Project tasks

Welcome to the Hackathon projects!

Each team should choose **either** project option A or B.

In both cases, you will create a new gravitational wave detector design, either by optimizing parameters of the LIGO model to meet or by using a set of provided parameters to construct a new design entirely.

You will then produce a noise budget, before exploring a choice of more advanced behaviours in the detector.

Unlike previous tasks, the projects are more exploratory and open ended, so there might not be a single ‘correct’ answer. We instead want to see how your skills with Finesse and your knowledge of optics and gravitational wave detectors have grown, and for you to be able to explain the thought process and motivation behind your design decisions.

Option A: The ‘Third Generation’

The current generation of detectors (LIGO, Virgo) are considered the second generation of gravitational wave detectors: ‘Advanced’ detectors that delivered a wide range of improvements to the existing sites after the initial detectors were completed. Common features of the second generation include having arm lengths of a few km (3km at Virgo, 4km for LIGO), room temperature optics weighing tens of kg (e.g. 40kg at LIGO), and the DRFPMi optical configuration. These detectors are continuing to be upgraded - for example by implementing frequency-dependent squeezing and homodyne readout.
In the future, new detectors are planned, like the Einstein Telescope in Europe and Cosmic Explorer in the USA, which are intended to again significantly improve on the current generation. Common features of these ‘third generation’ detectors include using arm lengths of *tens* of km, new optical materials, cryogenically cooling the optics, which would weigh up to several *hundred* kg, and several possible changes from the DRFPMi configuration.

Your task is to develop a 10km DRFPMi model with parameters as specified in the table below. A (slightly) simplified version of the advanced LIGO Finesse file is provided as a reference. Build up your own model, using what you learned in the earlier tasks to mode match the PRC and (identical) SRC to the arm cavities. You might find this guide helpful. As you saw in task 2, it is not possible to have a well mode-matched PRC, that is geometrically stable and has high gain, with the simple three-mirror cavity setup used there. To solve this, LIGO uses a 2-mirror telescope in the recycling cavities to ensure the configuration is geometrically stable. The parameters above instead propose using 303m lenses immediately before the ITMs, making your task significantly easier!

Once you have constructed your interferometer design and mode-matched it, complete the *noise budget* task in the *Common tasks for all projects* section below.

In your remaining time, you may choose to explore some of the *Advanced Topic Ideas* below, or pick any other topic(s) of interest to your team*.

*If you pick the latter, we strongly recommend that you discuss your plans with the mentors first (please copy all four in your communications)!.

General Parameters	
Input power	500 W
Arm Circulating Power	~3 MW
Laser wavelength	1064 nm
Optical material	Fused silica
Loss per optical surface	37.5 ppm
Operational mode	Broadband (RSE)
DC power at dark port (needed for DC readout, set by [de]tuning the ETMs)	~10 uW
Arm Cavities	
Arm length	10 km
Arm cavity geometric stability	m=0.147
ITM Transmission	7000 ppm
ETM transmission	6 ppm
TM Mass	200 kg
TM thickness	30 cm
Central Interferometer	
Total length of each recycling cavity	310 m
Distance from ITMs to Beamsplitter	300 m
Beamsplitter properties	Same as aLIGO
Focal length of lens directly before ITM	303 m
Power recycling gain	21.6
Round Trip Gouy Phase in Recycling Cavities	~35°
PRM transmission	4.6 %
SRM transmission	10 %

Option B: ‘What do we do with LIGO next?’

Once the next generation of detectors are under development, one may ask “what do we do with the old sites?” One exciting option is to use them to target particular frequency ranges, rather than the broadband operational mode used by all current detectors. The parameters of the detectors could be adjusted so that the LIGO sites are extremely sensitive to kHz frequencies, which are of particular interest for probing neutron star physics. This provides an opportunity to observe a wide array of physics in an extreme environment, under conditions that would be near-impossible to recreate on Earth. The neutron star equation of state depends on many aspects of fundamental physics and how these interact with one another (e.g. General Relativity and Quantum Mechanics), making neutron star binary collisions an important testing ground for these theories.

