Shortening
the seismic attenuation chain
A compact mechanical filter

A. Allocca, L. Bellizzi, S. Bianchi, V. Boschi, E. Calloni, M. Carpinelli, P. Chessa , D. D'Urso, R. De Rosa, L. Di Fiore, F. Fabrizi, I. Ferrante, F. Fidecaro, A. Fiori, A. Gennai, A. Longo, L. Massaro, M. Montani, L. Papalini, M. Palaia, M. Razzano, D. Rozza, P. Ruggi, L. Trozzo, M. Vacatello, A. Viceré

09/05/2023

General strategy for LF noise

Seismic noise underground 200 times less than at Virgo - LIGO

Position/acceleration sensors readout hits the noise floor of instrument

Local control is effective only upstream the attenuation chain

Otherwise one needs the full interferometer, which injects technical noise -> Active Noise Mitigation Division

- **Rely on passive attenuation**
- **Gain by reducing the normal mode frequencies**
- **Improve upstream isolation with better sensing and actuation**

ET-LF requirements

LF noise is given by

- Microseism motion
- Newtonian noise
- Upconversion of residual motion into the detection band

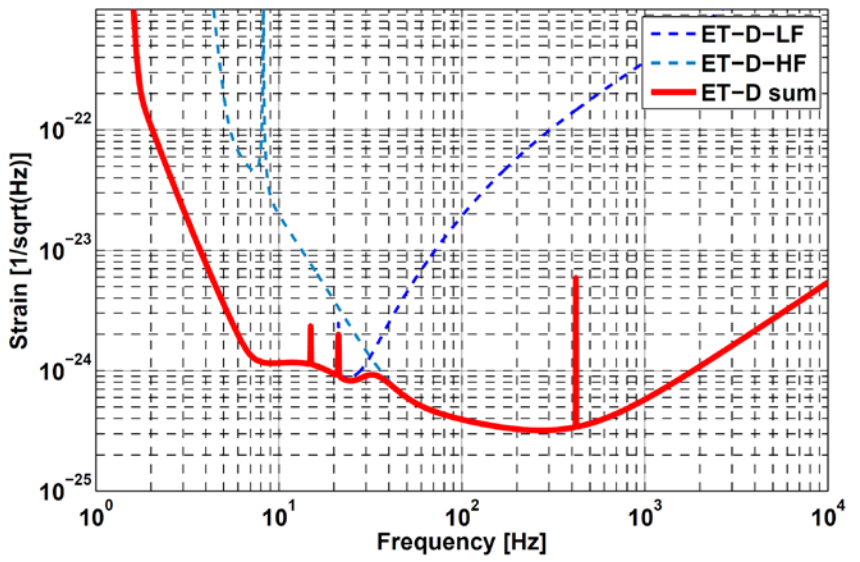
Design curve based on 17 m tall suspensions

