ROSS - Rotordynamic Open Source Software

Raphael Timbó[1], **Rodrigo Martins**[2], **Gabriel Bachmann**[4], **Flavio Rangel**[5], **Júlia Mota**[3], **Juliana Valério**[5], **and Thiago G Ritto**[2]

1 Petrobras - Petróleo Brasileiro S.A. **2** Universidade Federal do Rio de Janeiro, Department of Mechanical Engineering, Rio de Janeiro, Brazil **3** Universidade Federal do Rio de Janeiro, Graduate Program in Informatics, Rio de Janeiro, Brazil **4** Universidade Federal do Rio de Janeiro, Department of Electrical Engineering, Rio de Janeiro, Brazil **5** Universidade Federal do Rio de Janeiro, Department of Computer Science, Rio de Janeiro, Brazil

DOI: 10.21105/joss.01929

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Submitted: 02 December 2019
Published: 03 December 2019

Summary

There are several critical rotating equipment crucial to the industry, such as compressors, pumps and turbines. Computational mechanical models aim to simulate the behavior of such mechanical systems (Childs, 1993; Friswell, Penny, Garvey, & Lees, 2010; Gasch, Nordmann, & Pfützner, 2006; Ishida & Yamamoto, 2012; Vance, Zeidan, & Murphy, 2010). These models are used to support research and decision making. To this purpose, we present ROSS, an open source library written in Python for rotordynamic analysis.

Concerning rotordynamics softwares, there are some commercial finite element softwares that have a rotordynamic module ("ANSYS - Mechanical Rotordynamics," 2019; "COMSOL - Rotordynamics Module," 2019), some softwares based on a commercial platform (Matlab) ("Dynamics of Rotating Machines," 2019; "MADYN 2000," 2019), and others that are independent ("ROTORINSA," 2019; "XLTRC2," 2019). To use these softwares one needs to buy licenses, and they are not intended to be developed in a collaborative public manner.

ROSS allows the construction of rotor models and their numerical simulation. Shaft elements, as a default, are modeled with the Timoshenko beam theory, which considers shear and rotary inertia effects, and discretized by means of the Finite Element Method (Friswell et al., 2010). Disks are assumed to be rigid bodies, thus their strain energy is not taken into account. And bearings/seals are included as linear stiffness/damping coefficients.

After defining the element matrices and assembling the global matrices of the system, ROSS draws the rotor geometry, runs simulations, and obtains results in the form of graphics. It performs several analyses, such as static analysis, Campbell Diagram, mode shapes, frequency response, and time response.

The general form of the equation for the system, after matrix assembly is

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{C}(\Omega)\dot{\mathbf{q}}(t) + \omega\mathbf{G}\dot{\mathbf{q}}(t) + \mathbf{K}(\Omega)\mathbf{q}(t) = \mathbf{f}(t)\,, \tag{1}$$

where $\mathbf{q}$ is the generalized coordinates of the system (displacements and rotations), $\mathbf{M}$, $\mathbf{K}$, $\mathbf{C}$ and $\mathbf{G}$ are the mass, stiffness, damping and gyroscopic matrices, Ω is the excitation frequency, ω is the rotor whirl speed and $\mathbf{f}$ is the generalized force vector.

We have built the package using main Python packages such as NumPy (Walt, Colbert, & Varoquaux, 2011), SciPy (Jones, Oliphant, & Peterson, 2001) and Bokeh (Bokeh Development Team, 2019).

Besides the documentation, a set of Jupyter Notebooks is available for the tutorial and some examples. Users can also access these notebooks through a Binder server.

As an example, Figure 1 shows a centrifugal compressor modeled with ROSS.

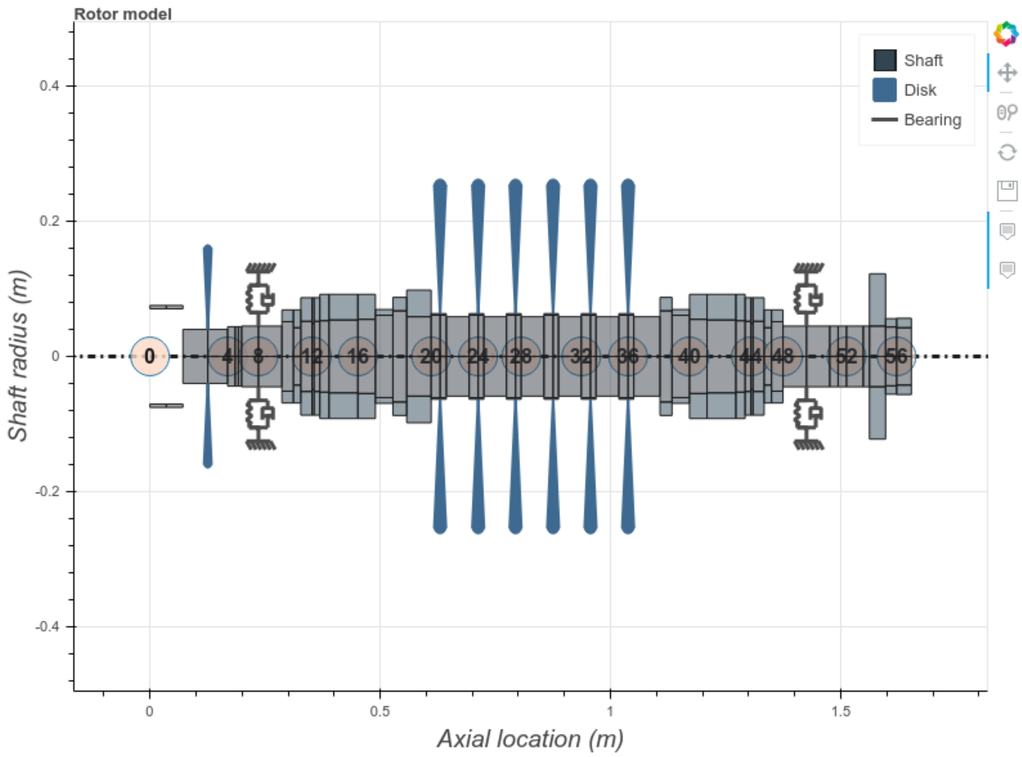

Figure 1: Centrifugal Compressor modeled with ROSS.

