  - Tilt, curvature mismatch, base change
- Completely modular
  - Arbitrary planar optics configuration
- Approximation used in Michelson cavity module for faster simulation
  - Thread is used to calculate multiple sideband evolution at the same time
- Photo diodes with arbitrary shape
  - Shot noise (classical poisson noise) turned off or off
- Radiation pressure
  - Classical implementation - Photon counting on each mirror, independent of shot noise
- Thermal lensing effect by thin lens approximation using effective refractive index
  - $n = n0 - R_m / R_{thermal}$

Physics update
- fast dual recycling cavity module -

- Short cavity is a bottle neck of fast simulation
  - » Time step = cavity length / speed of light
- Dual recycling Michelson cavity
  - » Assuming all quantities, mirror motions and field evolutions, change linearly for a duration of τ_L= long arm length / speed of light
  - » Explicitly calculate the field at time t using data at t-τ_L
  - » Speed gain = arm length / Michelson cavity length ~ 1000
- Formulation for scalar field case by Mathematica : done
- Implementation in matlab for validation : done
- Comparison between e2e elementary modules (very slow, but works) vs linear model in matlab : done
- Need to build a e2e module for fast simulation in e2e
- Need to calculate using modal model

Dual recycling fast model
- comparison with e2e -





Need to build e2e module to run e2e fast

Architecture update
FUNC_X using run time dynamic linking

- AdLib C++ class architecture by Matt hides all the complexity to do with time domain ticking and data exchange. Physicists can concentrate on physics. That's good.
- To add a new capability, e.g., locking algorithm, one wrote a module using C++, compiled and liked (difficult and time consuming) or built the functionality by combining modules (tedious and runs slow). That's not good.
- Using a new module, FUNC_X, one can easily add new functions using C++.
  - » out0=in0+in1;
  - » full C++ code with many lines
  - » Compiled and dynamically linked automatically at run time
- Much more flexible in development and runs faster

Why FUNC_X is easier/better than writing a C++ code explicitly

- out0 = in0 + in1;
  - » Not necessary to define class explicitly
  - » Not necessary to bother how to communicate with other modules
  - » Inputs and outputs are automatically setup
- Run e2e, not C++ compiler
  - » Compilation and dynamic link of necessary codes are automatically done
- out0 = amp*sin(omega*time_now());
  - » Macros, amp and omega, are automatically setup
  - » Utility functions, time_now(), are automatically setup
- No lock acquisition code in modeler
  - » Scalar version, modal version, revised version, obsolete version… NO MORE

GUI updates

- Bundler
  - » Multiple data streams combined into one stream
  - » Easier to develop, maintain and understand
- Opaque box
  - » Box contains many elements, many settings for all elements
  - » Make a box to behave like a simple built-in module
  - » Only important settings are visible and complexity is hidden
- Dual interface of FUNC_X setting
  - » A full blown C++ code can be assigned to a FUNC_X module
  - » The interface can be chosen to behave like a simple module with several physical settings, C++ code being hidden

Summary

- LIGO I simulation is ready
  - » Good playground for length and alignment control design
  - » Sensitivity curve properly simulated
- LIGO I simulation needs improvements
  - » Assist to improve another factor of 2
  - » Michelson cavity is degenerate and BS curvature affects beam profile
    - – Better modal model
  - » More realistic thermal lensing model
  - » Lock loss study needs more reality
  - » more noise, more reality
    - – scattering noise, acoustic coupling, beam clipping
- adv.LIGO simulation demands more
  - » physics (dual recycling cavity, better Modal Model or faster FFT)
  - » Speed (thread)
  - » Accuracy (quadruple precision by hardware !?)