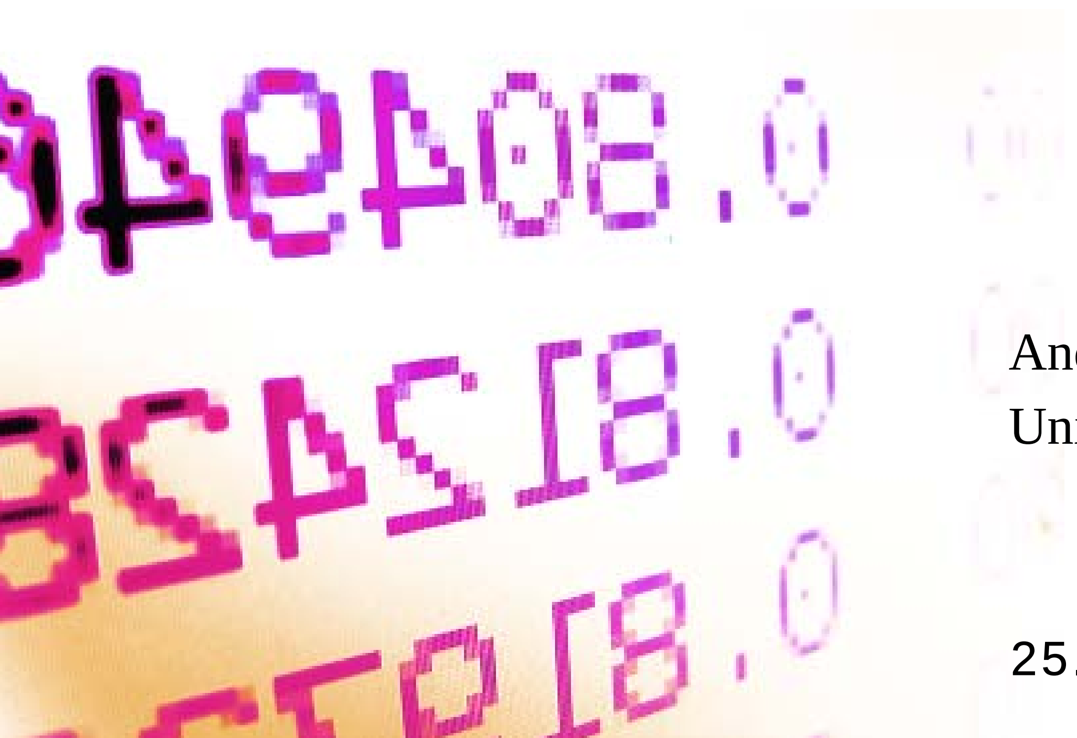
Finesse Update

+ Noise Propagation-Simulation Tutorial



Andreas Freise
University of Birmingham

25.10.2007 AEI, Hannover

LIGO-G070755-00-Z

Finesse

- General purpose interferometer simulation for laser interferometers (C code, frequency domain)
- Finesse Home, Version: 0.99.5
  http://www.rzg.mpg.de/~adf/
  - Linux, Windows, OS X binaries
  - 140 pages manual
  - Simple example files
  - Java GUI Luxor (by Jan Harms)
- GEO Simulation Wiki
  http://www.sr.bham.ac.uk/dokuwiki/doku.php?id=geosim:finesse
  - GEO 600 input file with 18 pages manual
  - External tools (Matlab interface, Beowulf cluster scripts, ...)
  - Other GW detector input files (iLigo, eLigo, advLigo, Virgo, ...)
  - Talks and tutorials

Code Changes

- Mostly changing Finesse from being a 'personal' code project to an open and manageable structure:
  - Code has been cleaned and partly re-written
  - Documentation within the code has been improved a lot (using Doxygen)
  - Code has been moved to a subversion repository and is now regularly accessed by more than one developer (You can join in, if you would like to implement a new feature in Finesse)
  - Nightly builds and tests are performed (some unit tests, mostly consistency checks against reference input files)
- Most recent main feature: client server TCP/IP communication between Finesse and Matlab (see talk from last meeting)

Matlab Interface

Finesse

Finesse in server mode:
An input file has been loaded but the 'xaxis' command is ignored - Waiting for client connection

After receiving a input value, Finesse sets the previously set Parameter(s) to that value ad computes ONE datapoint.
All outputs are computed and the Values are send back to Matlab.

(The parameter value remians At it's new value).

