Advanced LIGO Simulation

Hiro Yamamoto, LIGO Lab in Caltech

LIGO I experience

FP cavity : LIGO I vs AdvLIGO

Simulation tools

Time domain model

LIGO I experience

Core Optics Design

Carefully studied the thermal lensing effect using a static interferometer simulation tool

Concluded that the thermal effect is not crucial

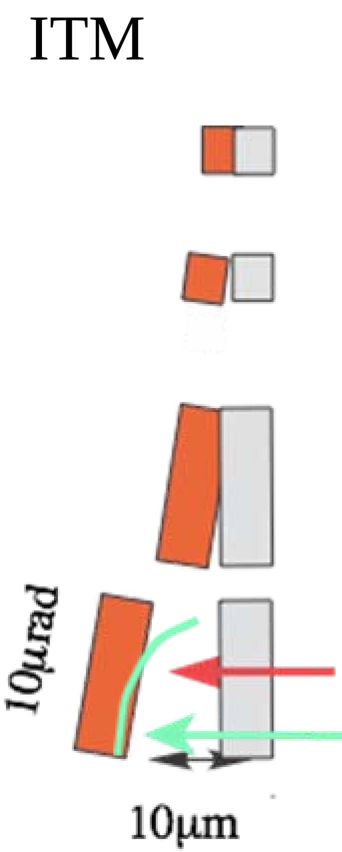
...

Commissioning revealed that thermal compensation system (TCS) is needed

TCS under development for advLIGO was adapted to cure the problem

Better simulation tools could have given better prediction

FP : LIGO I vs adv.LIGO - 1



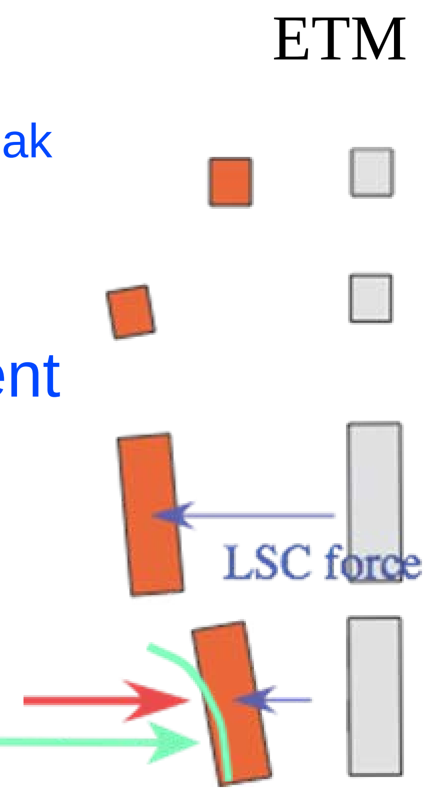
ITM, ETM actuation

- » LIGO I : direct actuation by coil and magnet
- » adv.LIGO : ITM no direct actuation, ETM weak electro static actuation

Radiation pressure : 50 time larger

Optical string : input power dependent

Angular instability due to radiation pressure torque



FP : LIGO I vs adv.LIGO - 2

- To suppress angular instability and thermal noise
  - » concentric design (length = 4000m, ROC=2076m, beam=6cm)
- Small change of mirror ROC affects carrier mode
- Mirror size affects performance
  - » advLIGO : beam (6cm) / mirror (17cm) < LIGO I : 3cm / 12cm
- Difficult choice of ROC and tight polishing tolerance
- Thermal deformation
  - » advLIGO : surface figure changes which affects the carrier mode
- Thermal compensation
  - » LIGO I : CO_2 laser on ITM -> still not perfectly corrected
  - » advLIGO : Ring heater -> hard to correct

Simulation tools

Time domain

- » end to end simulation packaged developed and used for LIGO I
- » design tool - primary or secondary
- » test drive in non-stationary and/or complicated systems
- » slow
- » beam profile too coarse

Static Interferometer Simulation

- » Details of fields in realistic core optics
- » Effects of imperfections of optics
- » ROC and its tolerance

