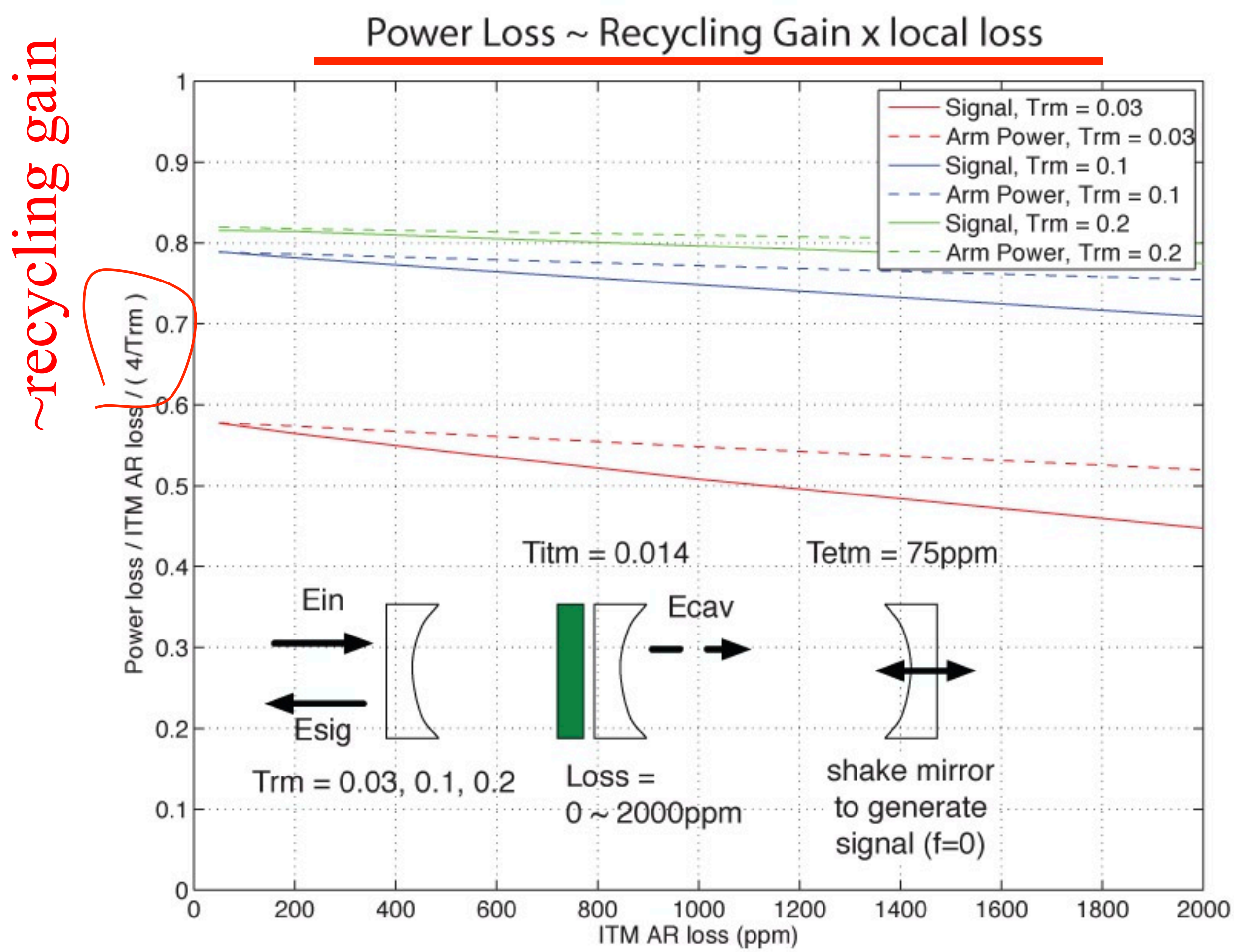
SIS 2010
Stationary Interferometer Simulation

- One time loss vs total loss
- SIS as interferometer simulator
  - » FFT-based field calculation written in C++
  - » FP and coupled cavity with BS
    - – Fast simulation of stable cavity
  - » Signal sideband generation
- SIS as analysis tool
  - » Random surface specified by analytic form or by real surface map
  - » Thermal lensing
  - » Calculation tools : PSD, modal expansion, etc
  - » Looping
- Future of SIS
- matlab class : Hankel toolkit