Matlab

katconnect(host, port)

Establishes a TCP/IP Connection

m2kat(parameterlist)

Sends parameter name(s) 'm1 phi'

Receives number of outputs (pds)

```
for i=0..100
    x=I*0.9
    out(i)=m2kat(x)
end
```

Sends numeric value for 'm1 phi'

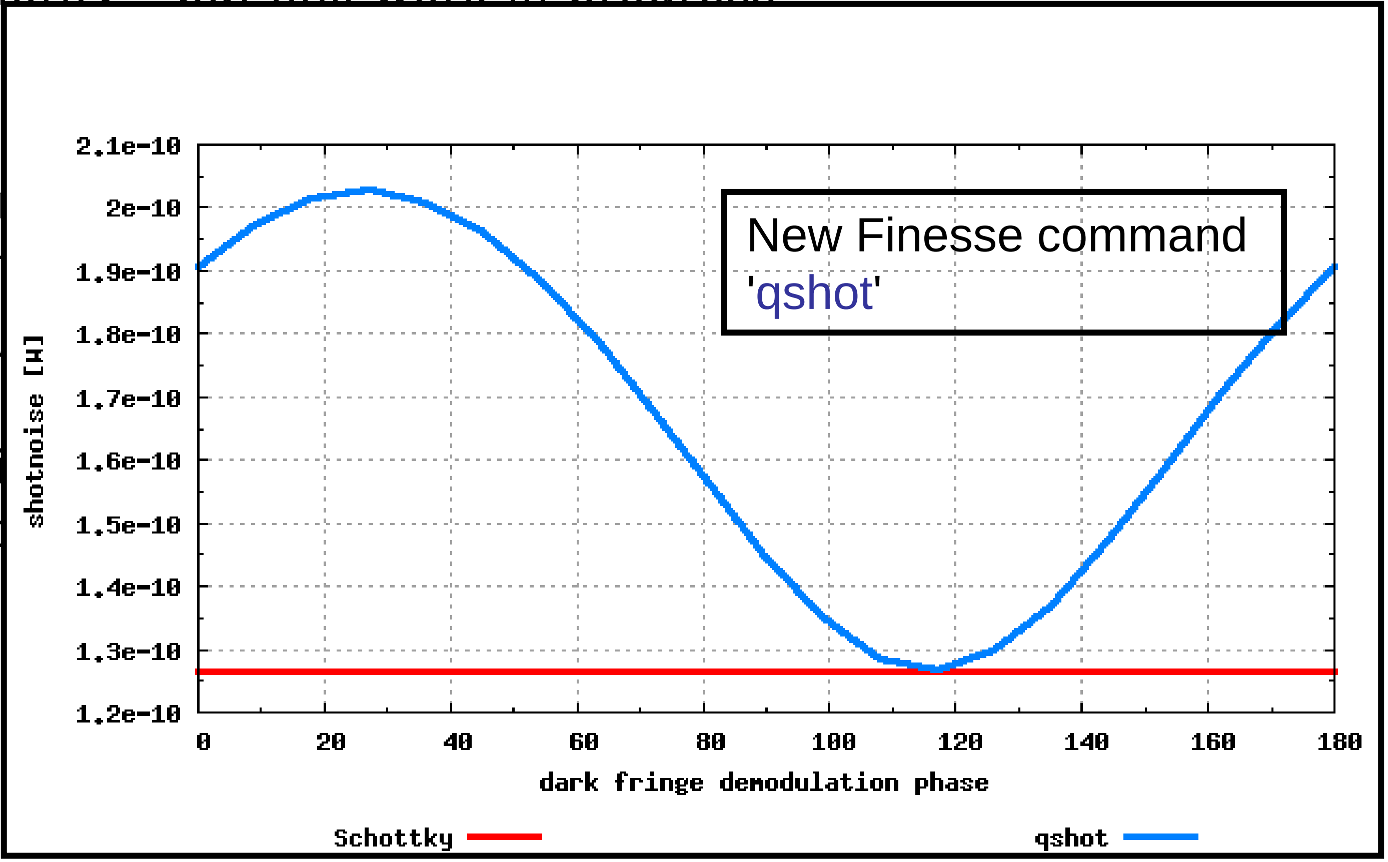
Receives values for all outputs

katdisconnect

Closing the connection

Quantum Noise, Radiation Pressure

- Highest priority - but still work in progress
  - Code h
    squeez
  - The ha
    frequer
  - Genera
    (qshot
    shotnoi
    radiatic



Status Summary

- Emphasis recently on using Finesse for GEO commissioning and providing more documentation, especially one more complex tasks
- Code changes focused on radiation pressure effects and on opening the project to new developers

Tutorial: Transfer Functions and Noise Propagations with Finesse

- Basics about computing transfer functions
- The command `fsig` and how to use it
- Doing a noise propagation from transfer functions
- The GEO 600 case

Transfer Functions

- In the frequency domain, transfer functions are computed by adding extra 'signal sidebands' to the system in the defined input and then computing their amplitudes in the desired output.

