OctoPyUs: A Python tool for transfer function simulation with the impedance matrix method

Matteo Montani[1,2], Alessandro Longo[1,2]

[1] Università degli Studi di Urbino "Carlo Bo", I-61029 Urbino, Italy
[2] INFN, Sezione di Firenze, I-50019 Sesto Fiorentino, Firenze, Italy
e-mail: matteo.montani@uniurb.it alessandro.longo@uniurb.it

1. BACKGROUND

The matrix impedance method used for transfer function computation can be used to model, in the frequency domain, the dynamic of complex objects such as the Virgo Superattenuator in term of connected elementary components or elements.

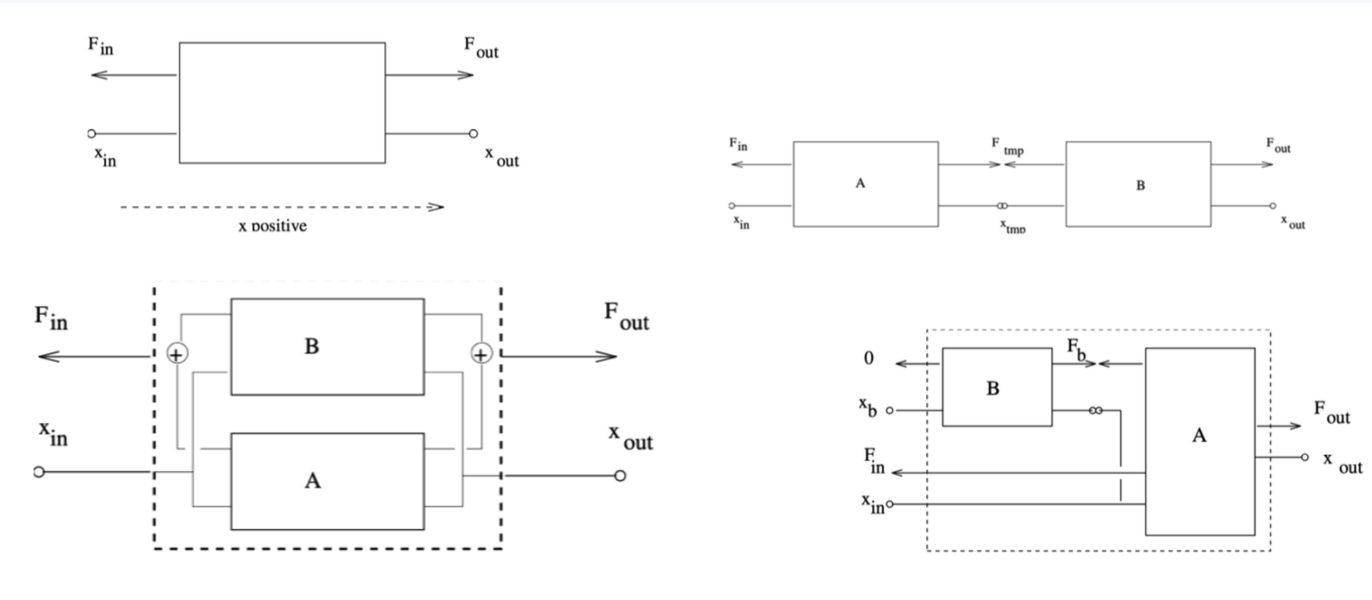


Fig .1 Schematic representation of an element (top left) and of a series of elements (top right) following the impedance matrix method approach. An example of parallel (bottom left) and derivation (bottom right) connection is also shown.

An element which is part of a mechanical chain can be represented as in Fig. 1. The arrows represent forces exerted by the element on the boundary (positive direction is outgoing), while the lines ended by circles represent the coordinates of the boundary of the element.

$$\begin{pmatrix} x_{out} \\ F_{out} \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x_{in} \\ F_{in} \end{pmatrix} \qquad \begin{pmatrix} x_{out} \\ F_{out} \end{pmatrix} = \mathbf{B} \cdot \mathbf{A} \begin{pmatrix} x_{in} \\ F_{in} \end{pmatrix}$$

Each element is represented as an impedance matrix connecting input and output (left). Joining the elements by means of different connections i.e. in series or in parallel, transfer functions of increasingly complicated physical systems can be obtained. For series connection of two elements A and B, the equivalent matrix is the matrix product (right)

