1 Michelson interferometer differential arm response

[1]:

```
import numpy as np
import matplotlib.pyplot as plt
import os
from ifo_configs import mich_freq_resp as MICH
from ifo_configs import N_shot
from ifo_configs import bode_amp, bode_ph
%matplotlib inline
plt_style_dir = 'stash/'
if os.path.isdir(plt_style_dir) == True:
    plt.style.use(plt_style_dir + 'ppt2latexsubfig.mplstyle')
plt.rcParams["font.family"] = "Times New Roman"
```

[2]:

```
# Some parameters
cee = np.float64(299792458)
h_bar = (6.626e-34)/(2*np.pi)
OMEG = np.float64(2*np.pi*cee/(1064.0*1e-9))
L = np.float64(4000.0)
nu = np.arange(1, 1000000, 1)
PHI_0 = np.pi/2 #[rad]
P_IN = 125 #[W]
```

1.1 Derivation*

For the simple Michelson we know that a change in arm length correlates to light at the AS port We also know that a differential arm length corresponds to a difference in phase of the light that impinges upon the BS For a gravitational wave we can quantify the phase difference in this following way:

$$\phi_A - \phi_B = \int_{t-2L/c}^{t} \Omega\left[1 + \frac{1}{2}h(t)\right]dt - \int_{t-2L/c}^{t} \Omega\left[1 - \frac{1}{2}h(t)\right]dt \tag{1}$$

The phase difference can then be quantified by:

$$\phi_A - \phi_B = \int_{t-2L/c}^{t} \Omega h(t) dt \tag{2}$$

where

$$h(t) = h_0 e^{i\omega t} \tag{3}$$

Ω is the **optical angular frequency**

After evaluating this integral we get:

$$\Delta\phi = \phi_A - \phi_B = \frac{2L\Omega}{c} e^{-iL\omega/c} \frac{\sin(L\omega/c)}{L\omega/c} \cdot h_0 e^{i\omega t}$$

Where the first term in the phase difference carries all the time independent frequency information. This is what we are calculating below.

For the sake of being explicit, we are going to plot:

$$\Delta\phi(\omega) = h_0 \frac{2L\Omega}{c} e^{-iL\omega/c} \frac{\sin(L\omega/c)}{L\omega/c}$$

This accounts for the differential phase as a function of gravitational wave frequency, though we have not established the amount of optical gain the Michelson offers. This can be understood through a first order taylor approximation about a selected Michelson offset angle ϕ_0:

$$P(\omega, \phi_0) = \frac{P_{\text{in}}}{4}[r_x^2 + r_y^2 - 2r_x r_y \cos(\phi_0 + \Delta\phi(\omega)]$$

$$P(\omega, \phi_0) \approx \frac{P_{\text{in}}}{4}\Big[r_x^2 + r_y^2 - 2r_x r_y \big(\cos(\phi_0) - \Delta\phi(\omega)\cdot\sin(\phi_0)\big)\Big] = \frac{P_{\text{in}}}{2}\Big[1 - \big(\cos(\phi_0) - \Delta\phi(\omega)\cdot\sin(\phi_0)\big)\Big]$$

Where we define a response gain function H_{MICH}:

$$\mathrm{H}_{\text{MICH}}(\omega, \phi_0) = \frac{P_{\text{in}}}{2} \cdot \Delta\phi(\omega) \cdot \sin(\phi_0)$$

