LIGO End-to-End Simulation Model

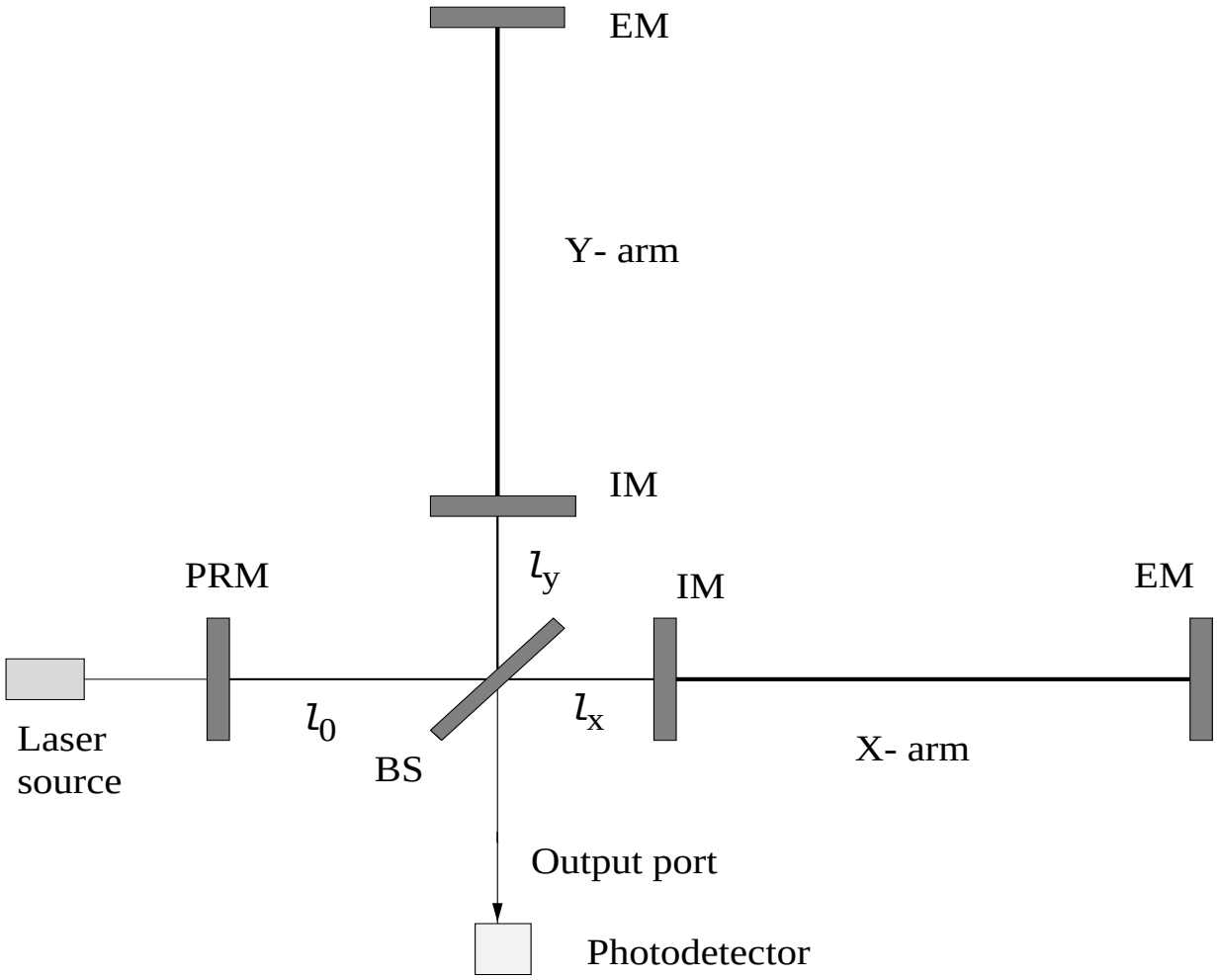
Biplab Bhawal

LIGO, Caltech

- Contributors: Matt Evans, Malik Rakhmanov, Hiro Yamamoto

- March 10, 2001
- Pac. Coast Grav. Meet
- Univ. of California, Santa Barbara

LIGO Interferometer



- Interferometry with suspened mirrors
- A coupled, dynamic, multi-length, non-linear system