Your task is to adapt the provided LIGO model such that it is maximally sensitive to gravitational wave frequencies in the range <u>1-4kHz</u>. An example of the quantitative metric to use is Equation 8 of this paper - but you should feel free to simplify it (e.g. assume the antenna pattern is isotropic).

You should assume that only limited funding is available to improve upon the existing components (i.e. we can’t build a whole new facility), so you should limit your search to:

- Changes in the tuning of the optics
- Changes to the transmissivities of the optics (NB: assume the optical losses cannot be altered from the values provided in the LIGO model.
- Injecting up to 3dB frequency-independent squeezing and selecting an appropriate squeezing angle

Once you have completed the modifications to the LIGO model, complete the *noise budget* task in the *Common tasks for all projects* section below. In your remaining time, you may choose to explore some of the *Advanced Topic Ideas* below or pick any other topic(s) of interest to your team*.

*If you pick the latter, we strongly recommend that you discuss your plans with the mentors first (please copy all four in your communications)!

Common tasks for all projects

Noise Budget

Every group should provide:

- A quantum noise limited sensitivity curve for the new detector, showing how each major parameter change (compared to the original LIGO model) has changed/improved the detector's sensitivity to gravitational waves. Explain physically why these effects occur and why these design decisions were made.
- On a separate plot, a noise budget which considers:
    - The quantum noise limited sensitivity
    - Laser frequency noise
    - Seismic noise
    - Coating thermal noise.
      For the coating thermal noise, given the subtleties in layer structure optimization etc, you may assume the following empirical ASD for the quadrature sum of four optics:
      $\sqrt{S_{CoatingBrownian}}(f) = \frac{w_{LIGO}}{w_{new}} \frac{7 \times 10^{-20}}{\sqrt{f}} \frac{m}{\sqrt{Hz}}$, where w_{new} is the beam size on the ETM in your design, and w_{LIGO} is the beam size on the Advanced LIGO ETM.
    - Any other noise sources you think you can reasonably estimate from your detector parameters

  Indicating at what frequency range each noise source is most important, by what factor they would need to be suppressed so that the detector is quantum-limited at all frequencies, and qualitatively suggest methods to achieve this suppression.
- A qualitative description of the kinds of gravitational wave sources the new detector is likely to be better or worse at observing that LIGO.

Advanced Topic Ideas

Below are some topics you might like to explore once you have completed both the tasks specific to your project and the common task listed above. You may choose to spend any extra time going through all stages of one of these, complete the first few steps of each, or do something else entirely.

Linear controls

For project option A:

Develop a control scheme for the linear degrees of freedom of your new detector. By emulating the LIGO scheme, plot the DARM error signal, and calculate a new modulation frequency that can be used to get error signals for the CARM and PRCL degrees of freedom in your new interferometer design (recall task 1 from the first day). Confirm that the new sidebands can be used to give independent error signals that can distinguish between these degrees of freedom. How do these error signals compare to those in the LIGO model? Why?

For project option B:

Compare the control signals for the linear degrees of freedom in your model to those used in the original LIGO file. How do the differences in these signals affect our ability to control the detector? How do each of the parameter changes you have made influence the shape of the error signals, and why? Qualitatively, what kind of changes would need to be made to the feedback loops so that the interferometer stays locked with all optics at the correct tunings? Attempt to lock the interferometer using lock, put, and func commands.

Higher order mode effects

When optics are misaligned, or the interferometer contains mode mismatches, light is ‘scattered’ from the carrier 00 mode into higher order modes. Light can also sometimes be scattered from higher order modes back into the carrier.

Plot the power in the optical modes up to 4th order at the dark port. Study how this is affected by misaligned optics, or mismatches between cavities in the interferometer. What kind of mismatches/misalignments affect how ‘dark’ the dark port really is?

