# Virgo SR Error Signal Study Using Finesse 3

Jonathan W. Perry

# The Issue

- We need a global error signal to control SRM alignment.

- LIGO did a similar study finding that sideband-sideband demodulation best isolates SR alignment information.

- Question: can Virgo do something similar?

# The Study (in short)

- For different modulation/demodulation schemes, check the following:
  1. **Signal strength -> optical response to misalignment**
  2. BS/SR coupling -> compass plots
  3. Response to mode-mismatch -> apply lensing to compensation plate

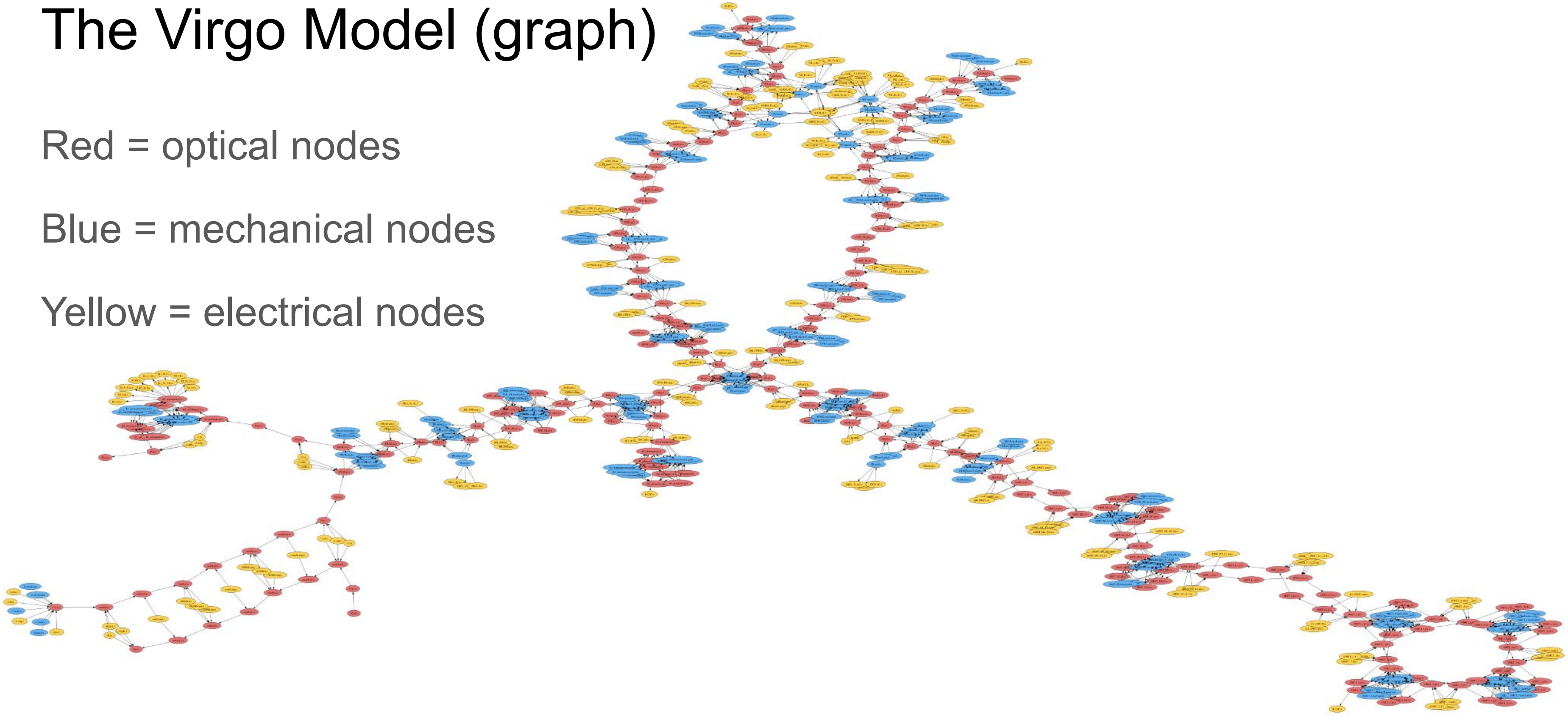
- To do this in Finesse we need a reliable Virgo model