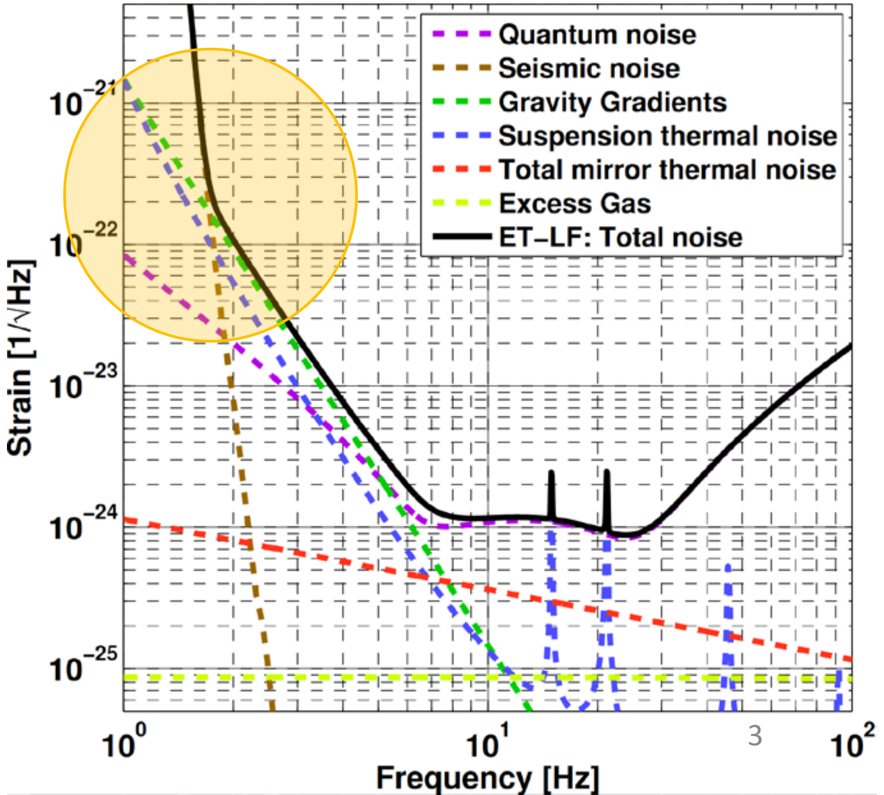
Reduction to less than 10 m:

- Significantly lower cavern excavation cost
- Suspension management similar to Virgo

Newtonian noise crossing:

$2\ 10^{-22}\ Hz^{-1/2}$ at 1.8 Hz (AdV: 3.2 Hz)





Challenge: Fit suspension in 10 m

1. **Act on ground / suspension interface actively**
2. **Act on suspension point actively/passively**
3. **Superattenuator chain design**
4. **Payload design compatibility: large vertical occupancy announced**

Seismic levels

Sardinia vertical

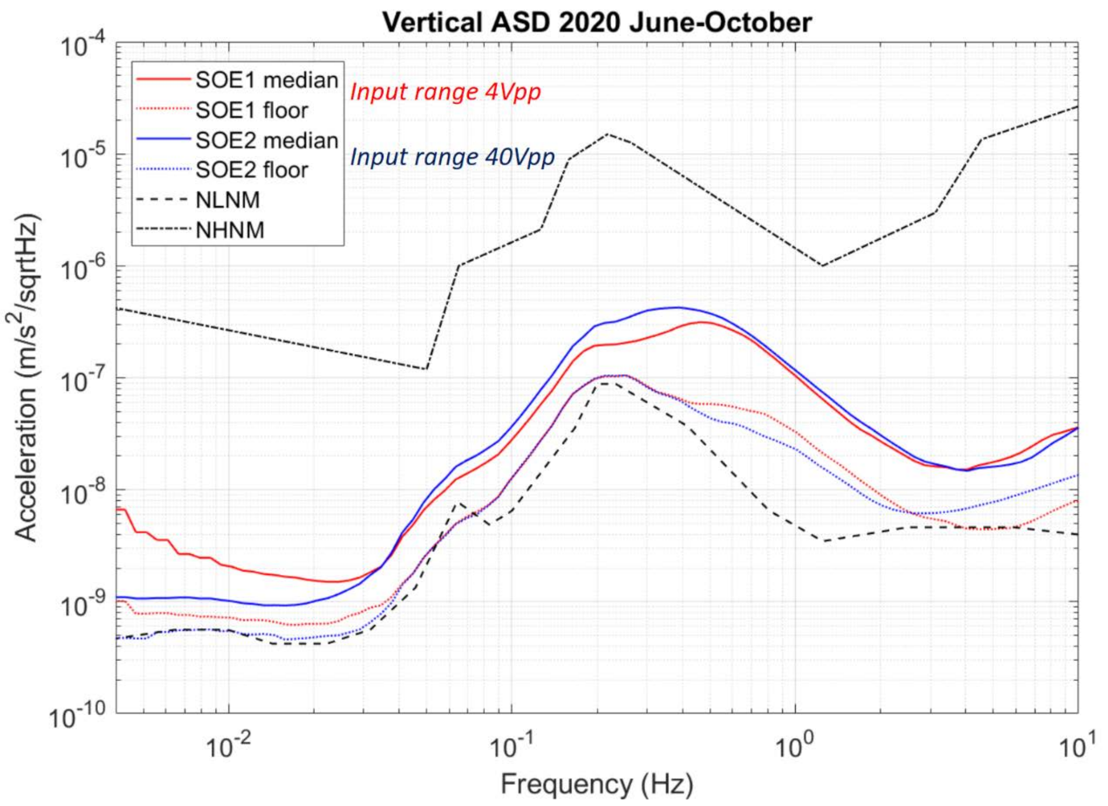
$3\ 10^{-8}$ ms^{-2} $Hz^{-1/2}$ at 2 Hz
$7.5\ 10^{-10}$ m $Hz^{-1/2}$ at 2 Hz