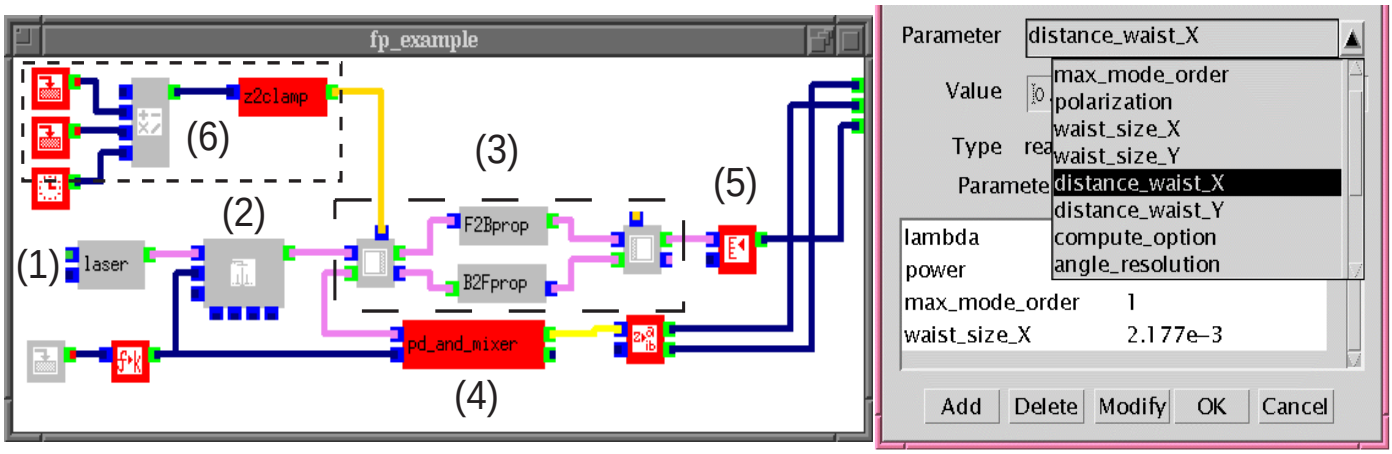
E2E : Definition & Aims

- Definition: A general simulation framework which enables studies of optical, electronic and mechanical components and their interaction in interferometer-based gravitational-wave detectors.

- Aims :
  - Better understanding of LIGO-Physics and LIGO-techniques : Dynamics, Misalignments, Noises, Lock-acquisition.
  - Trouble-shooting
  - Helping future design of full LIGO or various subsystems

How To Use E2E

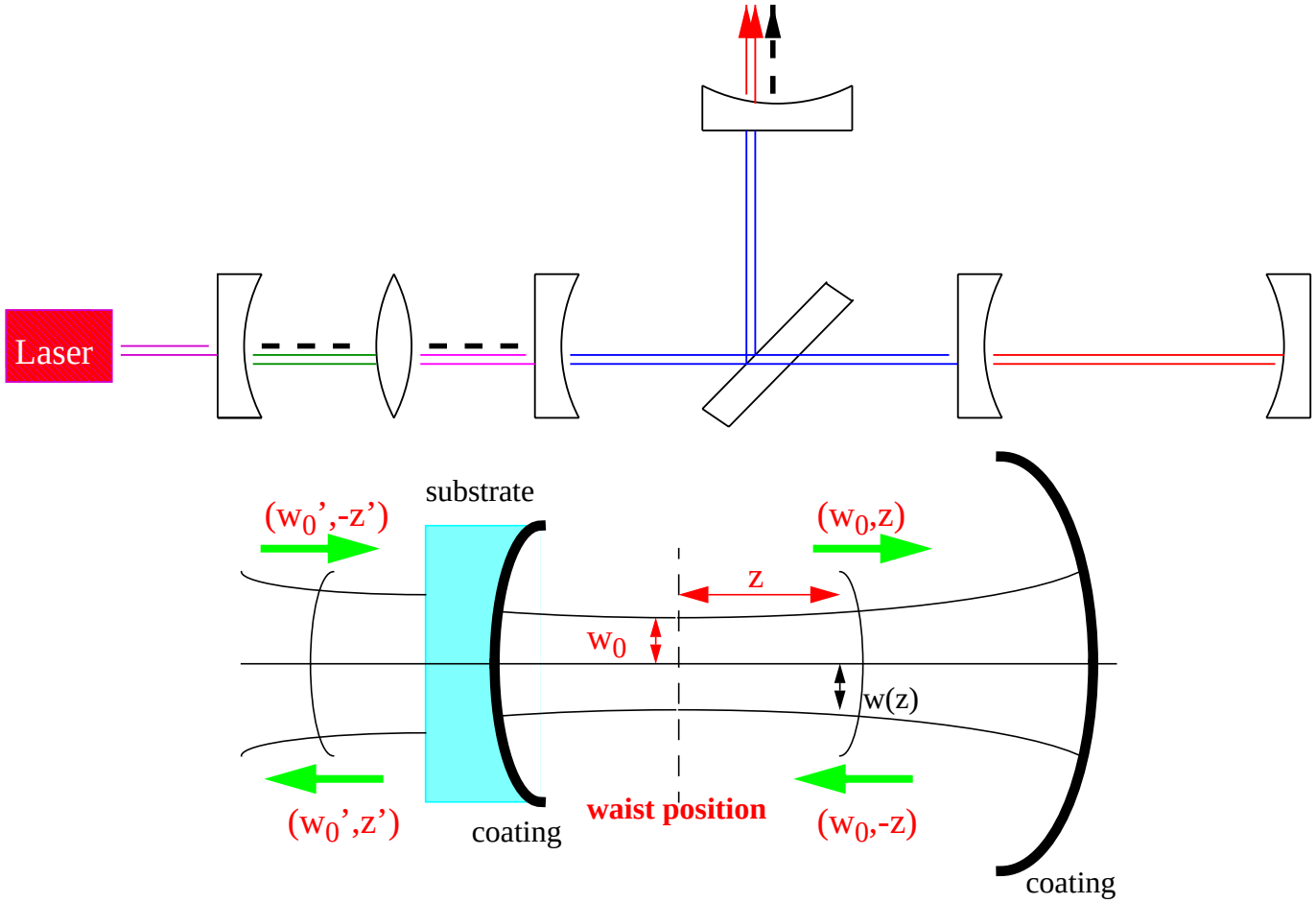
- Build up your laboratory:
  - (Like Matlab) E2E provides toolbox with specialised tools for studying all kinds of interferometric GW detectors
  - Set-up your experiment using an easy-to-learn graphical-user-interface called Alfi
  - Basic tools: laser source, mirror, propagator, telescope, lens, modulator, detector, digital filter, power-meter, ...simple algebraic and logic functions..... etc.
  - Set parameters of modules, make connections among them, ... run ....
  - Example E2E box: Fabry-Perot Cavity (using primitive modules)