One time loss vs total loss

simulation needed even for semi-quantitative loss estimation



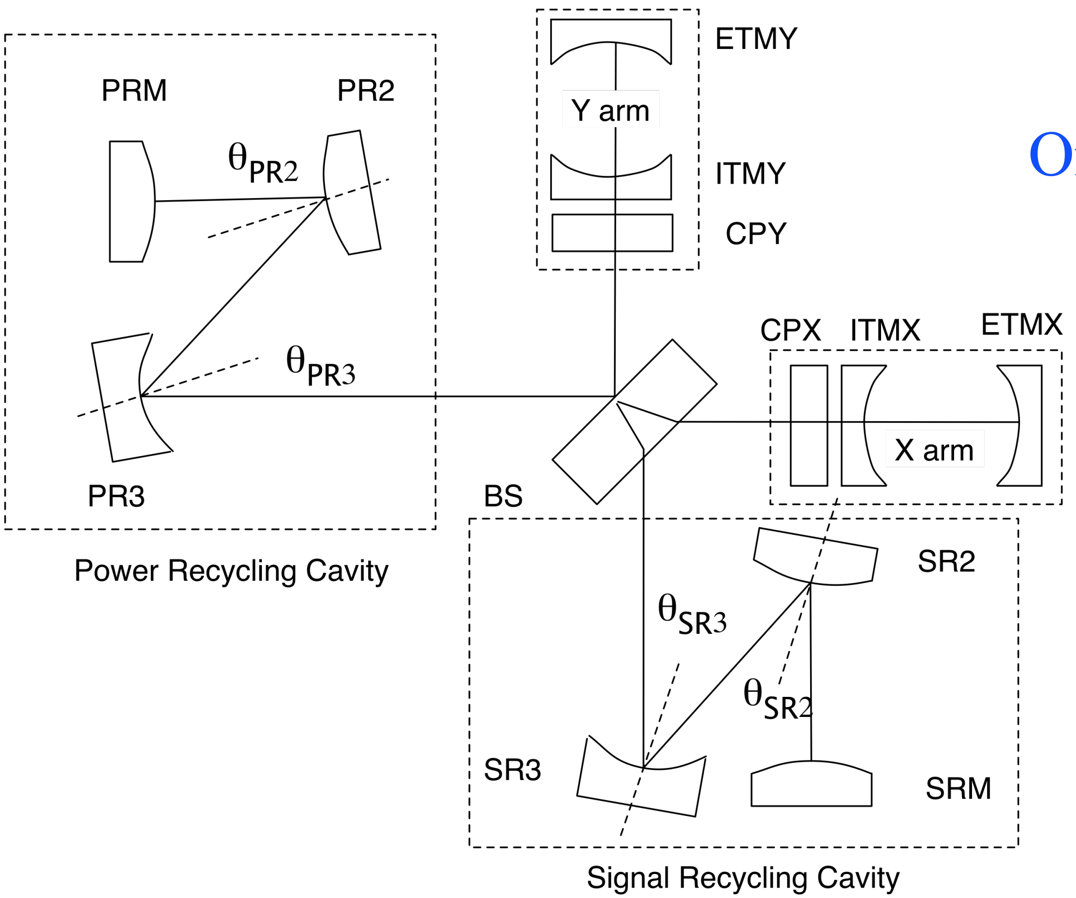
- Loss depends on the lock condition

- Loss due to mode mismatch shows similar relation, but much more complicated

SIS
Motivation and usage

- AdvLIGO design tool
- Interferometer configuration trade study
  - Demonstrated that the stable configuration is more immune to imperfection
- Effect of finite size optics
  - RM3&BS, flat, wedge angle, baffle, etc
  - Changed from symmetric arm design (6cm) to slightly asymmetric (5.3cm)
- Tolerance of radius of curvature of COC mirrors
- Surface aberration requirements and delivery check
  - Test mass and recycling cavity mirrors – see afternoon talk
- Subsystem performance simulation
  - TCS, ISC, COC, AOC, ...
- Parametric instability
  - highly distorted field, hard to be expressed by simple functions