Suspension point

10^{-13} m $Hz^{-1/2}$ at 2 Hz

Mirror

$2\ 10^{-22}$ $Hz^{-1/2}$ at 2 Hz
10^{-18} m $Hz^{-1/2}$ at 2 Hz



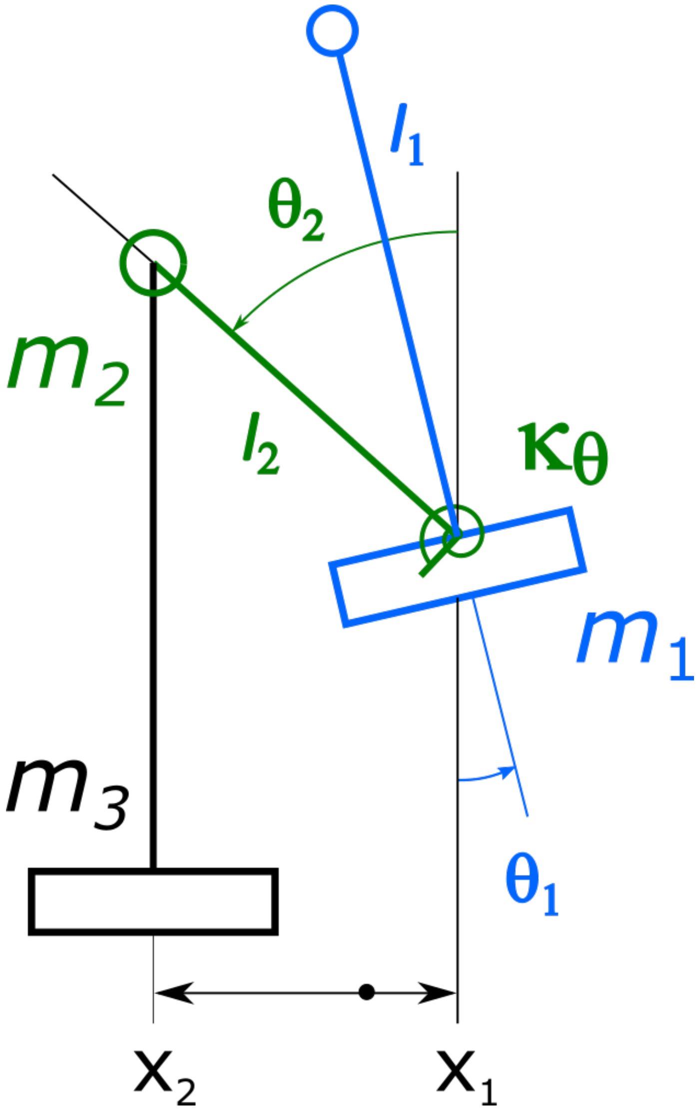
Attenuation: $A=10^5$ including marionetta and mirror
Assume 7 filters including mirror. Mean single stage attenuation $A_i = A^{5/7} = 5.2$
$A_i = \frac{f_0^2}{f^2 - f_0^2}$ leads to $f_0 = 0.8$ Hz, or a pendulum length of 40 cm for f = 2 Hz