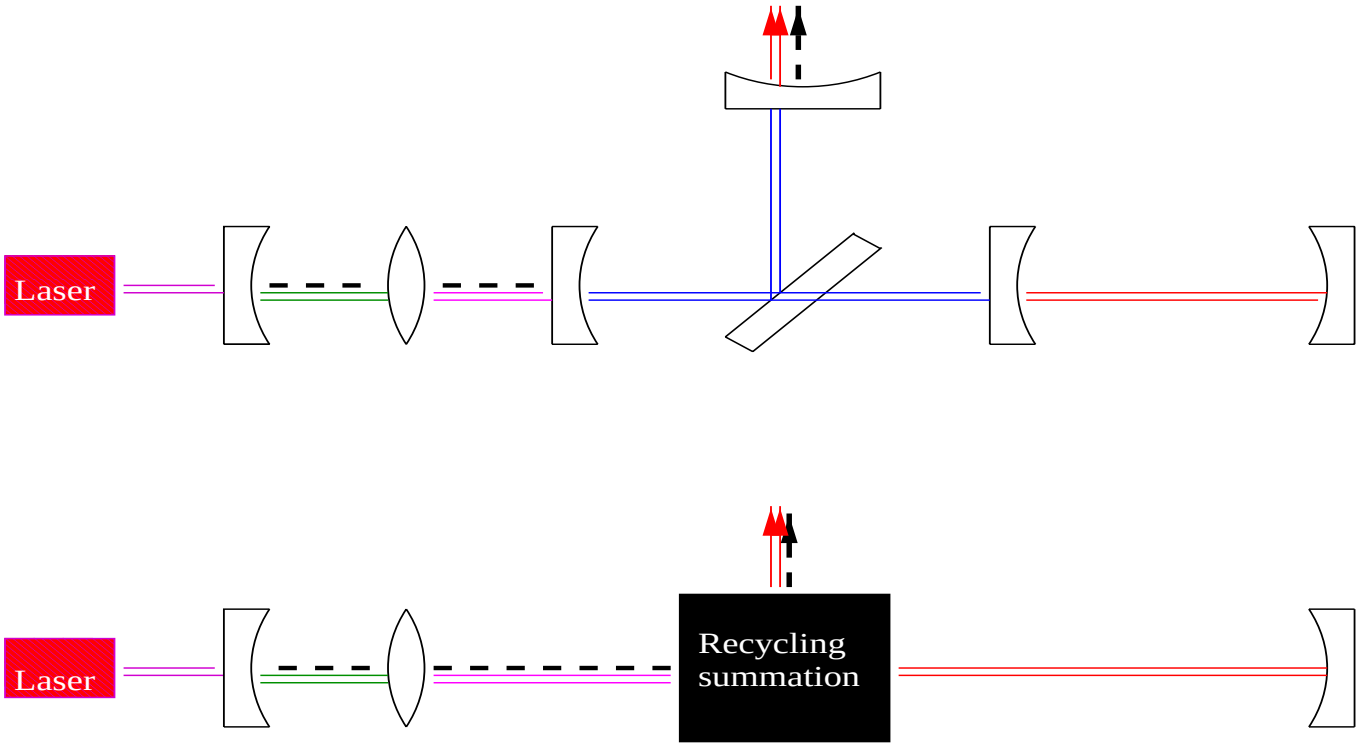
(a) GUI front-end of e2e (b) parameter setting windows