SIS cavity system



Only coupled cavities so far

PRC - BS - X arm
PRC - BS - Y arm
SRC - BS - X arm
SRC - BS - Y arm

SIS field calculation

- FFT-based field calculation written in C++
  - » fftw as the FFT library
- Fast calculation of stationary field in a FP cavity
  - » MIT method by using two round trips to make best guess
- Fast calculation of stable recycling cavity
  - » Adaptive grid algorithm with tweaks to avoid computational round off error
- Lock using promptly reflected CR as a local oscillator
  - » Closer to real lock algorithm, but not quite
- Signal sideband generated by oscillating mirrors
- Telescope to guide large field to detector with small aperture

Using SIS
main menu of actions

Field calculation

```
lock        : Lock the cavity
calcField   : Calculate stationary field
signalGen   : Generate audio signal by sinusoidal motion of mirrors
timeTrace   : Move mirror and save field evolution
telescope   : calculate telescope outputs
```

analysis

```
delL        : Print and set the cavity length
modeAmp     : Decompose a field by LG or HG
saveField   : Save field in a file
mirrorInfo  : View mirror information
storeMap    : Store mirror maps
summary     : Print summary status
```

setting

```
simSpec     : Set simulation parameters
loadSimSpec : load simulation setup
runSpec     : Set run conditions, like convergence criteria
```

SIS
analysis tool - 1

- Random phasemap
  - » Mirror phasemaps generated whose PSD matches with a given form
  - » Simple one (PSD~f^{-2}) was used to set aLIGO test mass RMS requirement
  - » FFT – inverse FFT method used to generate random map based on actual mirror phasemaps (a.la. Bondu)
    - – Realistic requirements with realistic defects
    - – Requirements for the future interferometer – LG33 beam
- Thermal deformation
  - » Hello-Vinet formula for a cylindrical mirror built-in
- Mode expansion
  - » Resulting fields can be expanded by predefined modes
    - – Ease of understanding various effects

SIS analysis tool - 2

- Any maps (reflection, transmission, surface deformation) can be specified by data files or analytic formula or combination.
  - » HR_surface = DATAFILE(“TEM02_w60cm_N512.dat”) + 1e-9* zernikeFlat(2,2,r/r0,theta)
- Looping
  - » Repeated simulation with different parameters
    - – Calculate fields with ROC(ITM) = 1900:10:2100, ROC(ETM) = 2100:10:2300, and store locked cavity length, power, beam size on each mass
  - » Repeat for convergence search
    - – Self consistent field with thermal deformation
    - – No deformation -> field -> thermal bump -> field -> new thermal bump -> … repeat until the change of the power becomes small

Future of SIS

- Physics
  - » Simulation of Power recycled configuration
  - » Simulation of dual recycled configuration
  - » Realistic locking
  - » Auto alignment control
  - » BS thermal lens
- Computing
  - » Speed up
    - – Fast algorithm of finding stationary fields in a given cavity configuration
    - – Adaption of GPU
    - – Adaption of parallelization by taking advantage of multi-core CPU
  - » Better interface to matlab