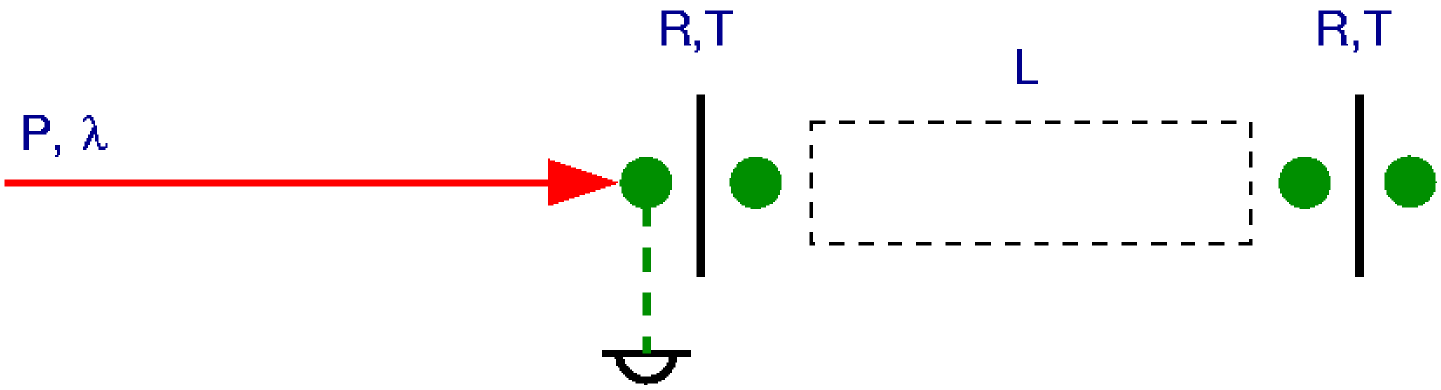
- The command

  fsig name component [type] f_s phi_s

  is used to generate these sidebands
- A photodiode with demodulation (not the amplitude detector ad) is used to detect the signal amplitude

  pd[n] name [f_{mod} phi_{mod} …] f_s phi_s

A Simple Example

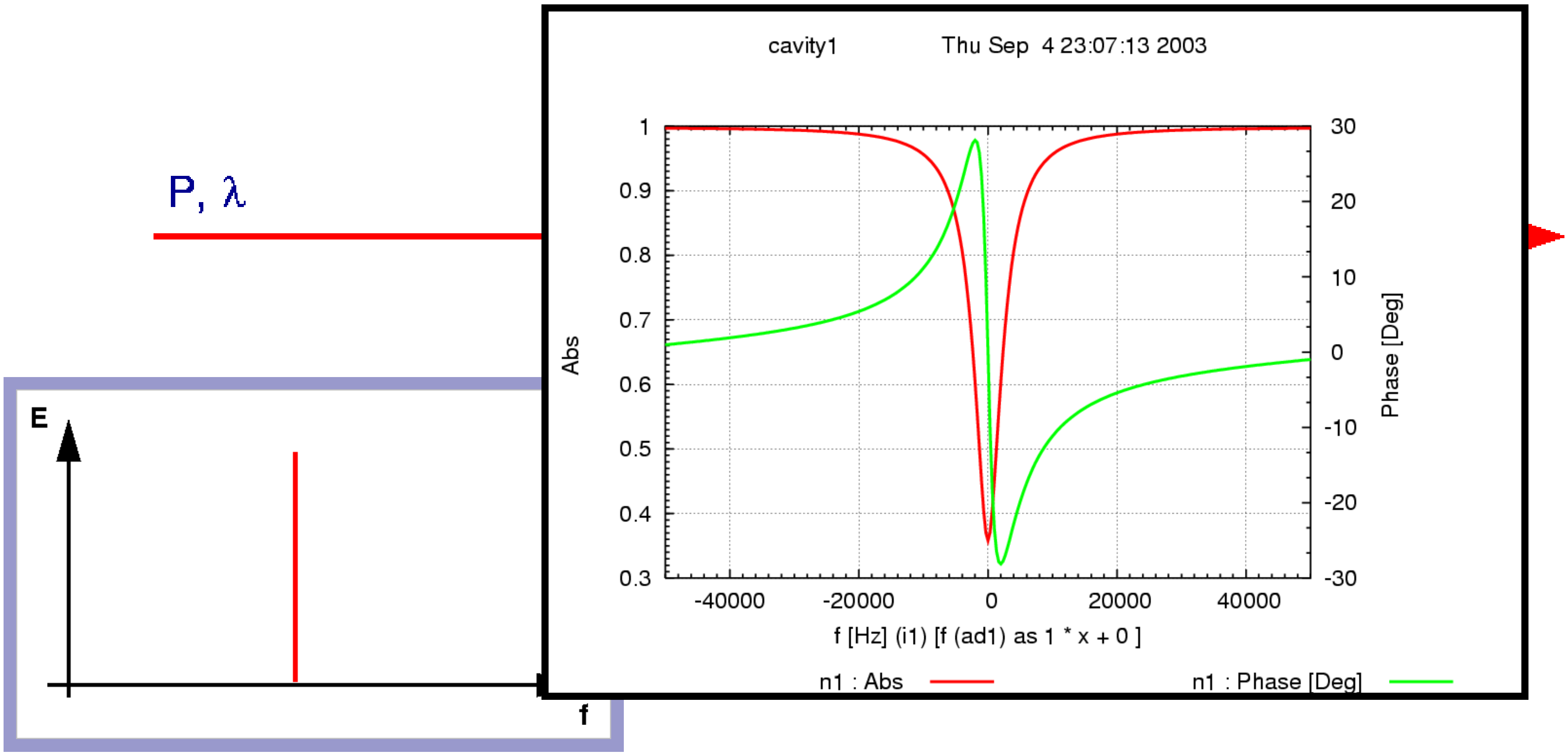


Simple cavity: two mirrors + one space (4 nodes)