Time-domain Modal Model

- "field" carries information about frequencies and spatial modes
- spatial modes in Hermite-Gaussian basis
- Tilt, shift, curvature mismatch are treated using mode decomposition matrix
- Modal basis changes (i) after passing thru lens/curved mirror (ii) on reflection at an angle from a curved mirror



Composite Systems
(*summation modules*)

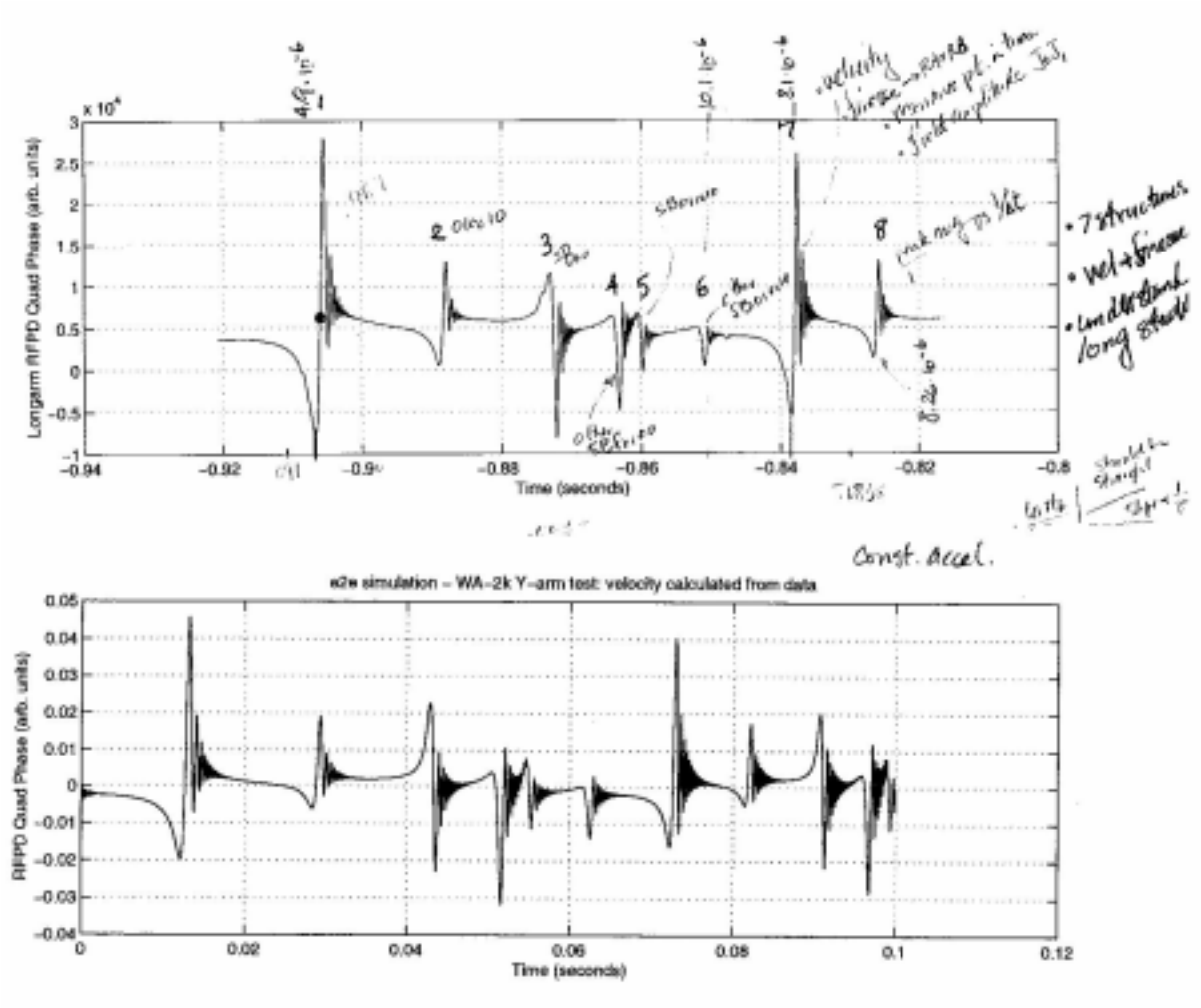


- Aim: fast computation
  - Fabry-Perot cav
  - triangular cavity
  - power-recycled Michelson cavity or the LIGO recycling cavity