# The Virgo Model (graph)

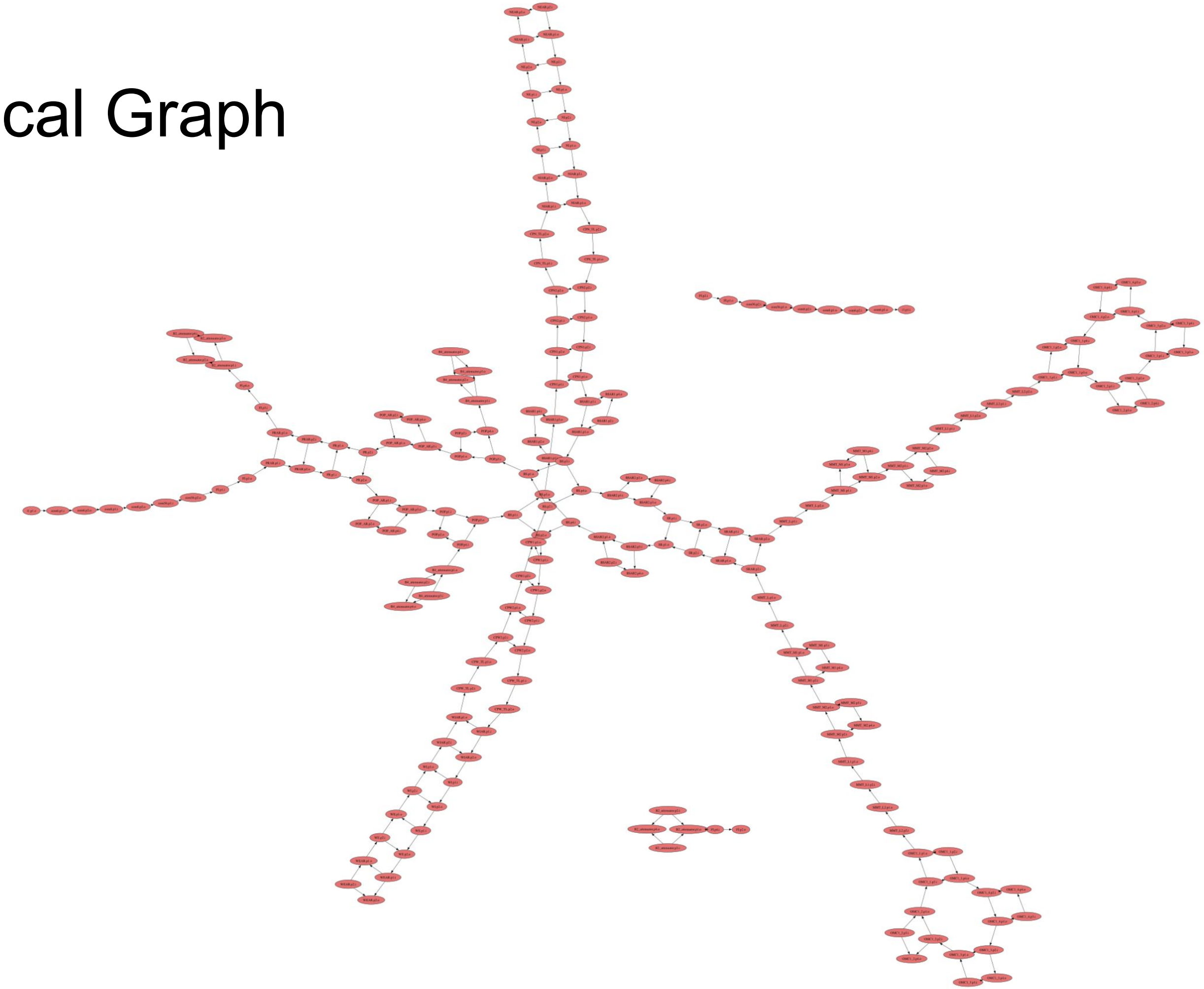
Red = optical nodes

Blue = mechanical nodes

Yellow = electrical nodes



# The Optical Graph



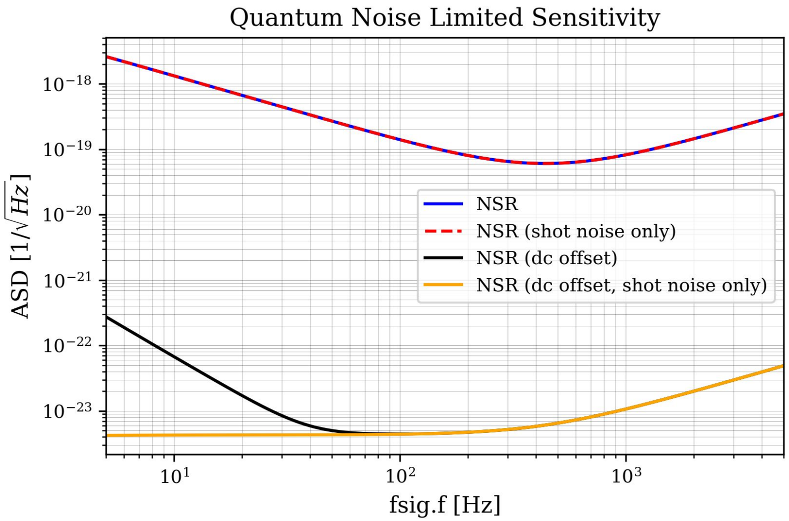
# The `finesse-virgo` Package

Python package which adds Virgo specific functionality to Finesse 3.

Repository: https://gitlab.com/ifosim/finesse/finesse-virgo

Documentation (WIP): https://finesse.ifosim.org/docs/finesse-virgo/latest/