The gravitational wave signal is extracted from the HG00 mode, which in the real detectors is separated from the control sidebands and any higher order mode content using an ‘output mode cleaner’. How do misalignment and mode mismatch affect the quantum-noise-limited sensitivity?

Misalignments and mismatches can also affect how effectively some noise sources are suppressed by the interferometer. Explore how laser frequency noise is affected by mode mismatch. Why does this happen?

Plot the SRCL error signal for your interferometer model. What happens to this when optics are misaligned? How might this affect our ability to lock SRCL? Why does this happen?

Frequency-Dependent Squeezing

In the very near future, LIGO will implement *frequency-dependent* squeezing. In this scheme, frequency-independent squeezed light from a squeezer is first reflected from a new “filter cavity”, before being injected into the interferometer as normal. This filter cavity causes the squeezing angle of the injected light to be frequency-dependent. With careful choices of parameters, this can be used to improve the quantum-limited sensitivity of a detector across the *whole* gravitational wave signal frequency range.

The following equations give the optimal filter cavity parameters for an aLIGO-like setup:

$$\Delta\omega_{fc} = \gamma_{fc}$$

$$\gamma_{fc} = \frac{\Omega_{SQL}}{2}$$

$$\Omega_{SQL} \approx \frac{t_{srm}}{1 + r_{srm}} \frac{8}{c} \sqrt{\frac{P_{arm}\omega_0}{mT_{itm}}}$$

Here, $\Delta\omega_{fc}$ is the detuning of the filter cavity, and γ_{fc} is the half-bandwidth of the filter cavity, both in units of $2\pi \times$ Hz. See the extra information provided on the following pages for further detail. PyKat includes a function to do this calculation for you, and return the required cavity detuning (in degrees) and input mirror transmissivity, ready to be put into Finesse — this is the function `filter_cavity_parameters` under `pykat.tools.filter_cavity`. (Remember you can run `help(function_name)` to get the documentation for that function)

Starting with a plane waves model, add a filter cavity with the correct parameters to inject 6dB of frequency-dependent squeezing into your interferometer. How do the parameters affect the sensitivity curve? Is this beneficial to your project goal? How?

Attempt to mode-match your filter cavity to the interferometer: set the cavity mirror curvatures to produce a stable eigenmode, and if needed add additional spaces and lenses between the cavity and interferometer. What happens to the detector sensitivity if the mode matching is bad? Why?

Information you might find useful…

In this section we've collected together some pieces of information and references that might come in handy, depending on what you decide to explore in your projects.

Loading in an external Finesse file

You have been provided with a Finesse script to use as a reference/starting point for your work. To load this into pykat and begin working with it as we usually do, you just need the 'load' command:

```
base = finesse.kat()
base.load("myfile.kat")
print(base)
```

You can also save scripts that you have created:

```
base.save("myfinessescript.kat")
```

Finding your way around the LIGO file

The LIGO file provided is quite long compared to the simulations we've explored so far! It also includes a couple of features you might not have encountered yet.

We strongly recommend that you make a by-hand sketch of the optical configuration, noting the names of components and nodes, so that you can quickly identify which optics are the "main" ones (e.g. the PRM, ITM, beamsplitter) vs ones that are, for example, part of a composite optic. Try comparing your sketch to a more general sketch of LIGO (E.g. this one) to see what elements we've chosen to include, and using the node graph tool (see below) to help you out.