Pendulum – Inverted pendulum

How to soften a suspension stage

- Spare length
- For κ_θ sufficiently stiff, the system is stable

```
• l1:  1.544, # Pendulum length\
• l2:  0.520, # IP length\
• T1: 2551.0, # Pendulum tension\
• T2: 1766.0, # IP compression\
• m1:   80.0, # Pendulum mass\
• m2:   80.0, # Filter mass\
• m3:  100.0, # Load\
• I1s:  20.0, # Pendulum moment of inertia \
• I2s:   0.8, # IP moment of inertia\
• k:  1700.0, # flex joint elastic constant\
```



Normal mode frequencies
0.68 Hz 0.74 Hz

Attenuation of Pendulum-Inverted Pendulum

$$A = \left(\frac{f_0^2}{f^2 - f_0^2}\right)^2$$

Horizontal attenuation of a single PIP

For f_0 = 0.75 Hz

A_2 : attenuation at 2 Hz 2.7 10^{-2}

Two PIP

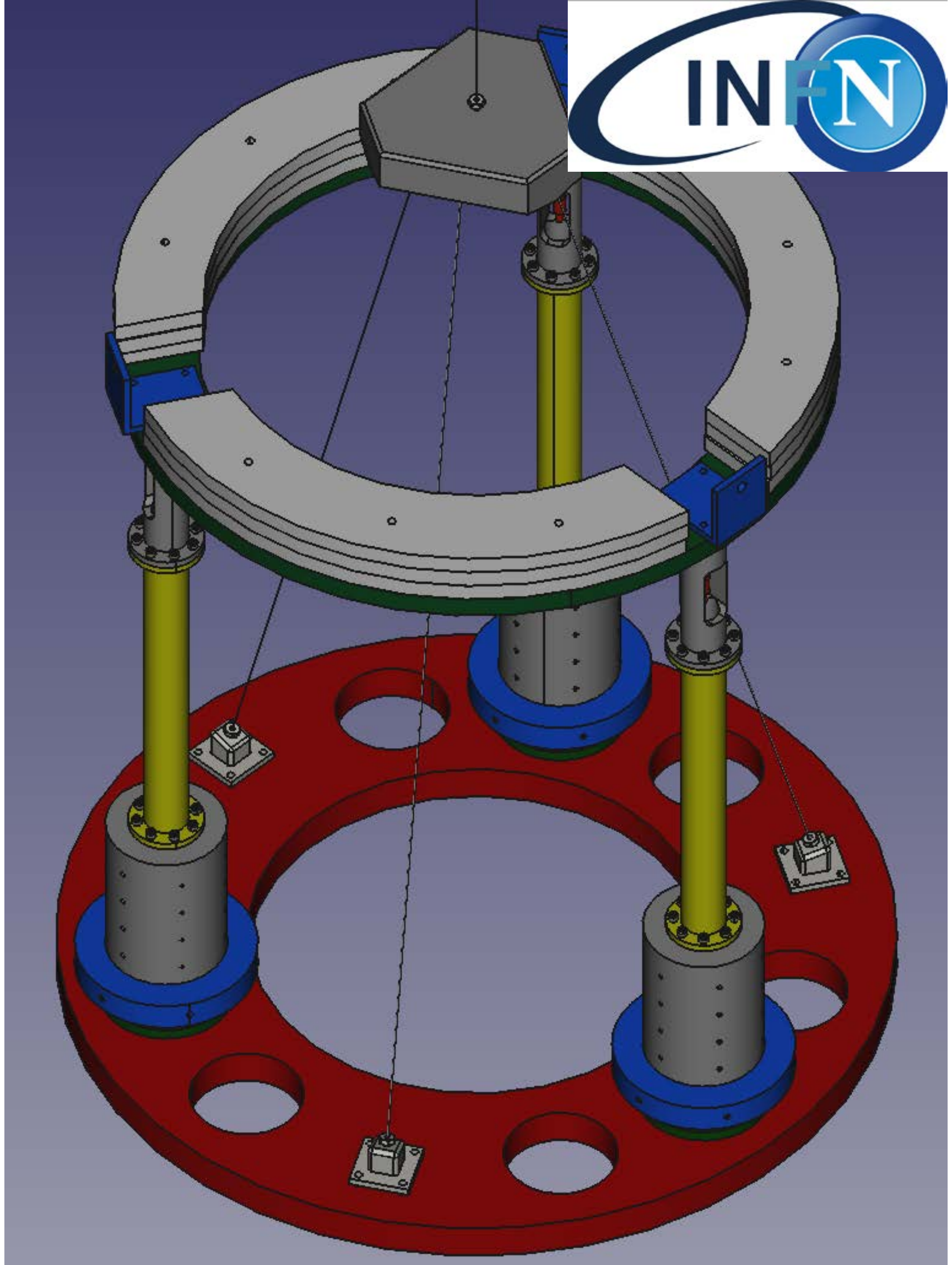
A_2 : attenuation at 2 Hz 7.2 10^{-4}

Three PIP

A_2 : attenuation at 2 Hz 1.9 10^{-5}

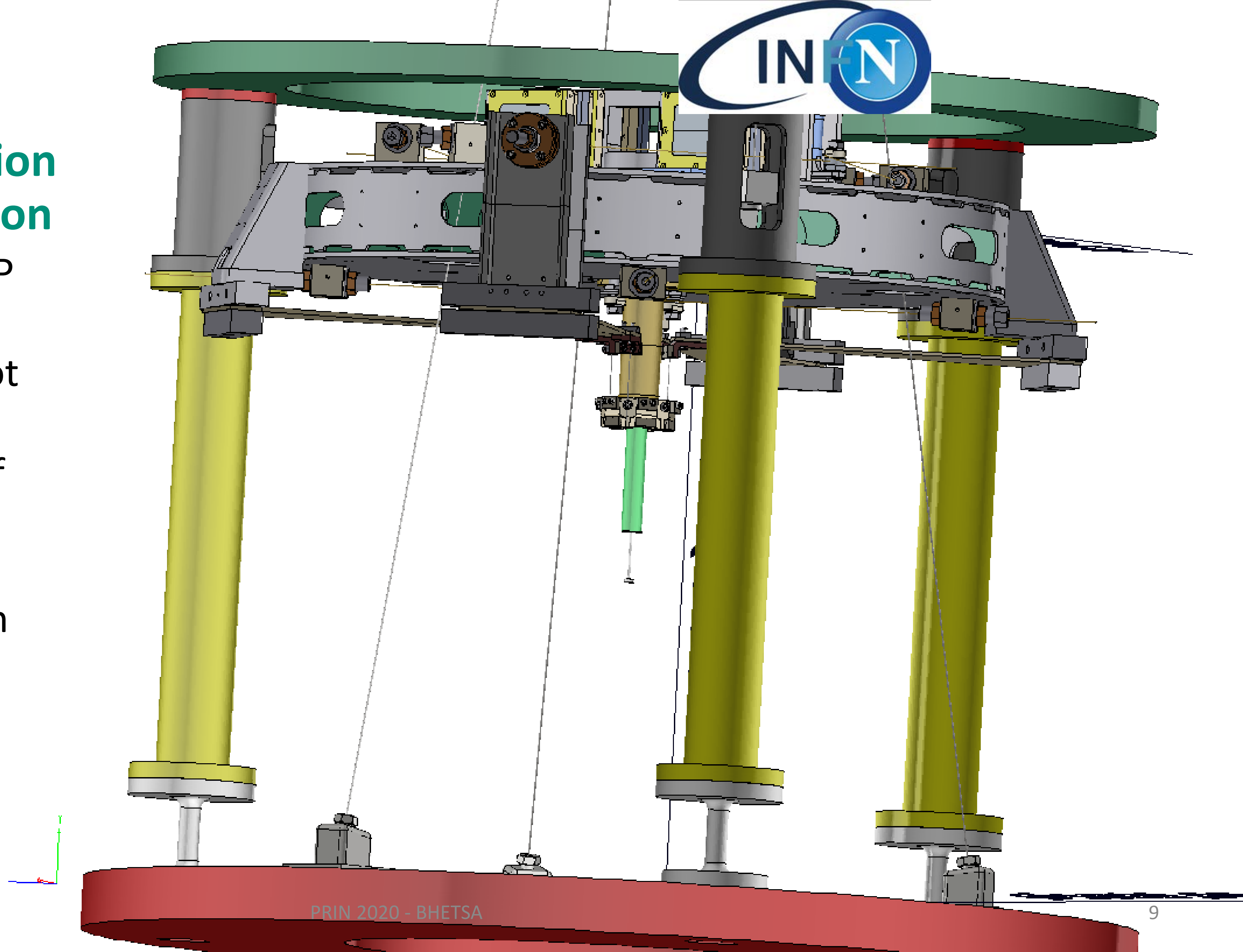
Basic structure

Explicit coupling of pendulum and roll DOF



Standard filter addition for vertical attenuation

- Filter suspended to IP legs
- IP counterweights not represented
- Includes one stage of vertical attenuation
- Hook to next stage above first pendulum mass



Overall length

- **A PIP chain can be built**
  - Hook of the second PIP above the first filter
  - Current PIP length 1.55 m
  - Two PIP can live in 2.60 + 0.40 = 3.0 m accounting for a dedicated vertical attenuation stage
  - Three PIP can live in 4 m
  - Proximity of different stages allows feedback control of normal modes, at least where sensor noise is not dominant (em or optical sensors like HoQI)

Simulation needs: OctoPYus package

- **Package overview**
  - MATLAB version developed by Paolo Ruggi
  - To be hosted on Gitlab
  - Now private repo, at some point public
  - Available for developers (via git)
  - Available for end users (via PIP, only on gitlab Package Registry, then move to Python Package Index

- **Structure**
  - Creation of configuration via GUI
  - Configuration saved to JSON format
  - Network solving to include feedback loops
  - Calculation of transmission matrices
  - Calculation of Transfer Function