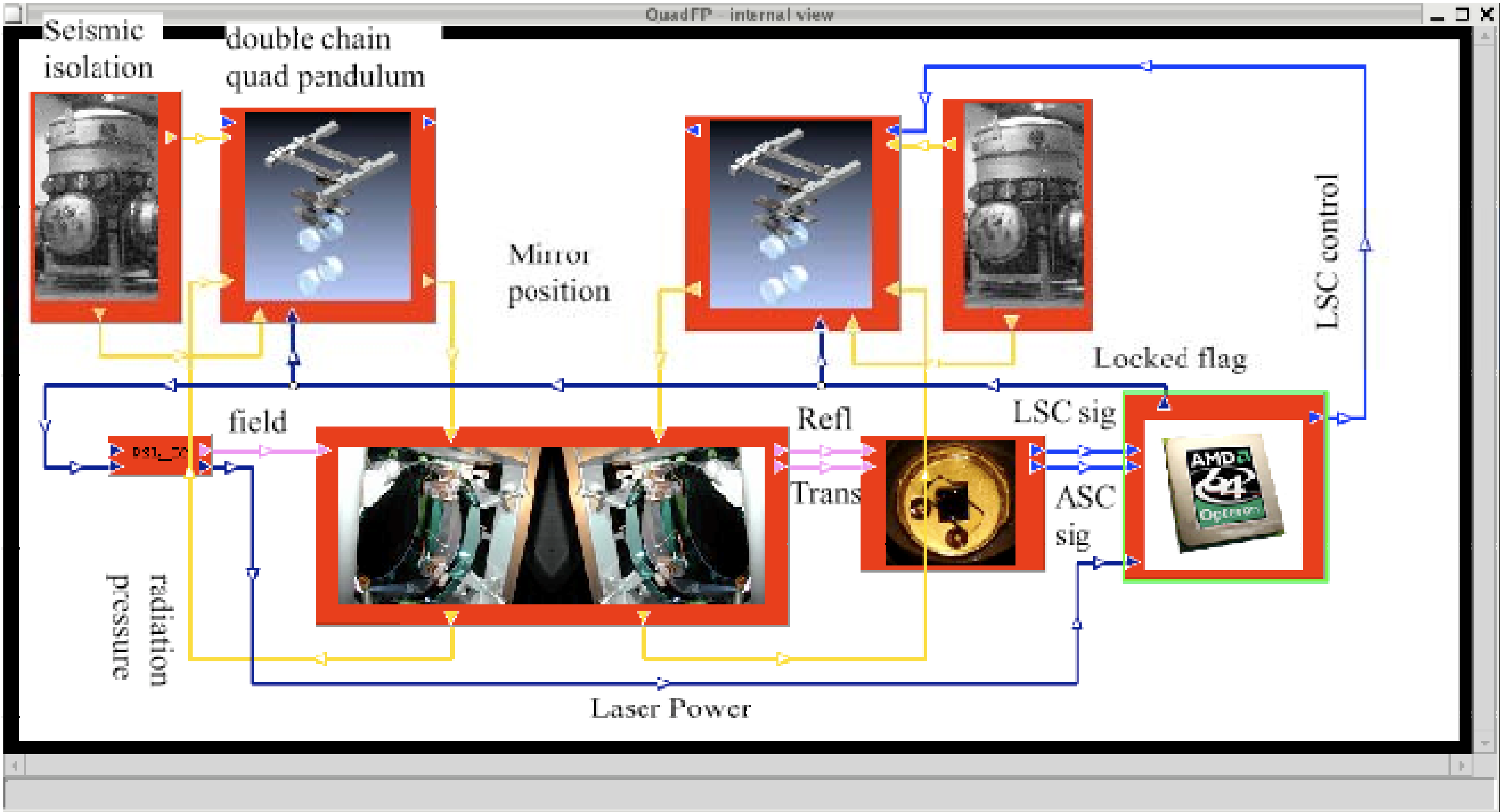
Frequency domain

- » stationary state
- » control design tool

e2e software ingredients

- matlab-like generic programming environment tailored for GW interferometer study
  - » object oriented system developed in house at Caltech using C++
- Graphical User Interface
- statespace, digital filter
  - » mechanical system simulation of other subgroups' models
  - » control systems
  - » quad precision option for steep spectrum
- c/c++ code integrator
- parallelization
  - » static and dynamic