Modularity & Flexibility
of E2E

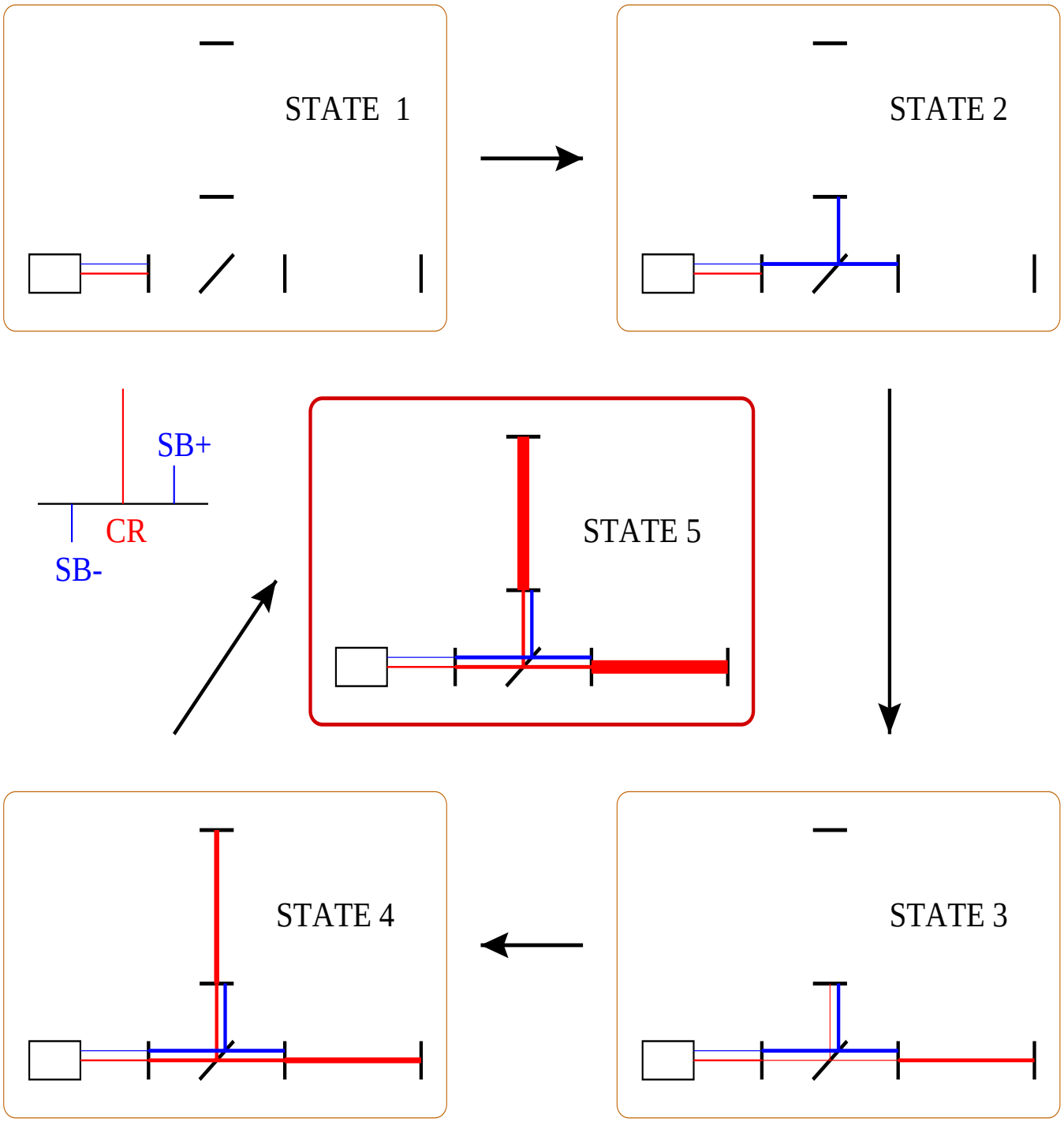
- You can study any configuration
- You can look at field at any port and analyze
- You, as user, donot need to know C++ or even programming or to touch the original code.
- You, as developer, can introduce new physics modules easily without touching the rest of the code but utilizing full advantage of that.

Hanford 2Km Arm: First light
(Actual and E2E-simulated)