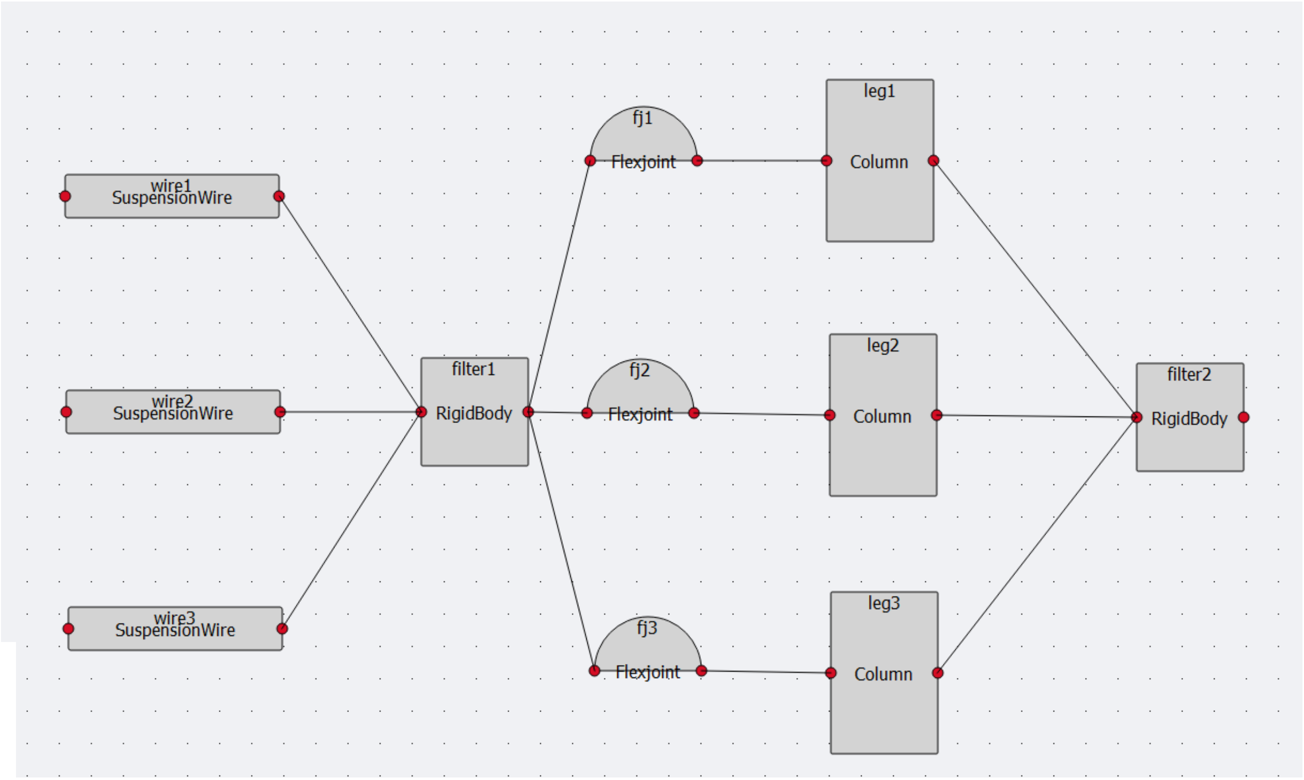
OctoPYus development

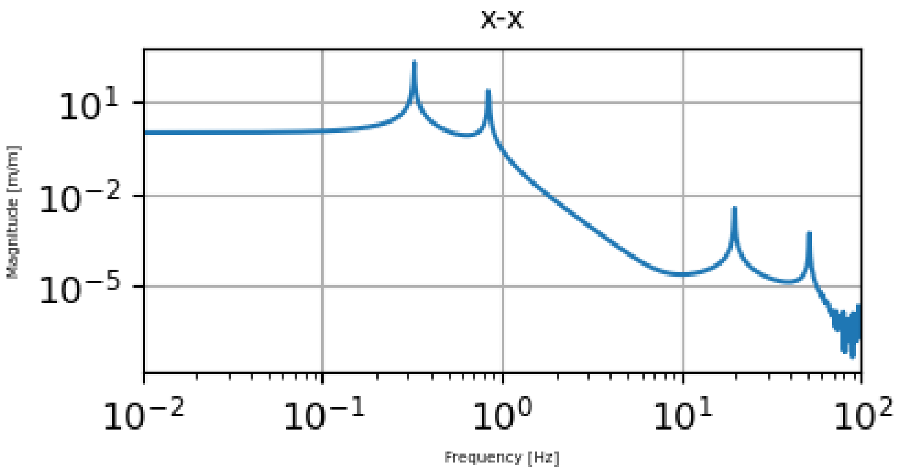
- **Main Features**
  - Python (should be easy and multi-platform)
  - Make it available for public usage (later on in the BHETSA life)

- **Development point of view**
  - Implemented automatic documentation via Sphinx
  - Implemented Pipelines for Continuous Integrationx
  - Unittest for testing pieces of code
  - Move to public platforms (Readthedocs and Python Package Index)
  - Test against different versions of Python3
  - Implement for usage with multi platform (Windows, MacOS2)
  - Integrating new features (e.g. NetSolver class)

Test case: schematic PIP

- **PIP**
  - Resonances @0.2 Hz (pend) and 0.8 (IP)
  - attenuation $1/f^4$





L. Massaro Master thesis

Coding progress

- **General structure**
  - Pipelines done
  - Switch to public repo/documentation/package index when ready

- **Package development**
  - Test with multi platform
  - Finalize GUI
  - Implement basic configurations as test cases
  - Integrate network solver (already in Gitlab, merge done, test integration)

Perspective

- **Prototype construction**
  - Principle verification vs simulations
  - Engineering verifications
  - Assembly
- **Design optimization**
- **Proposal for a short seismic attenuation chain for ET**

Collaboration welcome!