The shaft elements are in gray, the impellers represented as disks are in blue and the bearings are displayed as springs and dampers. This plot is generated with Bokeh, and we can use the hover tool to get additional information on each element.

One analysis that can be carried out is the modal analysis. Figure 2 shows the Campbell Diagram generated for this compressor; the natural frequencies and the log dec vary with the machine rotation speed.

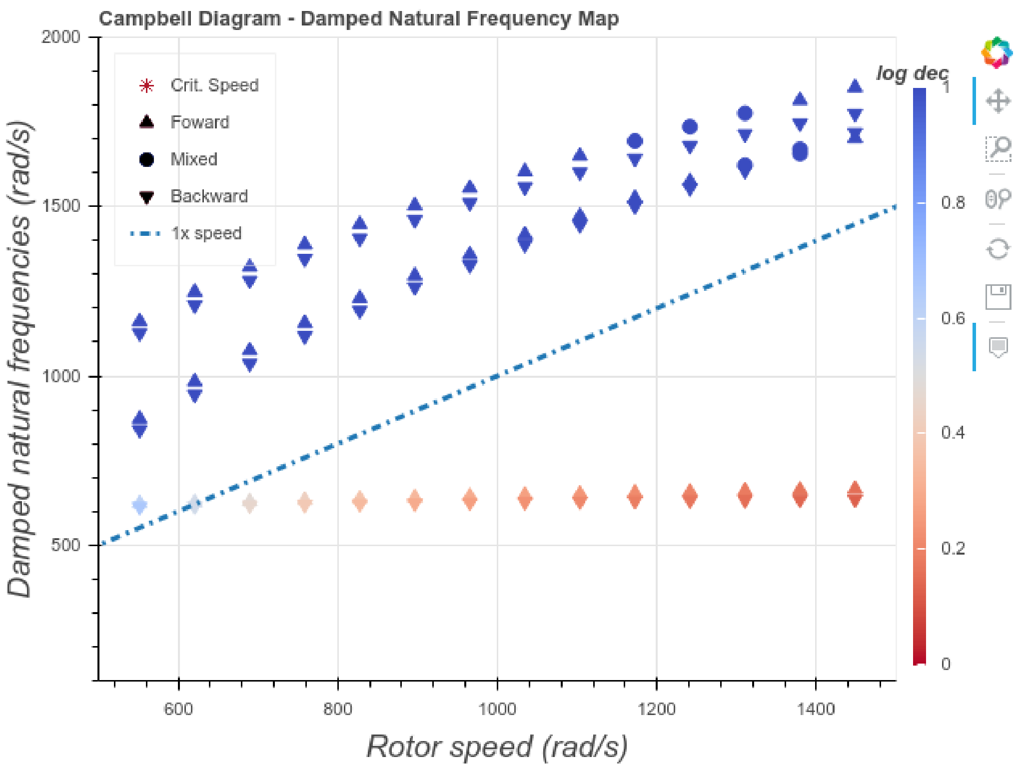

Figure 2: Campbell Diagram for the Centrifugal Compressor.

The Campbell Diagram is one of the results that can be obtained from the model, other types of analysis can be found on the documentation. The software is extensible and new elements or new types of analysis can be easily included.

Acknowledgements

We acknowledge that ROSS development is supported by Petrobras, Universidade Federal do Rio de Janeiro (UFRJ) and Agência Nacional de Petróleo, Gás Natural e Biocombustíveis (ANP).

References

ANSYS - Mechanical Rotordynamics. (2019). Retrieved from https://www.ansys.com/services/training-center/structures/ansys-mechanical-rotordynamics

Bokeh Development Team. (2019). *Bokeh: Python library for interactive visualization*. Retrieved from https://bokeh.org/

Childs, D. (1993). *Turbomachinery rotordynamics: Phenomena, modeling, and analysis*. John Wiley & Sons.

COMSOL - Rotordynamics Module. (2019). Retrieved from https://www.comsol.com/blogs/tag/rotordynamics-module

Dynamics of Rotating Machines. (2019). Retrieved from http://www.rotordynamics.info/

Friswell, M., Penny, J., Garvey, S., & Lees, A. (2010). *Dynamics of rotating machines*. Cambridge University Press. doi:10.1017/CBO9780511780509

Gasch, R., Nordmann, R., & Pfützner, H. (2006). *Rotordynamik*. Springer. doi:10.1007/978-3-662-09786-1

Ishida, Y., & Yamamoto, T. (2012). *Linear and nonlinear rotordynamics*. Wiley Online Library. doi:10.1002/9783527651894

Jones, E., Oliphant, T., & Peterson, P. (2001). SciPy: Open source scientific tools for Python.

MADYN 2000. (2019). Retrieved from http://www.delta-js.ch/en/software/

ROTORINSA. (2019). Retrieved from http://lamcos.insa-lyon.fr/logiciels_rotorinsa.php?L=2

Vance, J. M., Zeidan, F. Y., & Murphy, B. G. (2010). *Machinery vibration and rotordynamics*. John Wiley & Sons. doi:10.1002/9780470903704

Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, *13*(2), 22–30. doi:10.1109/MCSE.2011.37

XLTRC2. (2019). Retrieved from https://turbolab.tamu.edu/trc-software/