›› December, 1999

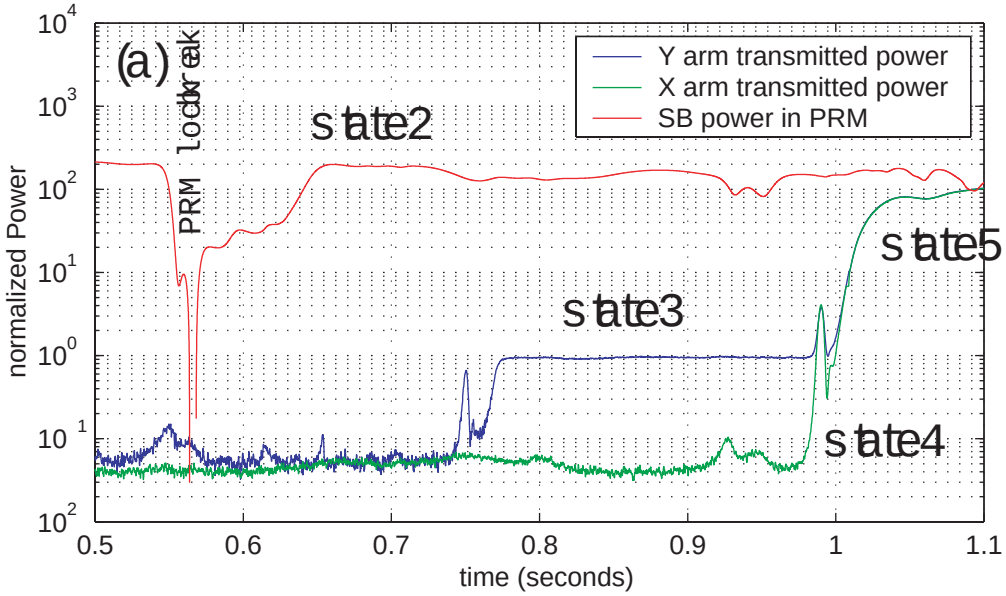
LIGO Lock-Acquisition States



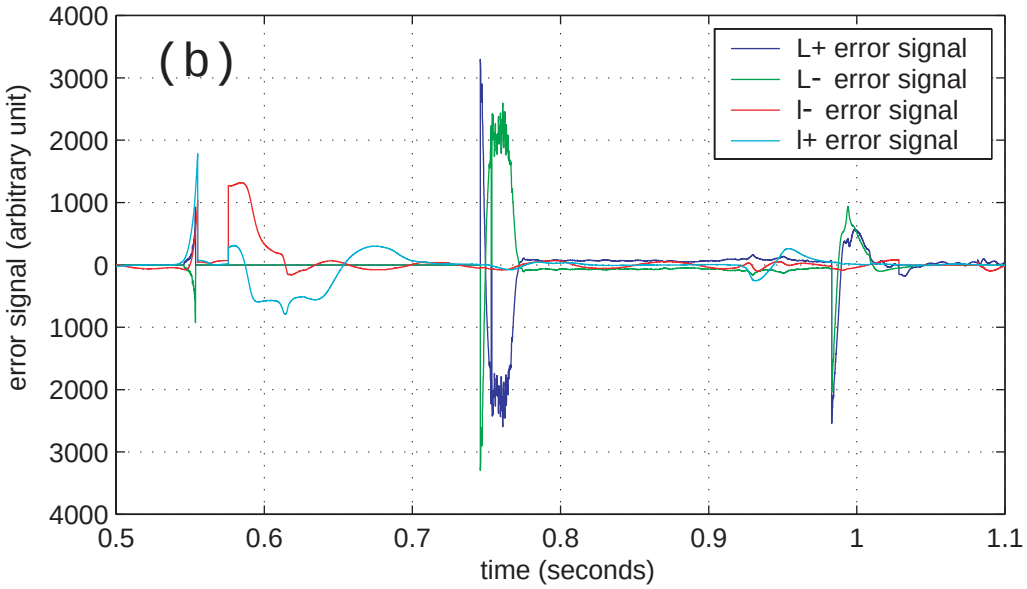
Lock-Acquisition
(actual Hanford 2KM IFO data)