2. TRANSFER FUNCTION COMPUTATION

- Multiply impedance matrices of elements of the system
- Take into account their mutual connections and their displacement outside the center of mass
- Obtain equivalent impedance Z describing the system
- From Z, the transfer function TF can be obtained:

$$Z = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \qquad TF = A - (B/D)C$$

In simplyfied cases the following equalities hold

$$A \cdot D^T - B \cdot C^T = I, \qquad A \cdot B^T = B \cdot A^T \qquad C \cdot D^T = D \cdot C^T$$

And transfer functions can be obtained from simplified relationships

$$TF = I/D^T$$

- Case XoXi: How a displacement of the input node (e.g. top of Superattenuator) induces a displacement of the output node (e.g. test mass)

$$TF = -B/D$$

- Case XoFo: How a force applied at the output node induces a displacement at the same output node

3. MASS and SPRING

Given the mass m and inertia I of the body, the impedance matrix representative of a massive element is given by

$$\mathbf{M} = \mathbf{I} \cdot \begin{pmatrix} m \\ m \\ m \\ I_x \\ I_y \\ I_z \end{pmatrix}$$

The motion equation of a point mass is

$$M\ddot{x} = F_{in} - F_{out}$$

which in the frequency domain gives

$$\begin{pmatrix} x_{out} \\ F_{out} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ M\omega^2 & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} x_{in} \\ F_{in} \end{pmatrix}$$

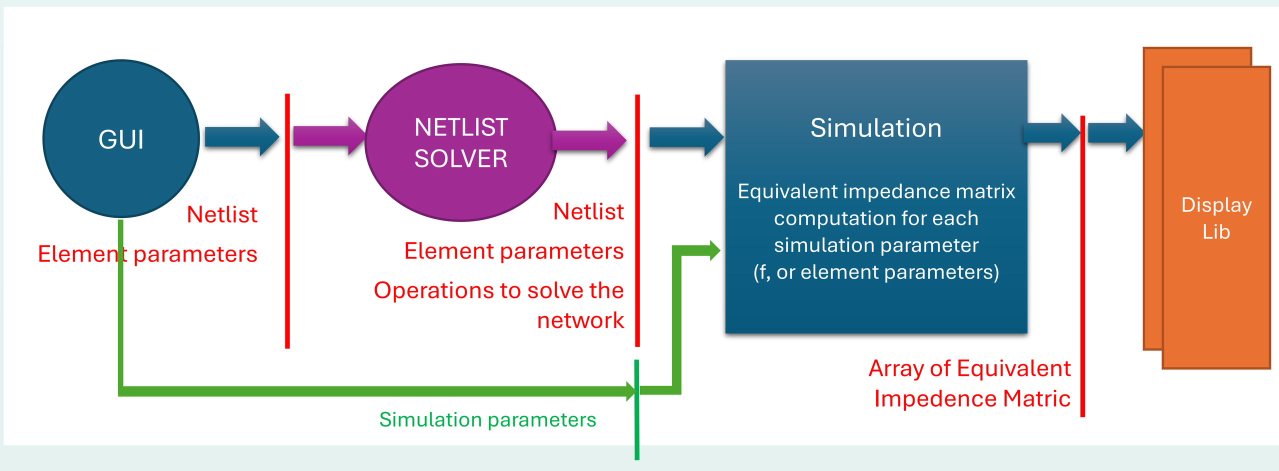
- Impedance matrix of spring element described in term of stiffness matrix k
- k: dimension up to 6 and up to 36 different elements
- Given k, the spring impedance matrix K is

$$\mathbf{K} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad \text{where} \quad \begin{cases} A = \mathbf{I} \\ B = -\frac{1}{k}\mathbf{I} \\ C = \mathbf{0} \\ D = \mathbf{I} \end{cases}$$

- I and 0 are the 6x6 null and identity matrix

4. SOFTWARE ARCHITECTURE

OctoPyUs flowchart



Netlist (leg.json)

Netlist (IP.json)

GUI: This component has not been developed yet; however, the data format it will generate has already been defined: a JSON file containing the system's netlist.