[3]:

```
MICH?
```

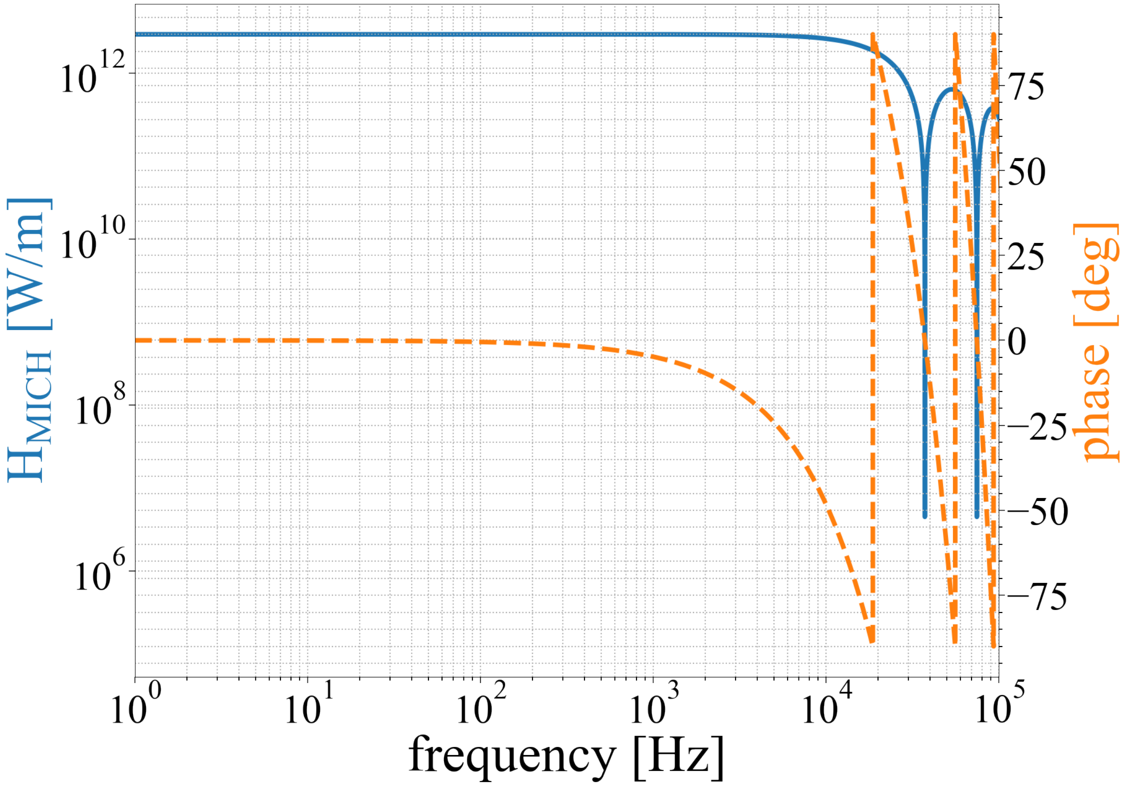
```
Signature: MICH(freq, Length, phi_0, P_in, OMEGA)
Docstring:
MICHELSON FREQEUNCY RESPONSE CALCULATOR
freq : standard (gravitational wave) frequency [Hz]
Length : Michelson ifo arm length [m]
phi_0 : static differential arm length tuning phase [rad]
P_in : input power [W]
File:      ~/Documents/git/SU/dissertation/code/ifo_configs.py
Type:      function
```

[4]:

```
H = MICH(nu, L, PHI_0, P_IN, OMEG)
```

[5]:

```
fig, ax1 = plt.subplots()
ax1.set_xlabel('frequency [Hz]')
ax1.set_ylabel('H$_{\mathdefault{MICH}}$ [$\mathdefault{W/m}$]',color='C0')
#ax1.plot(w/(FSR), F_w_cc_modsq*100)
ax1.loglog(bode_amp(H),linewidth=7.5, color='C0')
#plt.ylim([10e-6, 10e0])
ax2 = ax1.twinx()
#ax2.plot(w/(FSR), (180/np.pi)*np.arctan(F_w_cc.imag/F_w_cc.real), '--')
ax2.semilogx(nu,(180/np.pi)*np.arctan(np.imag(H)/np.real(H)), '--', linewidth=7.
 ↪5,color='C1')
#plt.xlabel('frequency [FSR]')
plt.xlim([1,1e5])
plt.ylabel('phase [deg]',color='C1')
fig.savefig('../figs/INTRO/mich_fr.pdf', dpi=300, bbox_inches='tight')
```