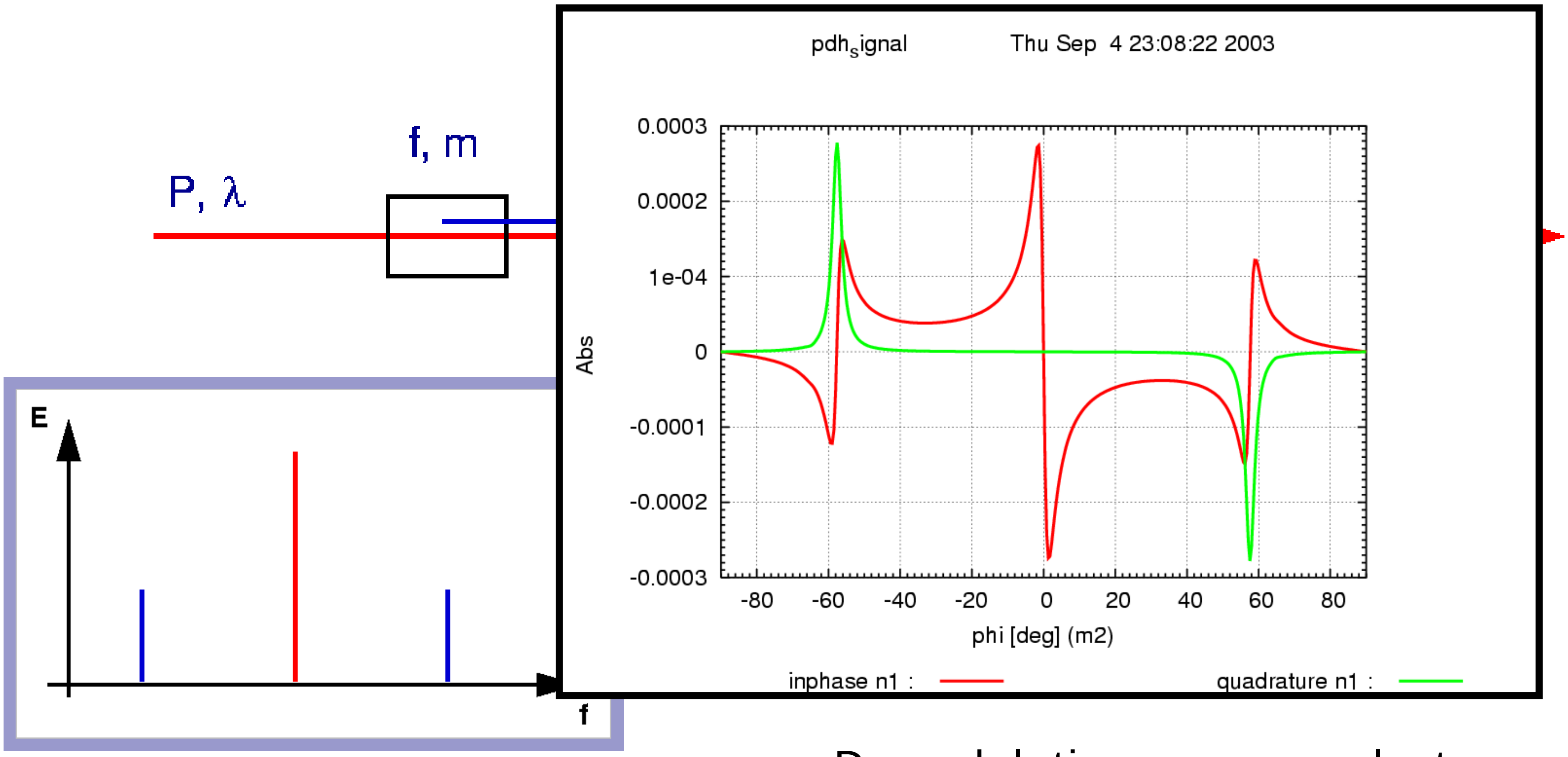
Light source (laser)

Output signal (detector)