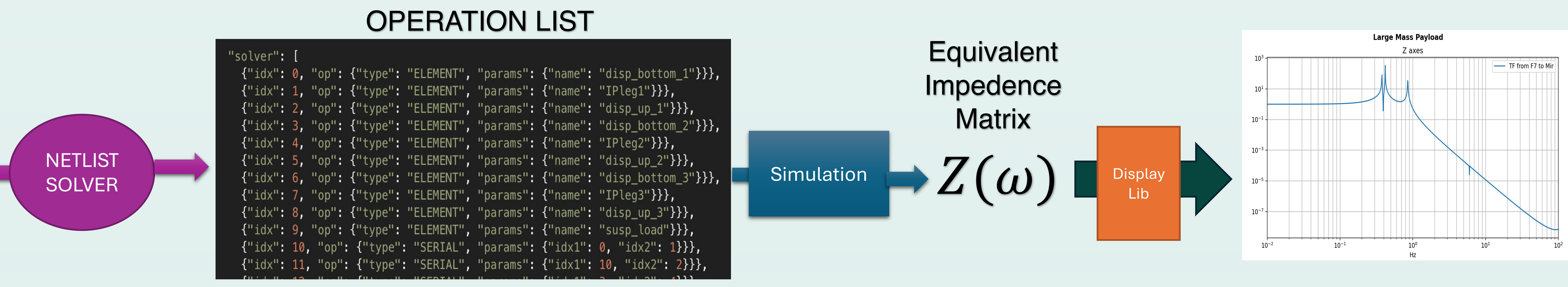
Netlist JSON File: Describes the system to simulate using a list of elements (whit their parameters) and their connections (nodeIn, nodeOut). The elType now implemented are: mass, wire/beams, blades, cabling, and custom library.

Solver: To generate the equivalent impedance matrix, it is necessary to create a list of operations that solve the specific network.

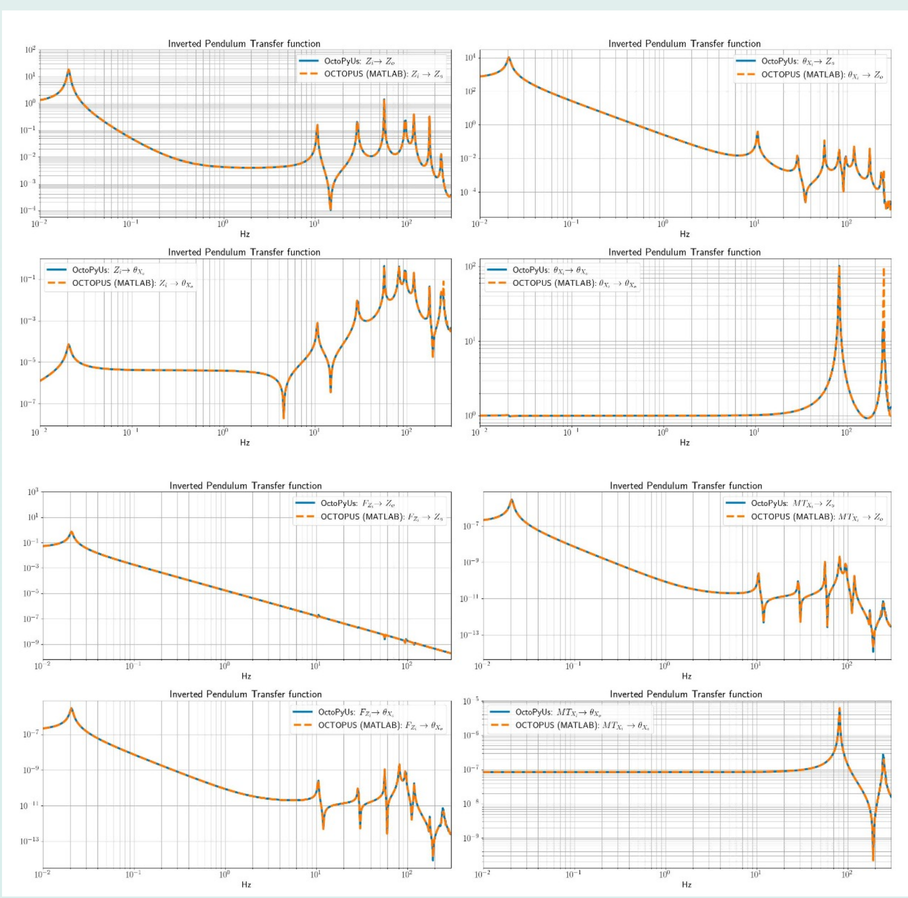
This module analyzes the netlist and produces a corresponding list of operations, which is then added to the JSON file in a new section.