Installation: `pip install finesse-virgo`



# The `finesse-virgo` Basics

1. Create a new Virgo object

```
import finesse.virgo
virgo = finesse.virgo.Virgo()
```

2. Tune the model to an operating point
   a. Pre-tune

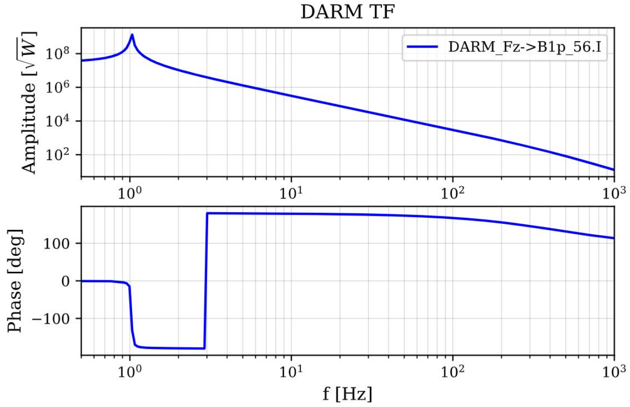
```
virgo.make(dc_lock=False)
```

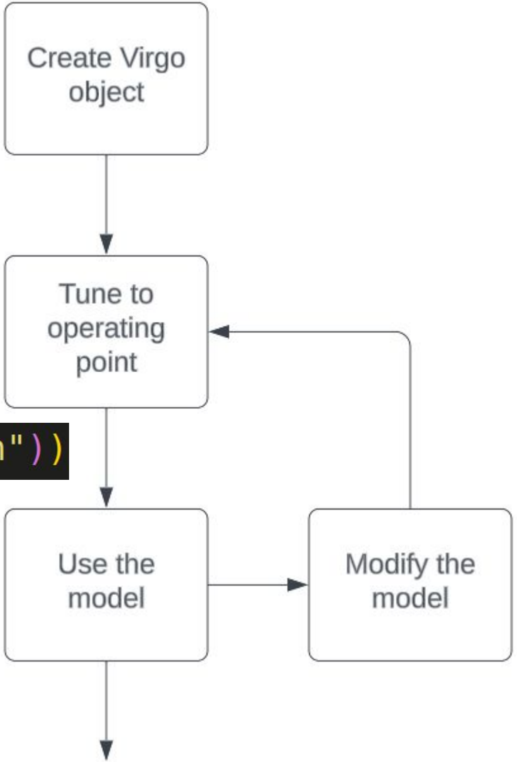
   b. Run the locks

```
virgo.model.run(RunLocks(method="newton"))
```