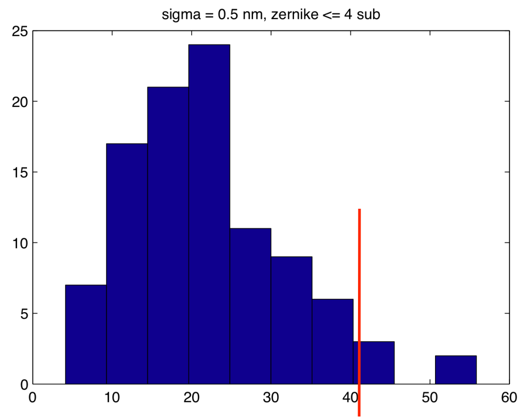
Hankel toolkit

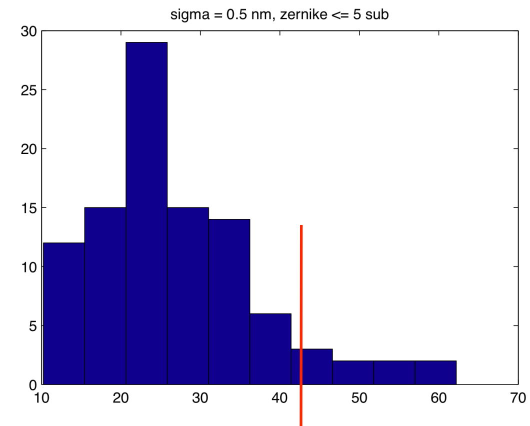
- Field calculation using Discrete Hankel Transformation
  - » Under the constraint that the system is axi-symmetric, fields can be calculated with the same accuracy as FFT-based calculation with a huge speed gain. (no need of stationary field building process, it is analytic)
- matlab class : HankelTK, CavityTK
  - » Interactions and propagation with magnification
  - » Field calculation using modal model can be directly converted to Hankel based code
- Example
  - » Round trip loss in an aLIGO arm with unbalanced absorption
  - » SIS and Hankel : 200 ppm loss for 0ppm – 1ppm absorption, 10 ppm loss for 1ppm-1ppm absorption
  - » Modal model using n+m < 20 : only qualitative relation, not quantitative

mirror rms requirement

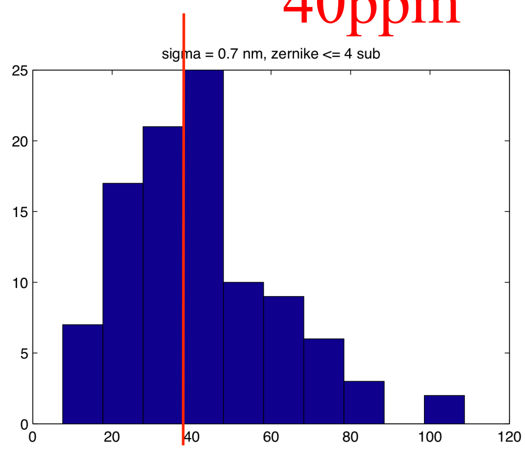
Zernike <=4 subtracted | Zernike <=5 subtracted