e2e example
FP cavity



e2e physics ingredients

- primitive optics, compound optics
  - » mirror, propagator, telescope, etc
  - » fast simulation of compound system
    - – dual recycled Michelson
- Shot noise and radiation pressure noise by photon counting
- Modal Model for spatial profile of beam and optics
- Triple (input optics, PRM, SRM, BS) and Quadruple (ITM, ETM) pendulum
  - » Mark Barton (suspension subgroup) provides ABCD matrix
- HAM and BSC seismic isolation system
  - » parameterization of design performance
  - » waiting for subgroup models

Dual recycled Michelson cavity

~100 times faster simulation by sacrificing frequency response at 10 MHz down to 100kHz

planewave or TEM00 only approximation

» to be expanded to use modal model

use linear approximation

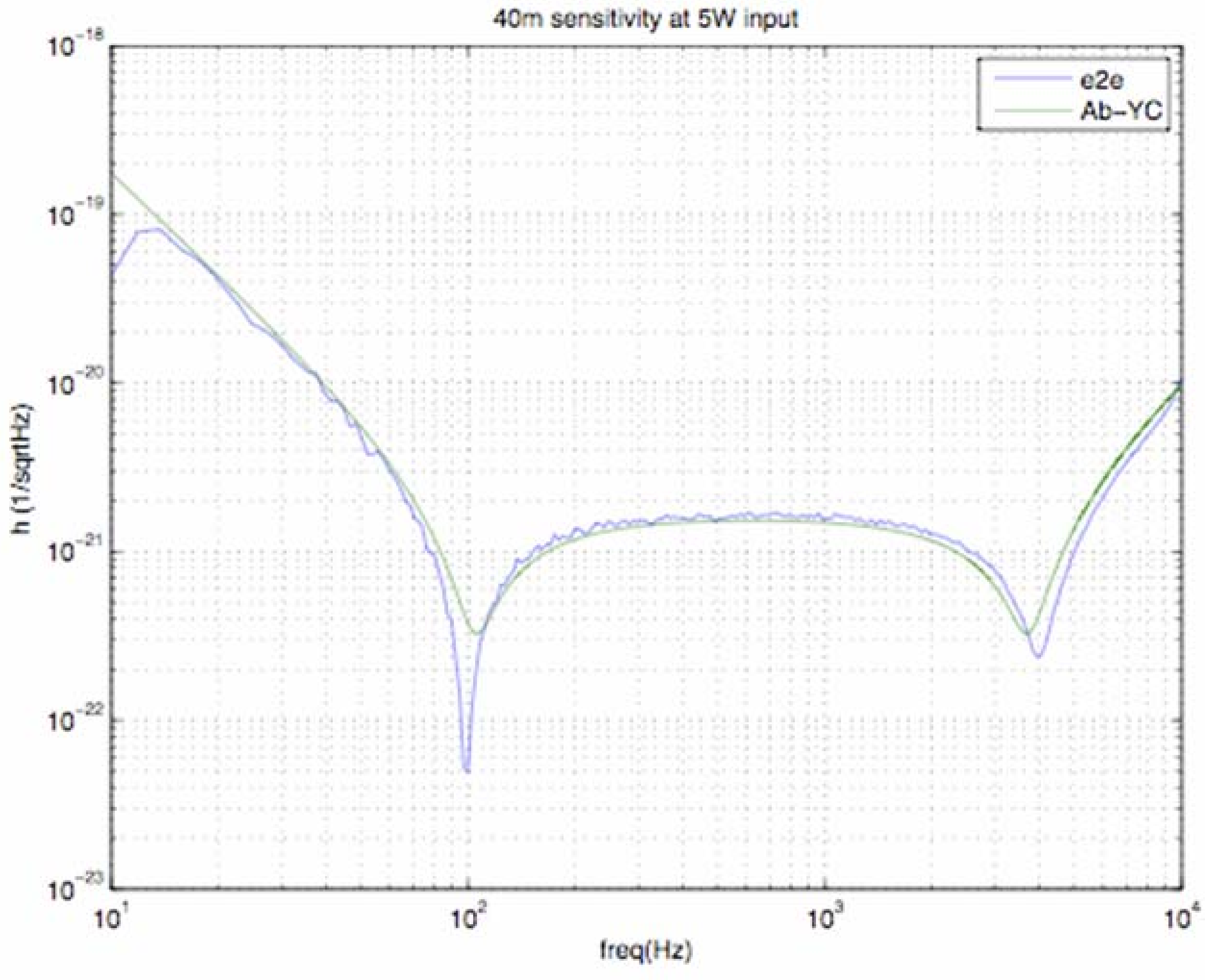
» all physics quantities, field and positions, change in linear between on time step