1.1.1 *The derivation noted was heavily influenced by Koji's explanation in DCC G1401144

Though with the provided frequency depdenence and optical gain, we still need to understand a starting noise floor spectra and compare to our anticipated * The noise floor is established

1.2 Shot noise

- A fundamental limit imposed by the statistical nature of photon counting
- The photon counting follows Poisson statistics
  - Photon counting variance (variance is equal to the mean)

$$< (n-\bar{n})^2 >= \frac{P\Delta t}{\hbar\Omega}$$

  - Power variance:

$$< (P-\bar{P})^2 >= \hbar\Omega\bar{P}\Delta t$$

  - PSD of the measured power between two uncoorelated moments in time:

$$S_{\mathrm{P}}(\omega) = \lim_{T\to\infty}\frac{2}{T}\Big\langle \Big| \int_{-T}^{T}(P(t)-\bar{P})e^{-i\omega t}dt\Big|^2 \Big\rangle$$

$$= \lim_{T\to\infty} \frac{2}{T}\int_{-T}^{T} \hbar\Omega\bar{P}dt$$

$$= 2\hbar\Omega\bar{P}$$

- Where the ASD is:

$$[S_P(\omega)]^{1/2} = [2\hbar\Omega\bar{P}]^{1/2}$$

The signal to noise is established by dividing the frequency dependent optical gain times the gravitational wave ASD $\left([\mathrm{S_h}(\Omega)]^{1/2}\right)$ by the noise ASD:

$$\mathrm{SNR} = \mathrm{G_{opt}}(\omega)[\mathrm{S_h}(\omega)]^{1/2}/S_\mathrm{N}(\omega) = \mathrm{H_{MICH}}/[S_P]^{1/2} = \left(\frac{\Delta\phi(\omega)}{h_0}\frac{P_\mathrm{in}}{2}\sin(\phi_0)\right)\Big/[2\hbar\Omega\bar{P}]^{1/2}$$

This is to say that for the stated gravitational wave ASD, and for an SNR of 1, we establish the following threshold for detector:

$$\left[\mathrm{S_h}(\omega)\right]^{1/2}\{\mathrm{SNR}\geq 1\} \geq \frac{[S_\mathrm{N}(\omega)]^{1/2}}{\mathrm{H_{MICH}}(\omega)}$$

Where

$$\frac{[S_\mathrm{N}(\omega)]^{1/2}}{\mathrm{H_{MICH}}(\omega)} = \frac{[2\hbar\omega\bar{P}]^{1/2}}{\Delta\phi(\omega)[P_\mathrm{in}/2]\sin(\phi_0)} = \left(\frac{\hbar\Omega}{\omega P_\mathrm{in}}\right)^{1/2}\frac{[r_x^2+r_y^2-2r_xr_y\cos(\phi_0)]^{1/2}}{\sin(L\omega/c)}e^{iL\omega/c}$$

[6]:

```
N_shot?
```

```
Signature: N_shot(OMEGA, P_in)
Docstring:
Interferometer shot noise calculator
OMEG: OPTICAL angular frequency [rad Hz]
Length : ifo arm length [m]
phi_0 : static differential arm length tuning phase [rad]
P_in : Input power [W]
File:      ~/Documents/git/SU/dissertation/code/ifo_configs.py
Type:      function
```

[7]:

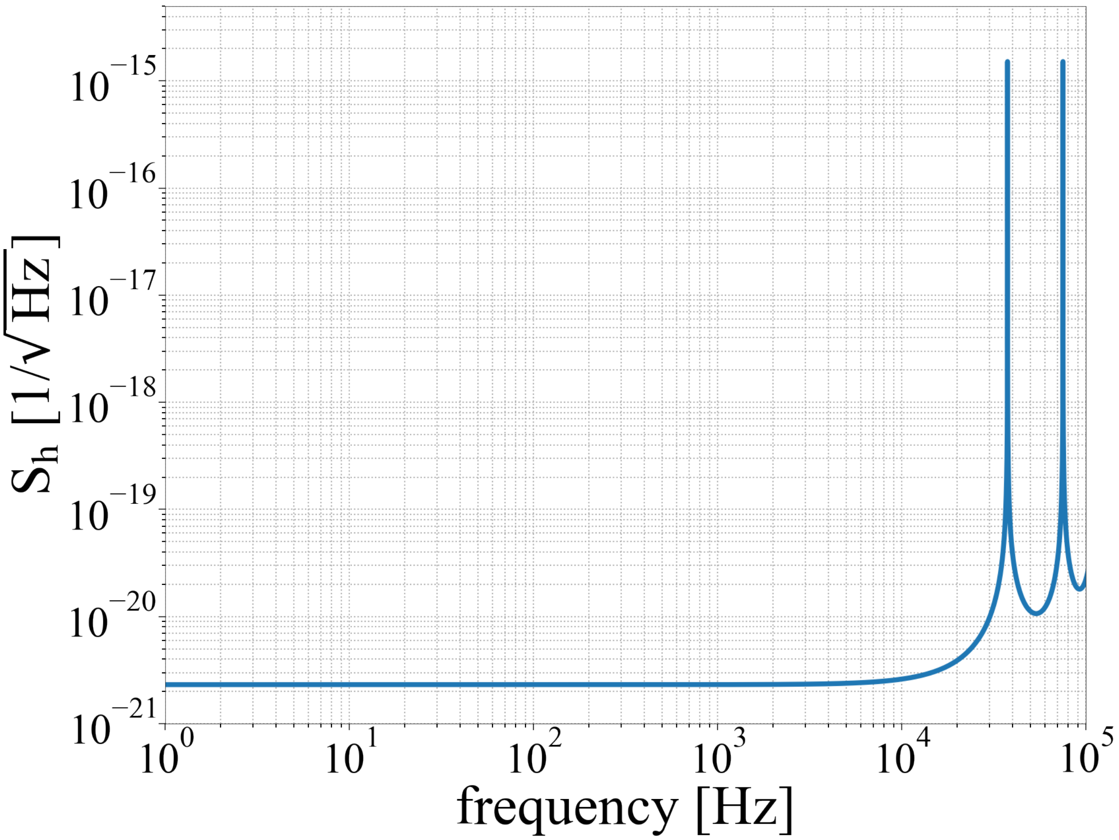
```
S_h = N_shot(OMEG, P_IN)
print(S_h)
```

```
6.831801787605637e-09
```

[12]:

```
#ax1.plot(w/(FSR), F_w_cc_modsq*100)
plt.loglog(nu, S_h/bode_amp(H), linewidth=7.5, color='C0')
plt.ylim([1e-21, .5e-14])
plt.xlabel('frequency [Hz]')
plt.ylabel('$\mathdefault{S}_\mathdefault{h} \;  \mathdefault{[ 1 /␣
 ↪\sqrt{\mathdefault{Hz}}]} $')
#ax2_ = ax1_.twinx()
```

```
#ax2.plot(w/(FSR), (180/np.pi)*np.arctan(F_w_cc.imag/F_w_cc.real), '--')
#ax2_.semilogx(nu,(180/np.pi)*np.arctan(np.imag(S_h)/np.real(S_h)), '--',␣
 ↪linewidth=7.5,color='C1')
#plt.xlabel('frequency [FSR]')
plt.xlim([1,1e5])
plt.grid(visible=True)
#plt.subplots_adjust(hspace = 1)
#plt.ylabel('phase [deg]',color='C1')
#plt.tight_layout(rect=[0,0,1,1])
#plt.title('')
#plt.subplots_adjust(bottom=.1, top=.85) #, right=.8, left=.1)
plt.savefig('../figs/INTRO/mich_sensi.pdf', dpi=300, bbox_inches='tight')
```


10^{-15}
10^{-16}
10^{-17}
10^{-18}
10^{-19}
10^{-20}
10^{-21}
S_h $[1/\sqrt{\mathrm{Hz}}]$
10^0
10^1
10^2
10^3
10^4
10^5
frequency [Hz]