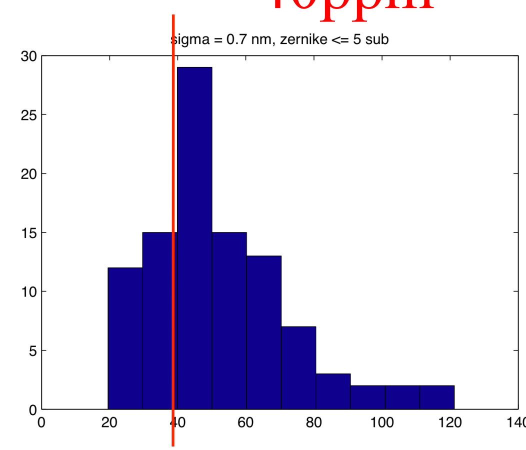
rms = 0.5nm





rms = 0.7nm





LIGO Thermoelastic bumps affect resonating mode

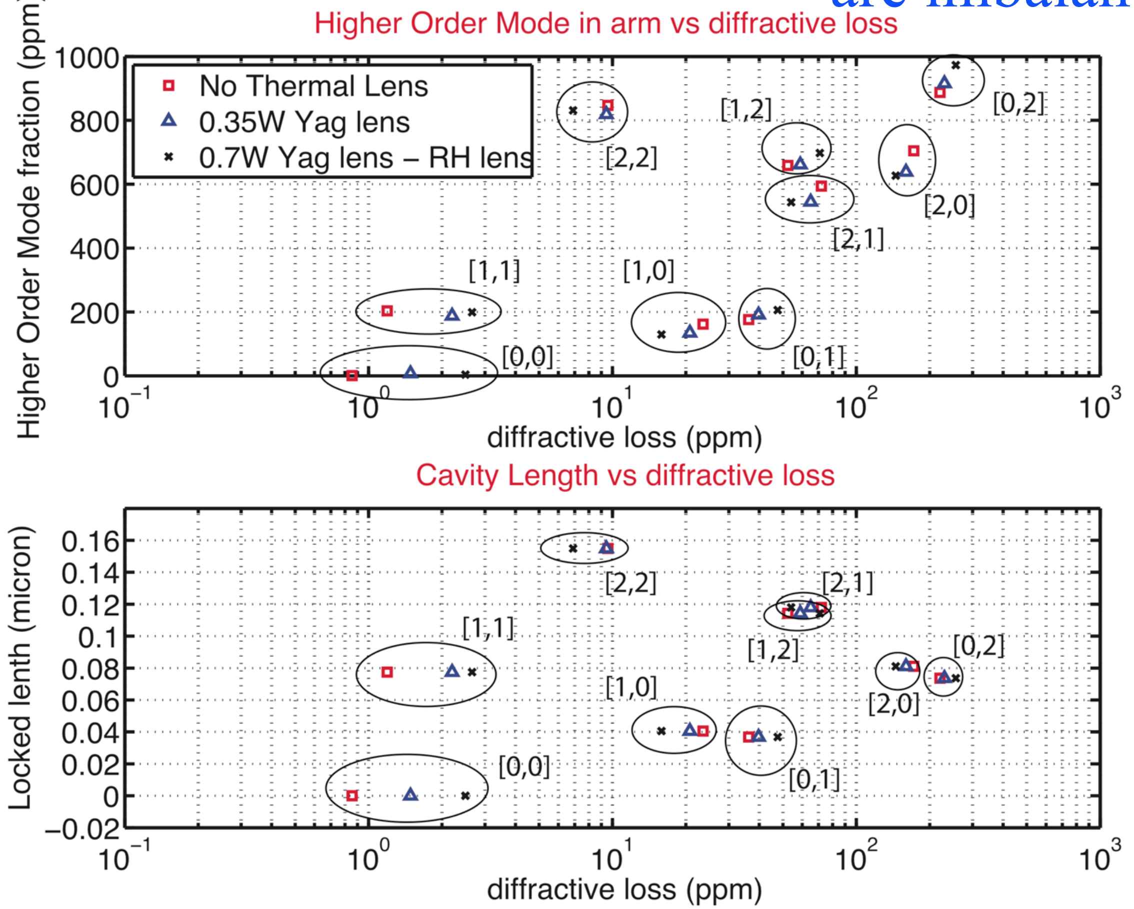
Large loss when absorptions are imbalanced

Thermoelastic

[a,b]: absorption in ppm

ITM 0.5*a

ETM 0.5*b



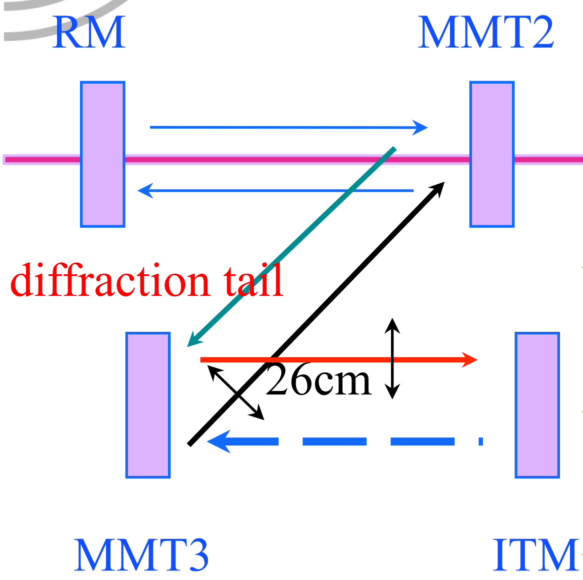
Thermal lens (ITM+CP)