Carrier light



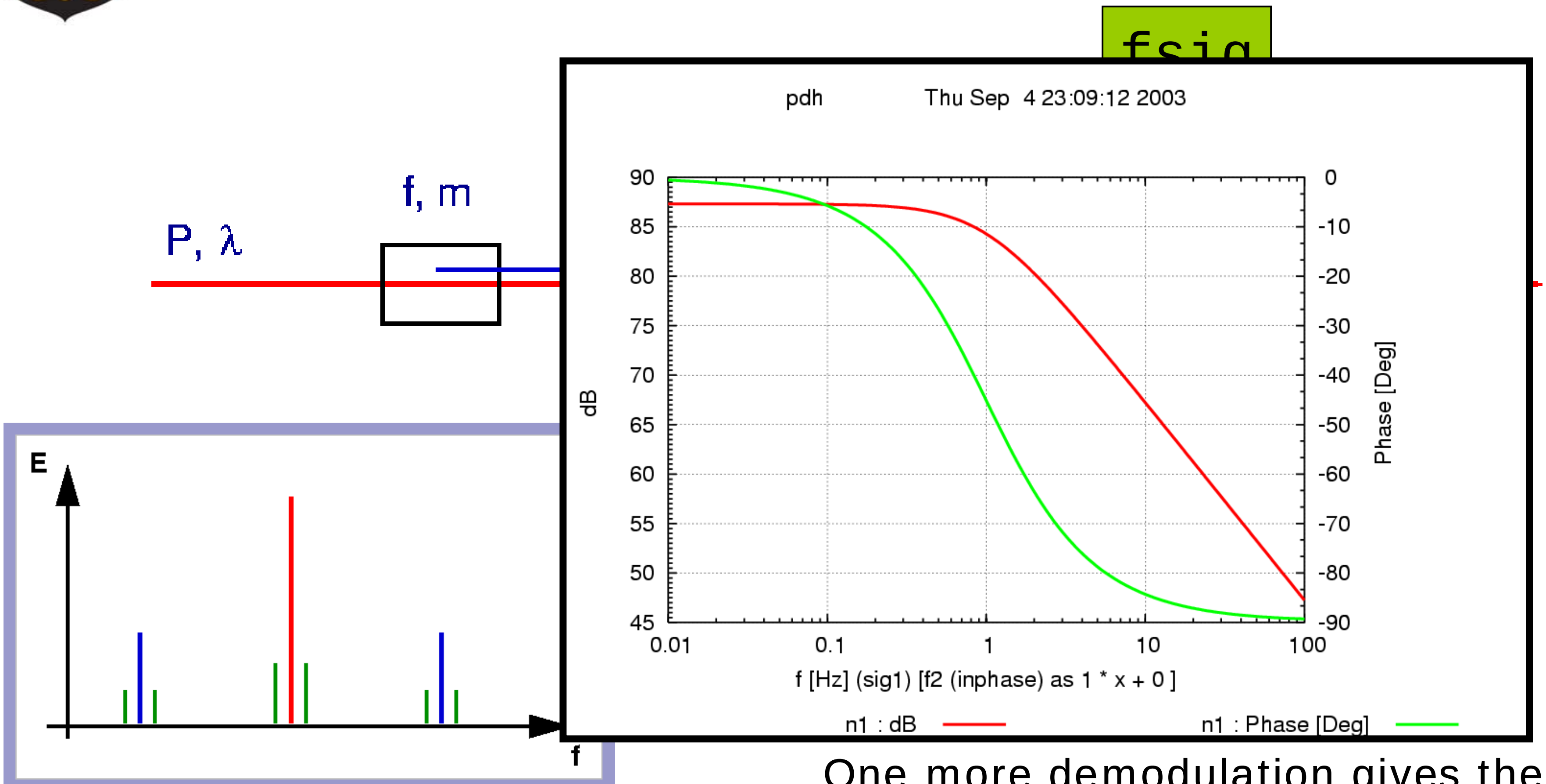
Modulation sidebands



Demodulation process selects specific beat signals

pd1 pdh f_{mod} phi_{mod} n1

Signal sidebands



One more demodulation gives the transfer function output:

pd2 pdh f_{mod} phi_{mod} f_s phi_s n1

The fsig Command

Laser component:

Type of modulation	Unit	Syntax	comment
phase	rad	`fsig sig1 phase laser f phi`	
Amplitude		`fsig sig1 amp laser f phi`	
frequency	Hz	`fsig sig1 freq laser f phi`	

(The units of the transferfunction are W/[Signal Units])

Usage:

- Note that signal sidebands added before a modulator are **not** being introduced to the modulation sidebands as well, which is **not** what happens in reality! Consequently the laser component should generally not be used with `fsig` when modulators are present (You can use a beam splitter instead, see following slides).

The `fsig` Command

Modulator component:

Type of modulation	Unit	Syntax	comment
phase	rad	`fsig sig1 eom f phi`	Oscillator phase noise
Amplitude		`fsig sig1 amp eom f phi`	Oscillator amplitude noise (currently being implemented)

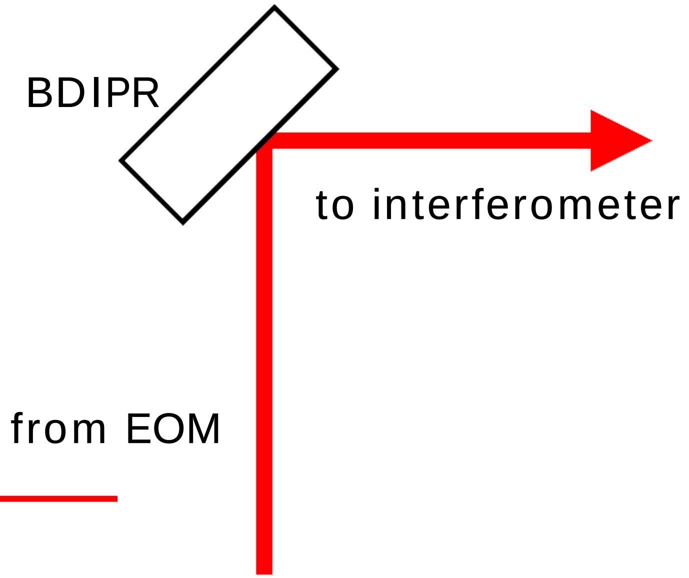
The fsig Command

Mirror or beam splitter component:

Type of modulation	Unit	Syntax	comment
phase of reflected light	rad	`fsig sig1 mirror f phi`	Convert to [m] with the command scale meter
Amplitude of reflected light		`fsig sig1 amp mirror f phi`	
Tilt of refl. light	rad	`fsig sig1 x/y mirror f phi`	Works fine but tests are not yet completed

Usage:

- Use a dummy beam splitter component (in GEO use BDIPR) for computations relative power noise (RPN) or laser frequency noise



The `fsig` Command

Space component:

Type of modulation	Unit	Syntax	comment
phase of transmitted light	(strain)	`fsig sig1 space f phi`	

Usage:

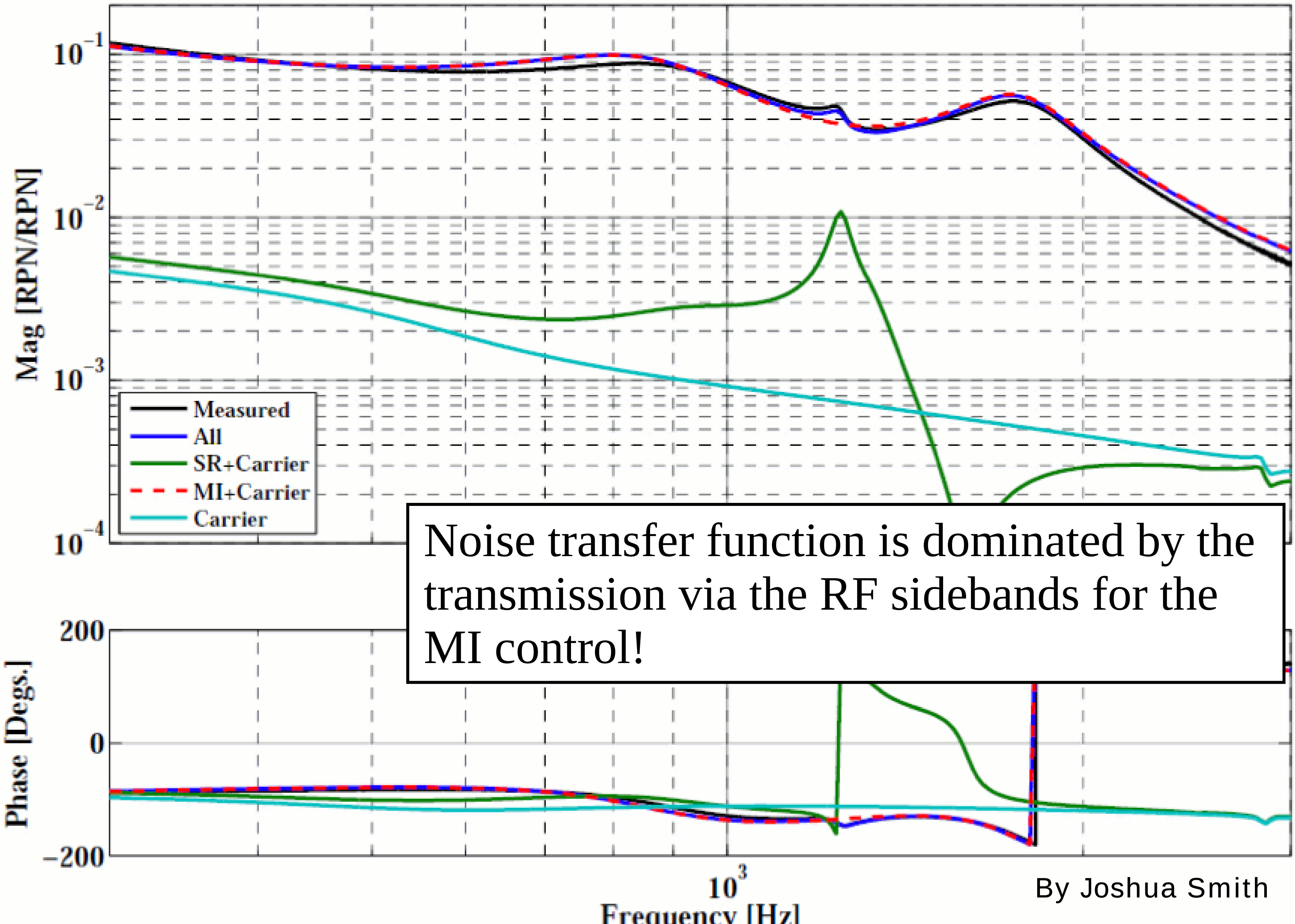
- Correctly computes the signal beyond the long-wavelength approximation in simple configurations (i.e. orthogonal arms) .