Blocks

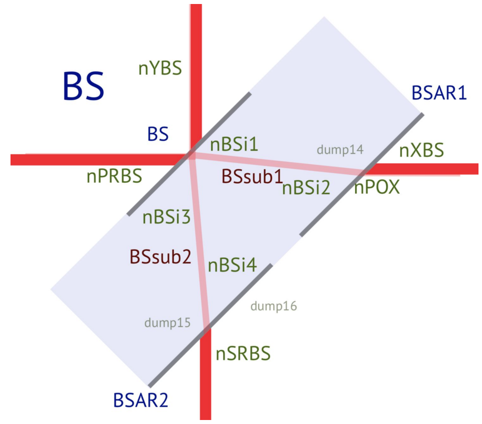
The provided file includes 'blocks' marked e.g. `%%% FTblock Xarm` / `%%% FTend Xarm`. These are purely for convenience: they separate the code into convenient chunks, making it easier to read, and can be used to selectively add/remove each of those chunks quickly using pykat.
For example, if you do not need to use the error signals while running a particular simulation, you could either remove them individually using something like `kat.AS_DC.remove()`, or remove the whole block together using `kat.removeBlock('errsigs')`. Other handy pykat features:

- List all block names: `kat.getBlocks()`
- Print just one block: `kat.getBlockString('errsigs')`

Composite optics

By default, optics are treated as infinitely thin reflective planes. This isn't very realistic, but sometimes doesn't affect the behaviour of a model much. Other times, for example when considering the relative behaviour of different fields propagating along an optical path, we need to take the thickness of the optic into account. This results in *composite optics*: a single component, modelled using multiple finesse commands. For example, the beamsplitter in the LIGO file looks like this:

```
%%% FTblock BS
bs1 BS 0.5 3.75e-05 0.0 45.0 nPRBS nYBS nBSi1 nBSi3
s BSsub1 0.0687 1.44963098985906 nBSi1 nBSi2
s BSsub2 0.0687 1.44963098985906 nBSi3 nBSi4
bs2 BSAR1 5e-05 0.0 0.0 -29.195 nBSi2 dump14 nXBS nPOX
bs2 BSAR2 5e-05 0.0 0.0 29.195 nBSi4 dump15 nSRBS dump16
%%% FTend BS
```



Usually, one surface will be the main reflective surface of the optic (i.e. the surface we actually want to have in our detector), while the other surface has an antireflective coating designed to have minimal impact on the optical field.

Debugging Finesse input files

- Pykat comes with a handy tool to generate a *node graph* of your model. This shows optics, components and detectors, nodes and the links connecting them, and can help with debugging when something in your file doesn't work. We briefly covered this in the workshop and there is more information in [this post](#) on the sandbox. This node graph may help you to identify misconnections.
- Try adding power or amplitude detectors at various nodes to check their output makes sense.
- If your complicated model isn't working, try commenting out chunks of your input file until it makes sense, then gradually add components back in. For example, you can check your arm cavity finesse, power build-up, transfer function, etc. by connecting the input laser directly to the arm cavity input test mass and commenting out everything else (e.g. the second arm, the recycling cavities, beam splitter, etc.).

Mode matching a coupled cavity

- Setting curvatures in a single cavity, geometric stability were covered Tasks 2 and 5 from the [Gaussian mentoring notebook](#)
- Plane waves coupled cavity:
  - Quirk: finesse's i-on-transmission means by default 3-mirror cavities are anti-resonant, you need to add 90deg to detuning to have sensible results.
- Mode matched coupled cavity: task 2 from yesterday.

Control signals for a coupled cavity (power-recycled arm)

- Controlling a single cavity: [Workshop notebook 9 (Locking and Control)](#); [Finesse Manual appendix A](#)
- [40m prototype design doc](#)

- LIGO linear sensing and control design: LIGO-T1000298
- How control frequencies were chosen for LIGO: Input Optics Subsystem Preliminary Design Document (LIGO-T060269) [includes HOM considerations]

Frequency-dependent Squeezing

- For a full description of the analytical formula for filter cavity parameters, see Decoherence and degradation of squeezed states in quantum filter cavities. Be warned though - the description is spread out across many separate equations and sections!