Simulation: Using the information provided by the Solver, the system's equivalent impedance matrix Z is computed.

DisplayLibA software library has been developed to save and load simulation results and to plot, using Matplotlib functions, the various transfer functions of the degrees of freedom relevant to the designer, as well as the contribution of thermal noise.



5. COMPARISON WITH OCTOPUS' INVERTED PENDULUM AND SIMULATION OF THERMAL NOISE



Transfer Function Validation

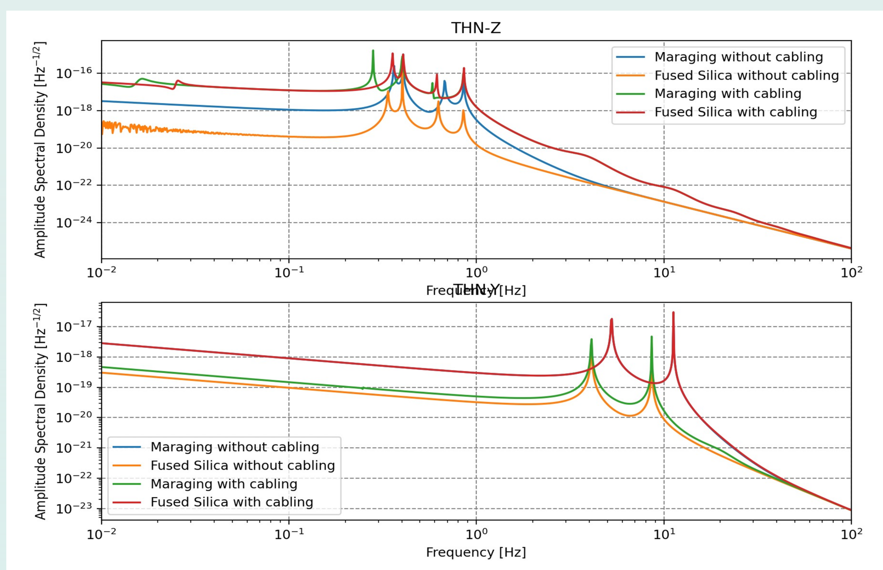
- The TF of the inverted pendulum (see IP.json and Figure) was computed using OctoPyUs for various d.o.f.
- It was compared with OCTOPUS' results
- OCTOPUS: MATLAB software. Validated by experimental measures and routinely used by commissioners and simulation experts
- Similar results are obtained for the simulation of the standard filter's TF (see VIR-0584A-24)

Thermal noise

Thermal noise represents a significant phenomenon that governs certain design aspects of the suspension chain in gravitational wave interferometers; the ability to perform simulations focused on this effect is therefore of great importance. The fluctuation-dissipation theorem states that the power spectral density $x^2(\omega)$ of the thermal fluctuations associated with the observable $X_P(\omega)$ is related to the transfer function $H(\omega) = \frac{X_P(\omega)}{F_P(\omega)}$ between the applied force and the displacement itself through the following relation:

$$x^2(\omega) = \frac{4K_B T}{\omega}\,\mathrm{Im}\{H(\omega)\}$$

Where K_B is the Boltzman costant, T the absolute temperature [2].



OctoPyUs Simulation results comparing maraging steel and fused silica fiber suspensions for the marionette. Calculations are based on a Virgo large-mass configuration, though the solution is adaptable to ET-HF at room temperature. The impact of 16 copper control wires (cabling) with a loss angle \phi = 0.5 is also included.

6. FUTURE WORK

- Development of active controls
- GUI for system modelling. GUI for interactive plotting
- Unmodelled topologies (optical spring)
- OctoPyUs software publication

7. REFERENCES

1. OctoPyUs Technical note (VIR-0584A-24)
2. OCTOPUS: a mechanical simulation tool based on impedance matrices (VIR-0784A-24)
3. OCTOPUS (zenodo)

OctoPyUs is supported by the PRIN 20202C5XA9 project "Black Holes for ET in SArdinia" (BHETSA), funded by the Italian Ministry of University and Research (MUR). More information about the project is available at http://bhetsa.df.unipi.it/.