3. Use the model

```
virgo.plot_DARM()
```





Create Virgo object

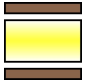
Tune to operating point

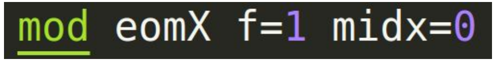
Use the model

Modify the model

# The Modifications (some)

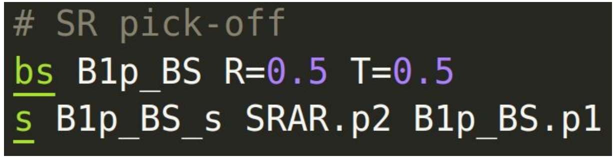
Before parsing:





```
mod eomX f=1 midx=0
```

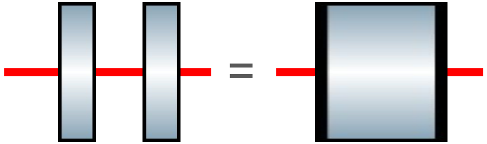
1. Additional modulator
2. Beamsplitter after SRM

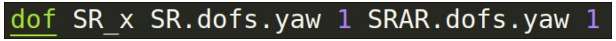


```
# SR pick-off
bs B1p_BS R=0.5 T=0.5
s B1p_BS_s SRAR.p2 B1p_BS.p1
```

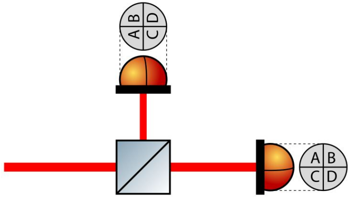
After parsing:

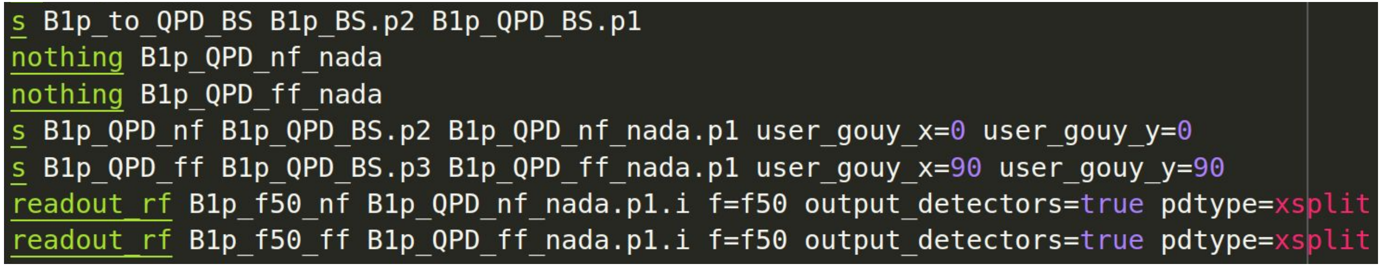
1. Define degrees of freedom (for composite mirrors)
2. Create QPD RF readouts (demodulators)





```
dof SR_x SR.dofs.yaw 1 SRAR.dofs.yaw 1
```

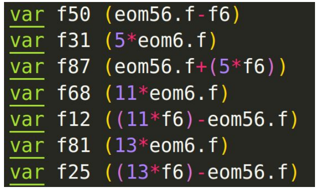




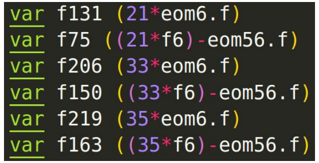
```
s B1p_to_QPD_BS B1p_BS.p2 B1p_QPD_BS.p1
nothing B1p_QPD_nf_nada
nothing B1p_QPD_ff_nada
s B1p_QPD_nf B1p_QPD_BS.p2 B1p_QPD_nf_nada.p1 user_gouy_x=0 user_gouy_y=0
s B1p_QPD_ff B1p_QPD_BS.p3 B1p_QPD_ff_nada.p1 user_gouy_x=90 user_gouy_y=90
readout_rf B1p_f50_nf B1p_QPD_nf_nada.p1.i f=f50 output_detectors=true pdtype=xsplit
readout_rf B1p_f50_ff B1p_QPD_ff_nada.p1.i f=f50 output_detectors=true pdtype=xsplit
```