» needed for frequency noise study

C++ class independent from e2e framework

Injection ports for scattered light study

Quantum noise



Parallelizing using thread

Parallelizing the GW simulation is difficult

- » all are sequential

Module level parallelization

- » Single and dual recycled Michelson cavity modules
- » Evolution of each sideband fields are calculated using different threads

Dynamic parallelization

- » Analyze speed of each component and dependence
- » Group related modules to one simulation chain
  - – each seismic isolation system and pendulum
- » Run independent chain using separate threads
- » Merge simulation chains when needed
  - – cavity, error signal

Hybrid of graphical interface and C++ coding

Graphical interface

- easy to understand the global structure
- hard to implement logics, like ISC

C/C++ coding

- suited for sequential coding

FUNC_X module

- C++ code is automatically compiled and linked dynamically at run time

e2e example FUNC_X

Base Node's Comments [READ ONLY]

Base node 'FUNC_X_2VxV' comments:
2 input vector, 1 output vector function. Input name is inVec0 and inVec1 and output name is outVec0.
Example : Cross product of two vectors
outVec0[0] = inVec0[1]*inVec1[2] - inVec0[2]*inVec1[1];
outVec0[1] = inVec0[2]*inVec1[0] - inVec0[0]*inVec1[2];

	Current Value	Type	Data Ty...	notes re current ...	DEFAULT Value	notes re DEFAUL...
outputVectorSize_0	32	parame...	integer		16	set in FUNC_X_2...
Equations	@@INCLUDE LSCCode.cc	parame...	string			set in FUNC_X_2...
MemberDecl	@@INCLUDE LSCCode.h	parame...	string			set in FUNC_X_2...
MemberImpl	DEFAULT	parame...	string			set in FUNC_X_2...
Global	@@INCLUDE LSCCode.h	parame...	string			set in FUNC_X_2...
Header	DEFAULT	parame...	string			set in FUNC_X_2...
Constructor	@@INCLUDE LSCCode.cc	parame...	string			set in FUNC_X_2...
Destructor	DEFAULT	parame...	string			set in FUNC_X_2...
DoWhenGlobalChanges	DEFAULT	parame...	string			set in FUNC_X_2...
showInstanceSettings	DEFAULT	parame...	bool		false	set in FUNC_X_2...
#include	DEFAULT	parame...	string			set in FUNC_X_2...

Use of e2e for adv.LIGO

- FP with quad pendulum
  - » lock force requirement
  - » alignment control and stability
  - » power ramp up
- Simplified advanced LIGO : dual recycled Michelson with FP arms
  - » optical response
- Full advanced LIGO simulation with interferometer sensing and control
  - » advanced LIGO framework
  - » 40m model

Future issues

- Modal model version of dual recycled Michelson cavity
- More precise field profile tracing
  - » FFT in time domain?
- 96bit real
  - » quad pendulum spectrum, f^{-8}, not correct above 15Hz (comparing double precision statespace vs quad precision)
  - » Cavity signal = ITM position - ETM position
- Better implementation of quantum noise
  - » injecting vacuum from dark port?
- Speed