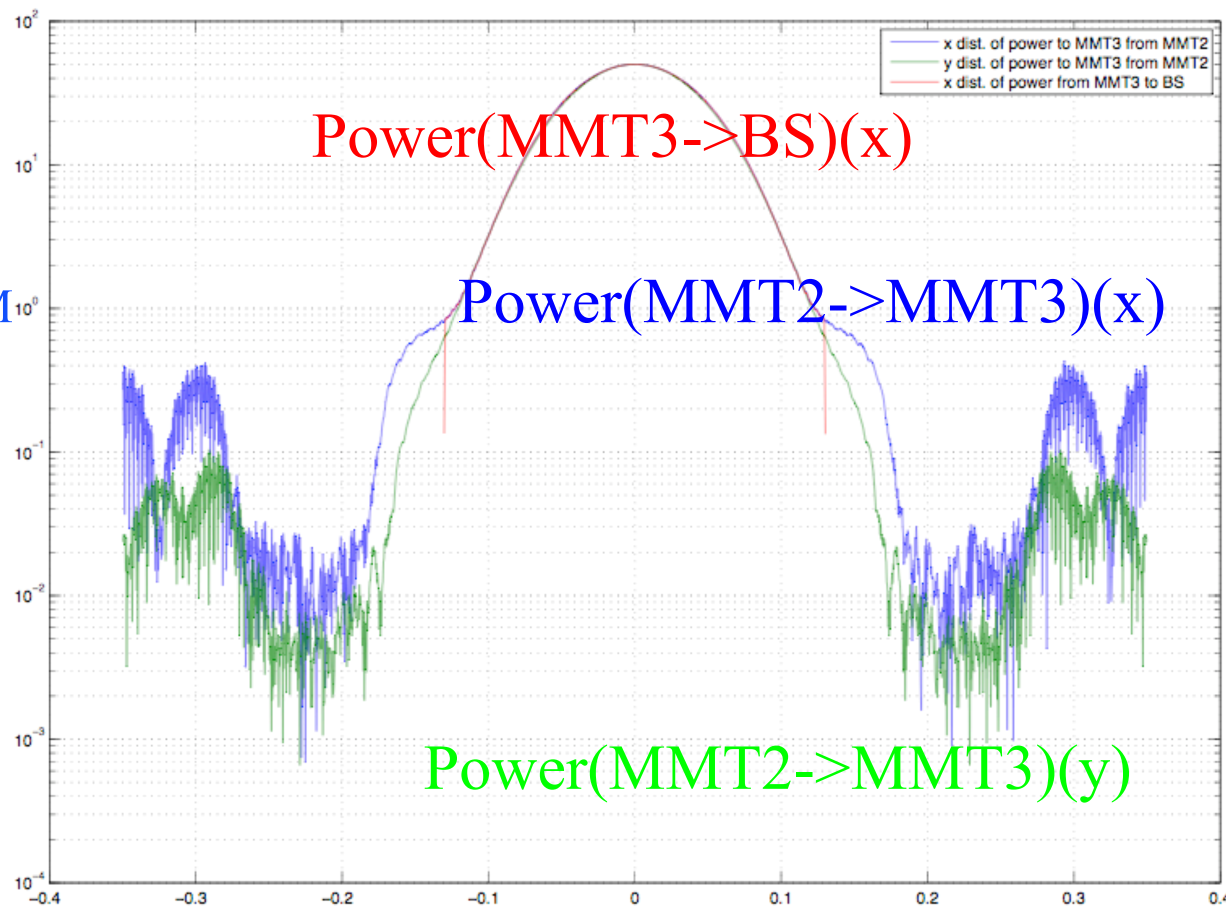
□ perfect cancelation

△ pure 0.5ppm abs

✖ 1ppm abs + RH lens

Power loss on MMT3 (ITMY<->SRM case)





loss = 330ppm
(energy outside of MMT3 surface)

Using SIS specification file

```
ITM.opt.T = 0.005
ITM.opt.R = 1 - ITM.opt.T
ITM.opt.ROC = 1971
ITM.opt.trans_phase = THERMALPHASE( beamWidth, PsubsPwr, PcoatPwr )
ITM.opt.HR_phase = THERMOELASTIC( beamWidth, PsubsPwr, PcoatPwr ) + rr/
(2*ROC_TCS) + DATAFILE( ITMMAP.dat )

ETM.opt.ROC = 2191
ETM.oscillation.amplitude = 1e-15  % 1e-9*x for rotational oscillation
ETM.opt.HR_phase = THERMOELASTIC( beamWidth, PsubsPwr, PcoatPwr ) + rr/
(2*ROC_TCS) + DATAFILE( ETMMAP.dat )

inputBeam.beamType = "LG"
inputBeam.power = 1
inputBeam.waistSize = 0
inputBeam.waistPosition = 0
inputBeam.matchToCavity = 1   % calculate waistSize and waistPosition to match
with the cold cavity
```