# The Experiment (optical response)

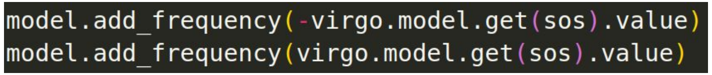
1. Set modulation frequency
2. Add sideband-of-sideband (SoS) frequency
3. Set demodulation frequency
4. Optimize demodulation phase
5. Misalign mirror



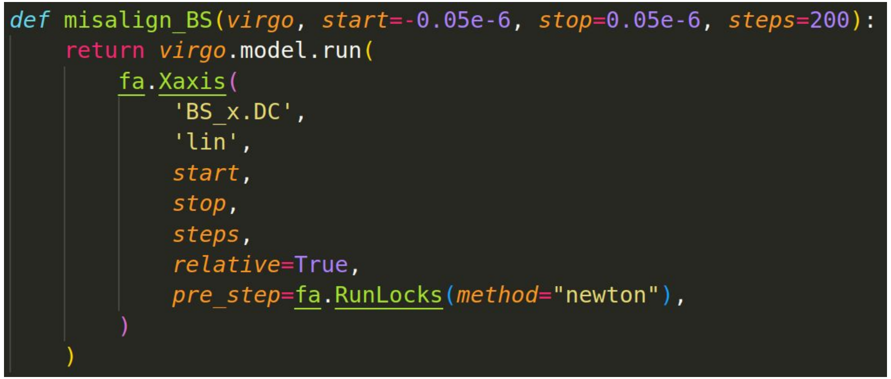
```
var f50 (eom56.f-f6)
var f31 (5*eom6.f)
var f87 (eom56.f+(5*f6))
var f68 (11*eom6.f)
var f12 ((11*f6)-eom56.f)
var f81 (13*eom6.f)
var f25 ((13*f6)-eom56.f)
```



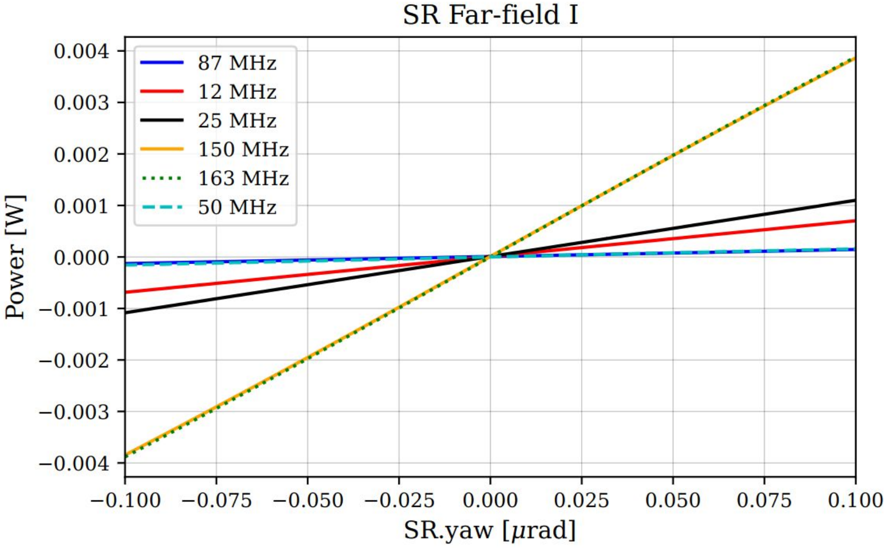
```
var f131 (21*eom6.f)
var f75 ((21*f6)-eom56.f)
var f206 (33*eom6.f)
var f150 ((33*f6)-eom56.f)
var f219 (35*eom6.f)
var f163 ((35*f6)-eom56.f)
```



```
model.add_frequency(-virgo.model.get(sos).value)
model.add_frequency(virgo.model.get(sos).value)
```



```
def misalign_BS(virgo, start=-0.05e-6, stop=0.05e-6, steps=200):
    return virgo.model.run(
        fa.Xaxis(
            'BS_x.DC',
            'lin',
            start,
            stop,
            steps,
            relative=True,
            pre_step=fa.RunLocks(method="newton"),
        )
    )
```



# Conclusion

- This study led to the implementation of a new modulation frequency (81MHz).
  - Turned off due to technical reasons (modulation latency, marginally stable cavities)
- The `finesse-virgo` package allows for convenient modelling of Virgo configurations.
- See additional notebooks in the repository for more examples.