- December, 2000

Figure 1. LHO 2k IFO data

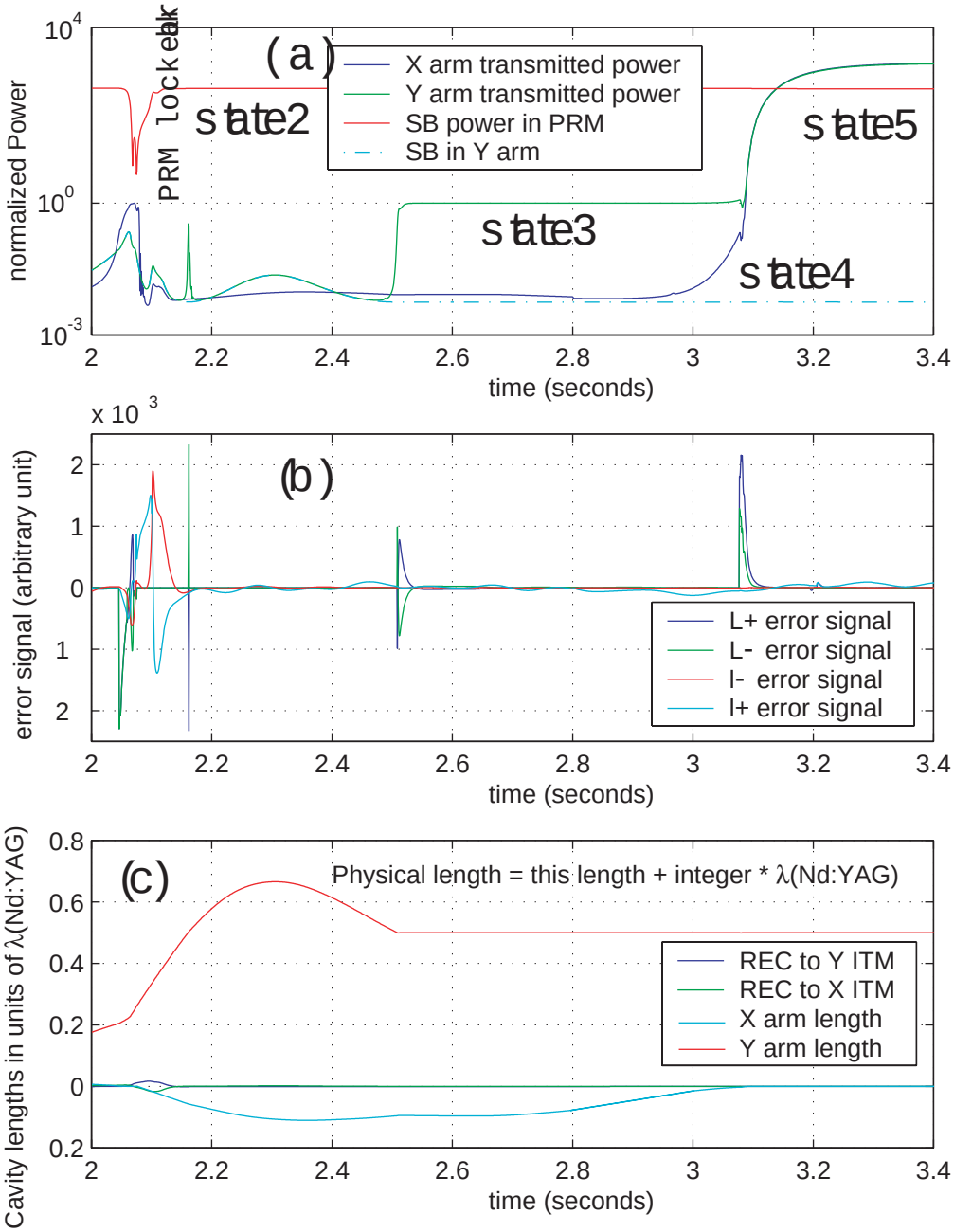


Arm powers are normalized by the power when one arm is locked.
SB power is normalized by the input SB power.



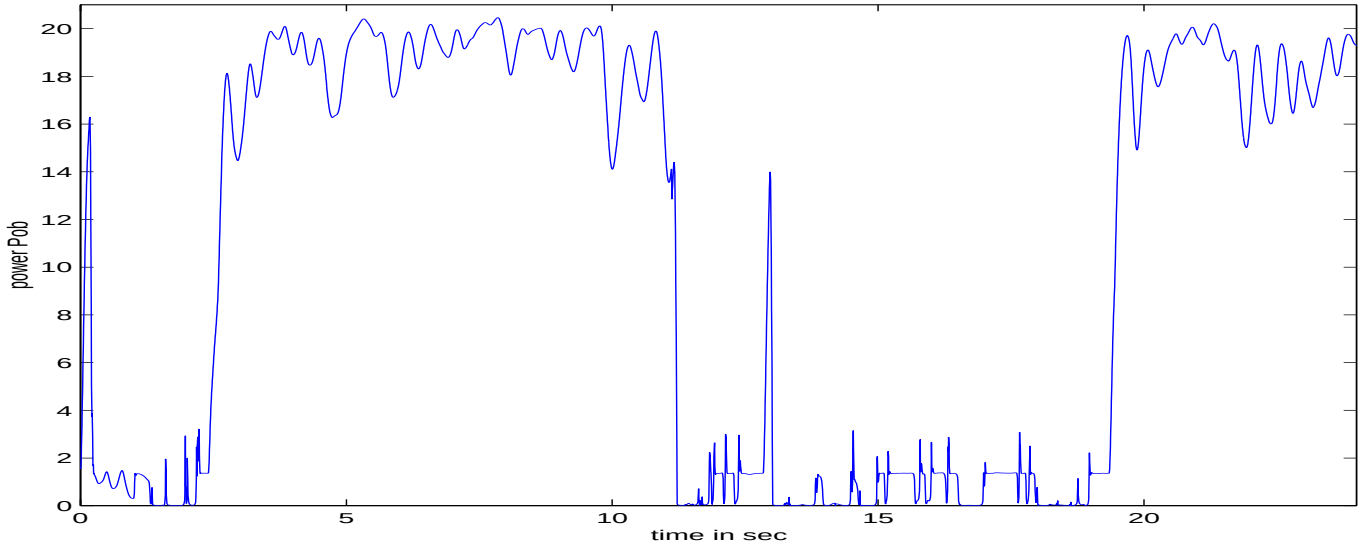
Lock-Acquisition
(E2E-simulated data)