Noise Budgets

- In reality there are hundreds of noise sources contributing to the overall sensitivity of detectors, but those you are asked to calculate are some of the more important ones. Some of these noise sources are not well known, and highly site-dependent (e.g. seismic and Newtonian noise), but you should be able to make reasonable guesses based on your detector parameters.
- We covered some simple noise sources and budgeting in the workshop in notebook 13 (Mich signals and noise)
- SpacePyQuest: a toy noise calculator for gravitational wave detectors. Use this only as a guide for how noise sources scale with frequency, but not as a rigorous calculator (see document attached to hackathon page).
- Pygwinc: a (beta version) noise calculator for Advanced LIGO-like instruments.
- A reasonably up-to-date noise budget for Advanced LIGO can be found in e.g. Martynov et al.
- Try not to find yourself lost in a long calculation for e.g. coating thermal noise. One alternative approach would be to understand how the noise scales with your detector parameters. You could for example fit a curve through a noise source shown in a noise budget for Advanced LIGO and scale it for your own parameters.

LIGO’s linear control scheme

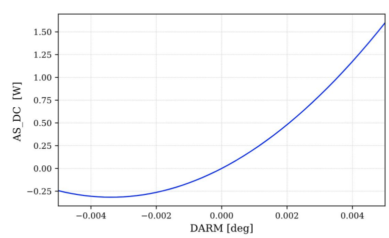
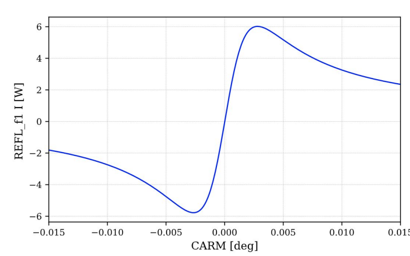
The 40m prototype design doc and LIGO-T1000298 provide a nice overview of how an interferometer can be set up with sidebands that can be used to control the many degrees of freedom. Here’s a summary of how this is implemented in practice at LIGO:

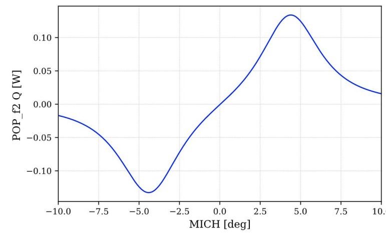
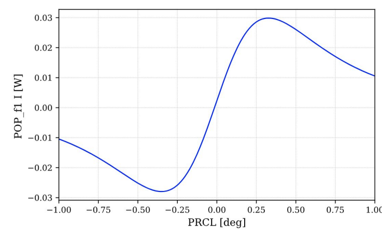
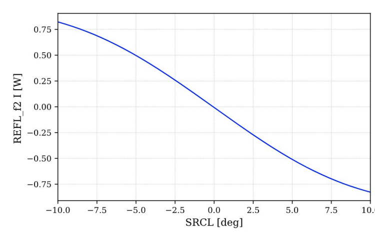
The graphic LIGO-G1601619 shows where all the optical signals are measured at LIGO. This includes:

- ordinary DC photodiodes measuring the power in the field
- quadrant photodiodes and wavefront sensors, generally used for measuring and controlling misalignment
- Hartman sensors, used for understanding beam distortions, and
- *RF photodiodes*, used to measure the linear control signals.

Compare the locations of the RF photodiodes to the pd1 detectors included in the aLIGO finesse file.

The detectors are used to measure error signals for each linear degree of freedom as follows:

DOF	**Location measured**	**Frequency**	**Sensor name in Finesse**
DARM (ETMs in antiphase)	Interferometer output (Dark port)	DC	AS_DC 
CARM (ETMs in phase)	REFL	9MHz	REFL_f1_I 
MICH (Whole arms in antiphase, i.e. +ETMX, +ITMX, -ETMY, -ITMY)	POP	45MHz	POP_f2_Q

<table>
<tr><td></td><td></td><td></td><td>
</td></tr>
<tr><td>PRCL
(PRM)</td><td>POP</td><td>9MHz</td><td>POP_f1_I

</td></tr>
<tr><td>SRCL
(SRM)</td><td>REFL</td><td>45MHz</td><td>REFL_f2_I

</td></tr>
</table>