Example 1

- Detector commissioning, using the transfer function only:
  - Comparing a measured transfer function with a simulated transfer function
  - Using the GEO Finesse input file and only add:

    ```
    pd1 DPpow 1 nDPout
    fsig sig1 BDIPR amp 1 0
    xaxis sig1 f log 1 10000 1000
    put DPpow f1 $x1
    ```

    This gives the power noise transfer function into the dark port (here only with respect to the carrier light)



Noise transfer function is dominated by the transmission via the RF sidebands for the MI control!

By Joshua Smith

Example 2

- Projecting noise into the sensitivity plot:
  - Use a known or measured noise level (spectral density)
  - Compute the optical gain with Finesse (transfer function: differential end mirror motion into dark fringe)
  - Compute the apparent strain amplitude by dividing the noise spectrum by the optical gain

GEO 600 Optical Gain

- The GW signal is detected in at least two electronic signals (inphase/quadrature, P/Q of the main photodiode)
- Reconstruction of GEO sensitivity uses a complex algorithm
- We need to compute the optical gain independently for P and Q:

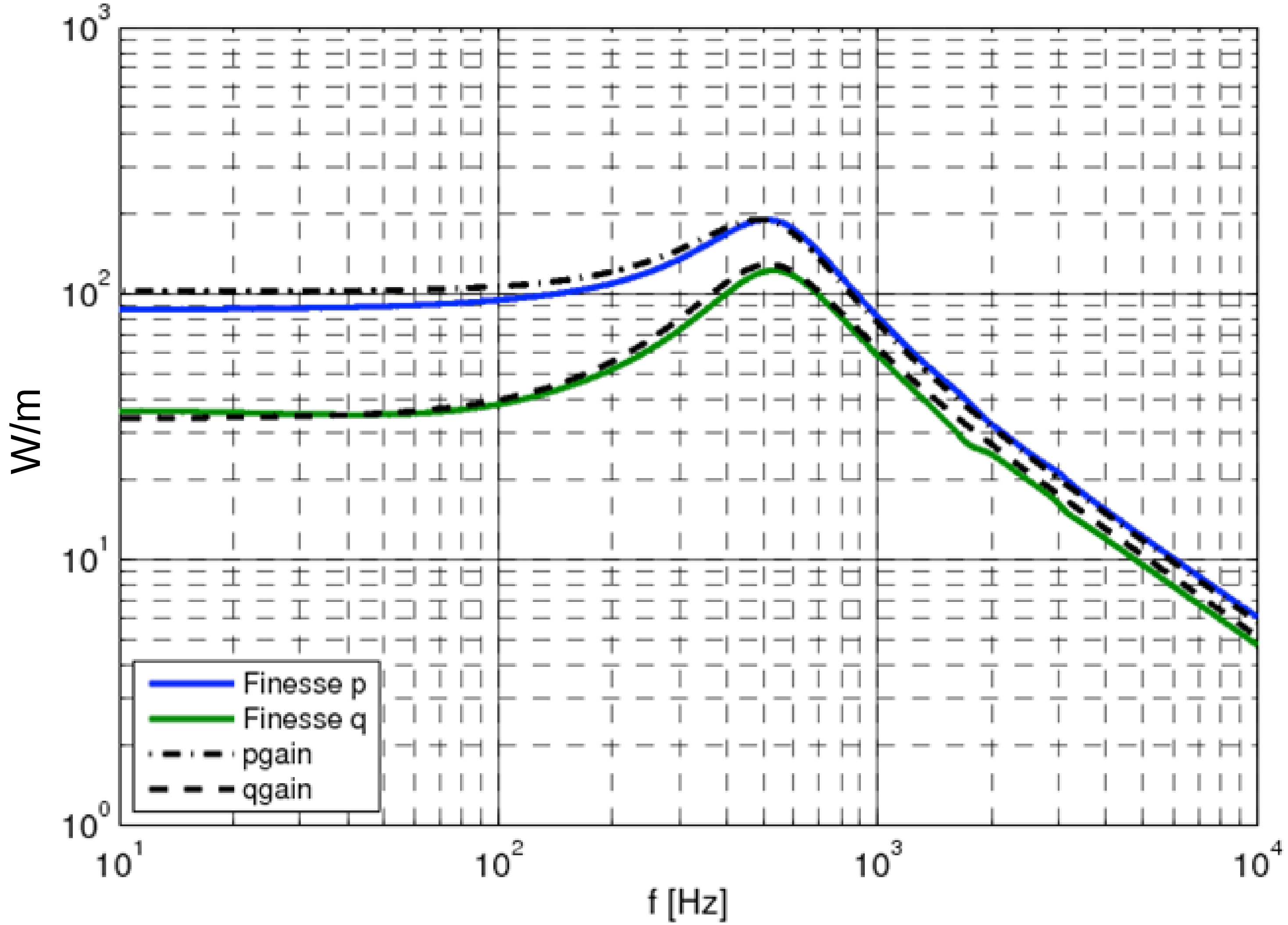
```
fsig sig1 MCN 1 0
fsig sig2 MCE 1 180
pd2 pdMI1 $fMI 4 1 nMSR2
pd2 pdMI2 $fMI 101 1 nMSR2
xaxis sig1 f log 10 10k 300
put pdMI1 f2 $x1

put pdMI2 f2 $x1
```

frequency 1 Hz, differential phase

There is always only one signal frequency!