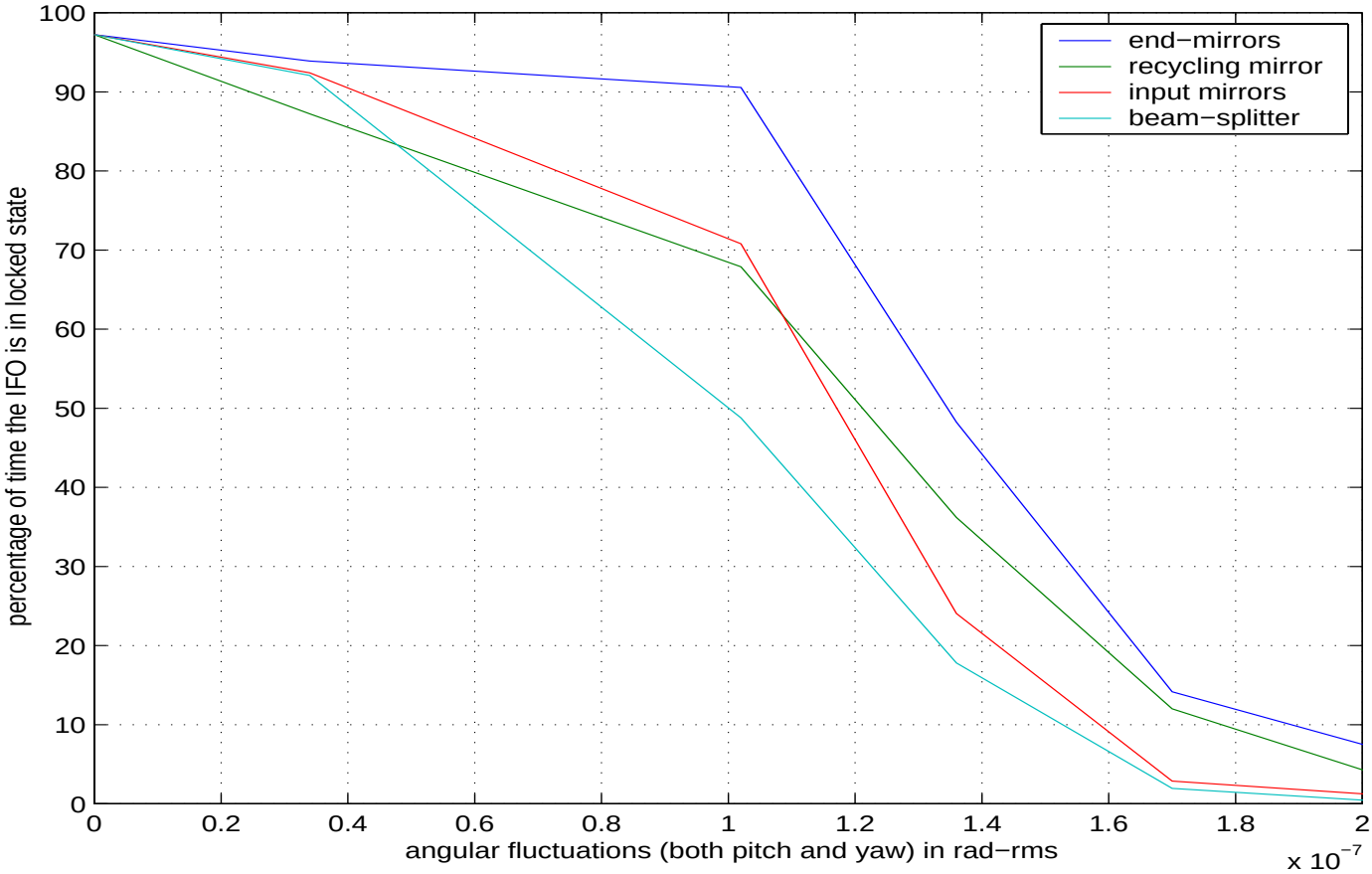
Figure 2. Simulated signal



Lock-acquisition &misalignments

(Study: power fluctuations, lock duration.....)





E2E for GW Data-analysis

- The modeled IFO is under your control
- Put noises and complexities to whatever level you feel comfortable about.
- Simulate Grav. waves
- Look at changes in IFO signals
- Generate data-stream and use your tools to analyse and extract signals

- OR
  - Try to understand the actual data-stream and noises in it comparing with simulation that includes causes you guess.