GEO 600 Optical Gain



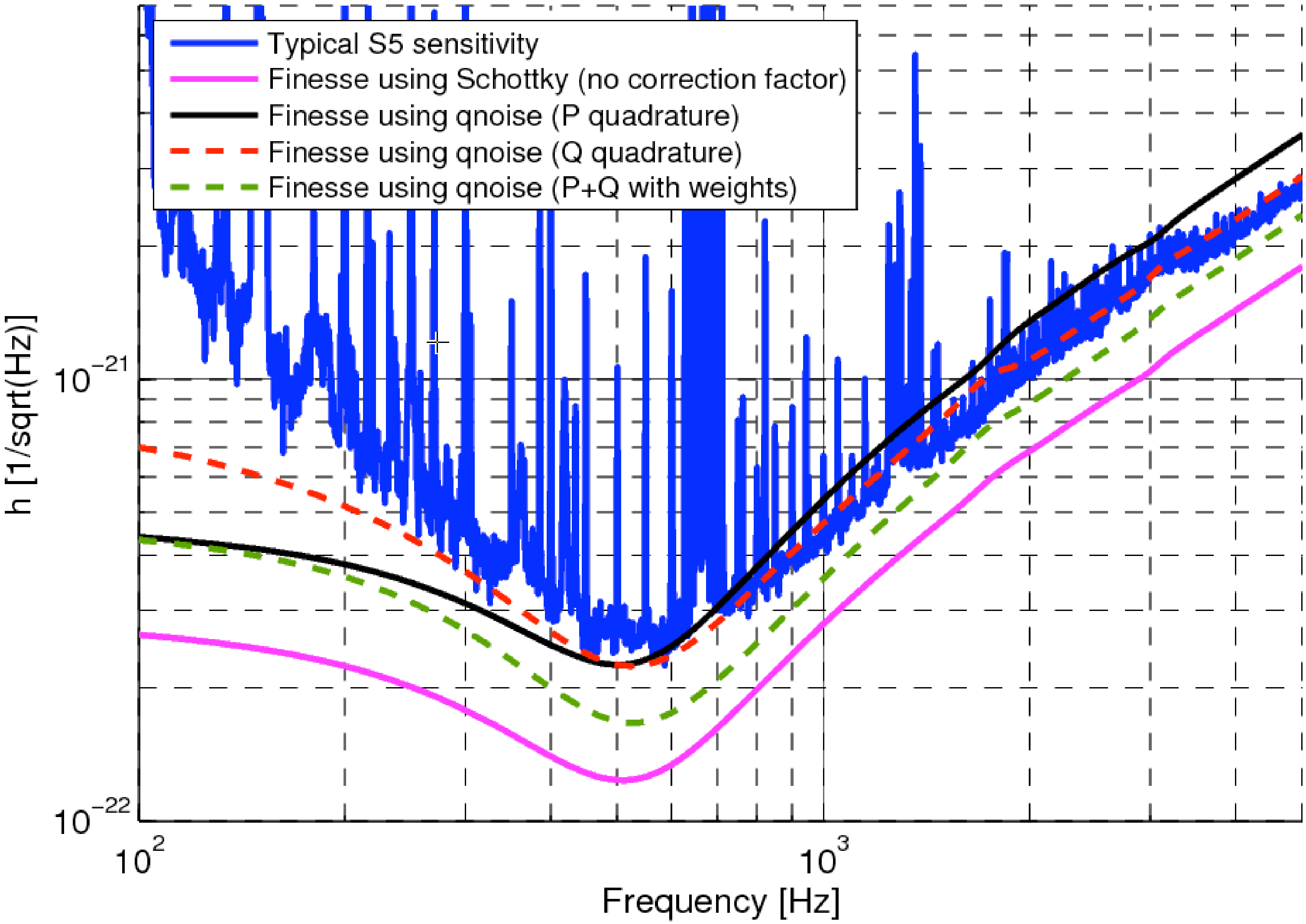
Optical Gain to Sensitivity

- Optical gain: TF in [W/m]
- Example shotnoise: We need to compute the shotnoise amplitude spectral density as S_{shot} in [W/sqrt(Hz)]
- Compute apparent displacement noise as:
  $S_{\Delta L} = S_{shot} / TF$ in [m/sqrt(Hz)]
- Or in the case of GEO: P and Q are computed separately and then merged with weighting functions:

$$S_{\Delta L} = \mathrm{sqrt}(w_p^2 S_{\Delta Lp}^2 + w_q^2 S_{\Delta Lq}^2)$$

(These computations can be done within Finesse)

GEO 600 Sensitivity



... end.

Weights for P and Q Channel

Simple approximation of weighting functions: