

# INTERACTIVE JUPYTER NOTEBOOKS IN EDUCATION: CASE STUDY OF AN AIRCRAFT ENGINE COMPRESSOR

*Nina Kumer, Matous Cejnek, Adam Peichl*

*Ústav přístrojové a řídicí techniky, ČVUT v Praze, Fakulta strojní, Praha*

*Abstrakt: V tomto projektu je popsáno řešení teoretického problému a následná tvorba interaktivního souboru jupyter notebook pro vysvětlení problémů. Teoretický problém představený v tomto článku je studie kompresoru leteckého motoru. Problém je řešen pomocí dynamické analýzy. Všechny potřebné odvození a rovnice jsou prezentovány v tomto článku. Nakonec jsou tyto rovnice vyřešeny pomocí nástroje jupyter notebook.*

*Klíčová slova: jupyter notebook, dynamická analýza, modální souřadnice, periodická síla, impulzní síla*

*Abstract: With this project we want to show a way how to solve theoretical problem, creating an interactive Jupyter Notebook. For that purpose, we decided to study an aircraft engine compressor. Given task is theoretical, but it could be easily used in real-life situation. We start solving our problem by studying theoretical background of dynamic analysis. It contains deriving all the equations needed, calculation eigenvalues and eigenvectors, transformation into modal coordinates, and harmonic and impulse analysis. Derived equations are later on solved in Jupyter Notebook, using NumPy. All the graphs are created with Matplotlib.pyplot.*

*Keywords: jupyter notebook, dynamic analysis, modal coordinates, periodic force, impulse force*

## 1 Introduction

The purpose of this project is to show a possible way how to perform dynamic analysis on a practical case of an aircraft engine compressor. We are examining multidegree-of-freedom system, its modal analysis and response under a periodic and non-periodic (impulse) force.

First, theoretical research was made and important equations were derived. Actual computation was done with Python in Jupyter Notebook [1]. It is an open-source web application that allows you to create and share documents that contain live code, equations, visualisations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualisation, machine learning, and much more. For the needs of this project, the most important packages are NumPy, which provides a high-performance multidimensional array object and tools for working with these arrays [2], and Matplotlib.pyplot, which is designed to be able to create simple and complex plots with a few command [3].

This project is important because it shows that once the theoretical equations are delivered, Python and Jupyter Notebook are a very useful tools for computing and visualising the actual results. That makes them more understandable and lifts our knowledge on a higher level.

## 2 Task

The explanation of the process of solving the problem is in this section. In subsection 2.1 the definition and goals are given. We start solving the problem in the next subsection, 2.2, where equations of motion are derived. We use them later in the subsection 2.3, where eigenvalues and eigenvectors are calculated. In subsection 2.4 physical coordinates are transformed into modal coordinates. We represent a continuous periodic function  $M_1(t)$  with Fourier series in subsection 2.5. Later on, we determine a steady-state response of the system. In the last subsection 2.6, we analyse a response of the system on impulse force, using convolution integral.

## 2.1 Definition and goals

Compressor is made of  $N$  disks with moment of inertia  $J_i$ . Free lengths of the shaft between the disks should be modeled as a torsion spring with coefficient  $k_i$ . Moment of inertia of each disk and spring coefficients are described as

$$J_i = J_0 e^{1-0.3i},$$

$$k_i = k_0 e^{1-0.3i},$$

where the parameters are given as follows:

Tab. 1: Given parameters

<i>parameter</i>	<i>unit</i>	<i>value</i>	<i>name/description</i>
$J_0$	Nm/rad	1400	moment of inertia
$M_0$	kgm <sup>2</sup>	0.4	moment
$N$	/	8	number of disks
$T$	s	0.17	time of 1 period
$t_0$	s	1.06	time, used when defining $M_2(t)$
$t_k$	s	4.66	time, used when calculation convolution integral

### Goals:

1. Determine equations of motion.
2. Determine eigenfrequencies and eigenvectors of the system. Graphically show eigenshapes. Normalise eigenvectors.
3. Make a transformation into modal coordinates.
4. Because of an injury on the first and the second part of the compressor, mentioned parts are periodically excited with function  $M_1(t)$ . Determine angular frequency of the system.

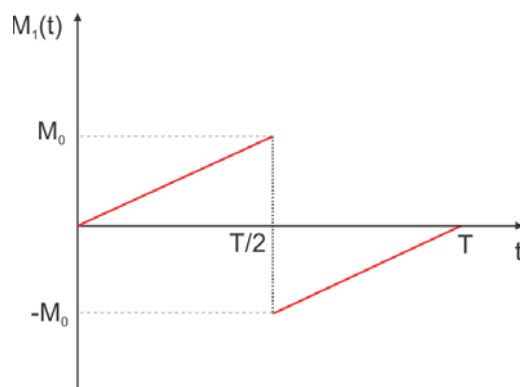


Fig. 1: Periodic function,  $M_1(t)$ .

5. Make approximation of the moment  $M_1(t)$  with Fourier series. Show it graphically.
6. Calculate the response of the system in steady state. Show it graphically.
7. On the system is now acting impulse  $M_2(t) = M_0 \tanh(-2\frac{t}{t_0})$ . Show the response of the system graphically.

## 2.2 Equations of motion

In order to be able to start solving any problem, we have to first make a model. It should be as simple as possible, but we should not neglect its important features.

If we make a discretization of a compressor with  $N$  elements, then this system has  $N$  degrees of freedom. In our case, that means 8 degrees of freedom.

The equations of motion of a vibrating system can be often derived in terms of generalized coordinates by the use

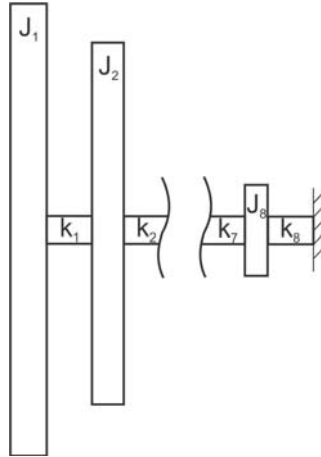


Fig. 2: Simplified model.

of Lagrange's equations. To be able to use them, we have to calculate a kinetic energy of each disk and potential energy of each torsion spring [4] as follows:

$$E_{kin} = \frac{1}{2} \sum_{n=1}^N J_n \dot{\varphi}_n^2, \quad (1)$$

$$E_{pot} = \frac{1}{2} \left[ \sum_{i=1}^{N-1} k_i (\varphi_i - \varphi_{i+1})^2 \right] + k_N \varphi_N^2. \quad (2)$$

We combine both expressions in Lagrangian - a function that summarizes the dynamics of an entire system. It is defined as  $L = E_{kin} - E_{pot}$ . Lagrange's equations can be now stated as

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\varphi}_i} \right) - \left( \frac{\partial L}{\partial \varphi_i} \right) = 0, \quad (3)$$

We obtain a system of  $N$  equations of motion. We write them in a matrix form as

$$[M] \{\ddot{\varphi}\} + [K] \{\varphi\} = \{0\}, \quad (4)$$

where  $[M]$  and  $[K]$  represent a mass and stiffness matrix, respectively;  $\varphi$  and  $\ddot{\varphi}$  are a twist and acceleration vector, respectively.

$$M = \begin{bmatrix} J_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & J_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & J_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & J_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & J_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & J_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & J_8 \end{bmatrix} = \begin{bmatrix} 0.806 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.597 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.442 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.327 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.243 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.180 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.133 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.099 \end{bmatrix}$$

$$[K] = \begin{bmatrix} k_1 & -k_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -k_2 & k_2 + k_3 & -k_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -k_3 & k_3 + k_4 & -k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_4 & k_4 + k_5 & -k_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -k_5 & k_5 + k_6 & -k_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_6 & k_6 + k_7 & -k_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & -k_7 & k_7 + k_8 \end{bmatrix} = \begin{bmatrix} 2819 & -2819 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2819 & 4908 & -2089 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2089 & 3636 & -1547 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1547 & 2693 & -1146 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1146 & 1995 & -849 & 0 & 0 \\ 0 & 0 & 0 & 0 & -849 & 1478 & -629 & 0 \\ 0 & 0 & 0 & 0 & 0 & -629 & 1095 & -466 \\ 0 & 0 & 0 & 0 & 0 & 0 & -466 & 811 \end{bmatrix}$$

### 2.3 Eigenvalue problem

We can find the solution of (4) by assuming a solution of the form  $\varphi_i(t) = \theta_i T(t)$ , where  $\theta_i = \text{const.}$  and  $T$  is a function of time. We substitute this new form in (4) and after some mathematical steps, we obtain eigenvalue problem equation:

$$[-\omega^2[M] + [K]]\theta = \{0\} \quad (5)$$

where  $\omega^2$  represents eigenvalue and its square root  $\omega$  is natural frequency of the system. Equation (5) represents a set of  $N$  linear homogeneous equations in the unknowns  $\theta_i (i = 1, 2, \dots, N)$ . For a nontrivial solution, the determinant of the coefficient matrix must be zero:

$$\det[-\omega^2[M] + [K]] = 0 \quad (6)$$

The expansion of (6) leads to an  $N$ -th order polynomial equation in  $\omega^2$ . The solution (roots) gives  $N$  values of  $\omega^2$ ; it can be shown that all roots are real and positive, when matrices  $[M]$  and  $[K]$  are symmetric and positive definite. If  $\omega_1^2, \omega_2^2, \dots, \omega_N^2$  denote  $N$  roots in ascending order of magnitude, their positive square roots give  $N$  natural frequencies of the system. The lowest value,  $\omega_1$ , is called the fundamental or first natural frequency. In order to calculate the values of natural frequencies, we first substitute  $\omega^2 = \lambda$ , then eq. (5) becomes:

$$[[M]^{-1}[K] - \lambda[I]]\theta = \{0\} \quad (7)$$

For solving this equation in Python we don't have to calculate when the determinant is going to be zero, but we'd rather compute dynamical matrix as

$$[D] = [M]^{-1}[K] \quad (8)$$

and then with the built-in function `numpy.linalg.eig` we obtain eigenvalues and eigenvectors of this matrix.

Tab. 2: Natural frequencies

$\omega_{01}$	$\omega_{02}$	$\omega_{03}$	$\omega_{04}$	$\omega_{05}$	$\omega_{06}$	$\omega_{07}$	$\omega_{08}$
6.5391	34.2906	56.6362	76.8568	94.3898	108.6722	119.2306	125.7111

### 2.4 Modal coordinates

We start the transformation into modal coordinates by referring to the eq. (4). Physical coordinates  $\varphi$  are now being replaced with modal coordinates as

$$\{\varphi\} = [\Phi] \cdot \{\eta\}, \quad (9)$$

where  $[\Phi]$  represents modal matrix, which contains all eigenvectors;  $[\eta]$  represents a vector of modal coordinates. If we insert (9) in (4), and multiply the expression with  $[\Phi]^T$  we obtain

$$[\overline{M}] \{\ddot{\eta}\} + [\overline{K}] \{\eta\} = \{0\}, \quad (10)$$

where  $[\overline{M}]$  and  $[\overline{K}]$  represent modal mass and modal stiffness matrix, respectively. They are both diagonally. The matrices are estimated as follows:

$$[\overline{M}] = [\Phi]^T \cdot [M] \cdot [\Phi] \quad (11)$$

$$[\overline{K}] = [\Phi]^T \cdot [K] \cdot [\Phi] \quad (12)$$

$$[\overline{M}] = \begin{bmatrix} 2.373 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.339 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.951 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.212 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8.706 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 14.250 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 30.253 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 116.911 \end{bmatrix}$$

$$[\overline{K}] = \begin{bmatrix} 0.0102 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5102 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5880 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.6692 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.7563 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16.8284 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 43.0079 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 184.7850 \end{bmatrix} \cdot 10^4$$

## 2.5 Harmonic analysis; Fourier series

Any periodic function of time can be represented by Fourier series as an infinite sum of sine and cosine terms. If  $M_1(t)$  is a periodic function with period  $T$ , its Fourier series representation is given by:

$$M_1(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \quad (13)$$

$$a_n = \frac{2}{T} \int_0^T M_1(t) \cos(n\omega t) dt \quad (14)$$

$$b_n = \frac{2}{T} \int_0^T M_1(t) \sin(n\omega t) dt \quad (15)$$

Because function  $M_1(t)$  is a odd function (it holds:  $-M_1(t) = M_1(-t)$ ), coefficients  $a_0$  and  $a_n$  are equal to zero. Even though Fourier series is an infinite sum, we can approximate most periodic functions with the help of only a few terms. In Python, an interactive plot was made. User can change the number of terms and observe how the graph changes. We decided that 50 terms is the best compromise between the number of used terms and fitting the real function, Fig.3. Where function  $M_1(t)$  is not continuous, we get a bigger deviation while using Fourier series, and we can not fix that, even if we increase the number of terms.

Steady-state solution of eq. (13) is given by

$$\{x_{P3}\} = \sum_{n=1}^{\infty} \{x_{nP3}\} \quad (16)$$

$$\{x_{nP3}\} = \{X_{nP3} \sin(n\omega t)\} \quad (17)$$

$$X_{nP3}^{(j)} = \frac{b_n}{\omega_{0j}} \cdot \frac{1}{1 - \left(\frac{n\omega}{\omega_{0j}}\right)^2} \quad (18)$$

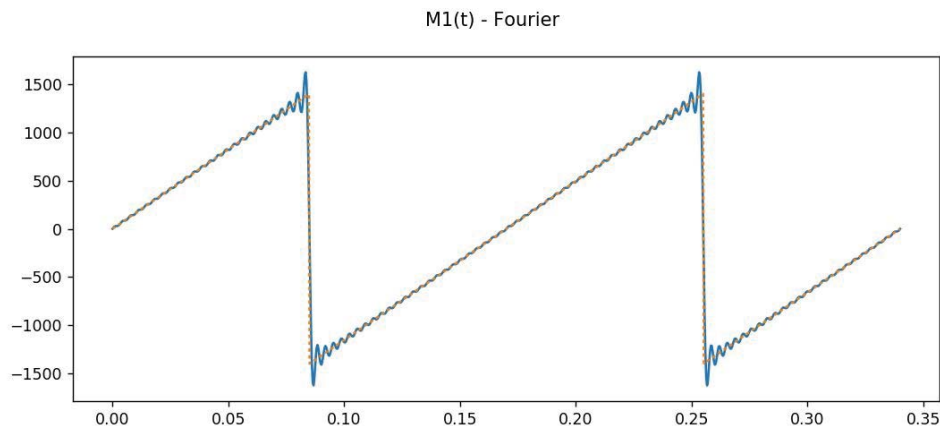
n: 

Fig. 3: Function  $M_1$  (orange) approximated with Fourier series (blue). We used 50 terms.

We have to keep in mind that we do not take into account any damping.

Because we are still dealing with multidegree-of-freedom system, we go back into the modal space. The equation is going to be similar to the eq. (10), only this time on the right side there is not a vector of zeros.

$$[\overline{M}] \{\ddot{\eta}\} + [\overline{K}] \{\eta\} = [\Phi^T] \cdot \{M_1\} \quad (19)$$

$$[I] \{\ddot{\eta}\} + [\overline{M}]^{-1} \cdot [\overline{K}] \{\eta\} = [\overline{M}]^{-1} \cdot [\Phi^T] \cdot \{M_1\} \quad (20)$$

The first and the second part of the compressor are periodically excited with function  $M_1(t)$ . We have to create a vector  $M_{1,act}$ , which has values of  $M_1$  only in the first two rows, others are zero. In that way we obtain vector  $b_n$

$$\{b_n\} = [\overline{M}]^{-1} \cdot [\Phi^T] \cdot \{M_{1,act}\} \quad (21)$$

The last step is a transformation of modal coordinates back into physical, using the relation (9), and plotting a graph.

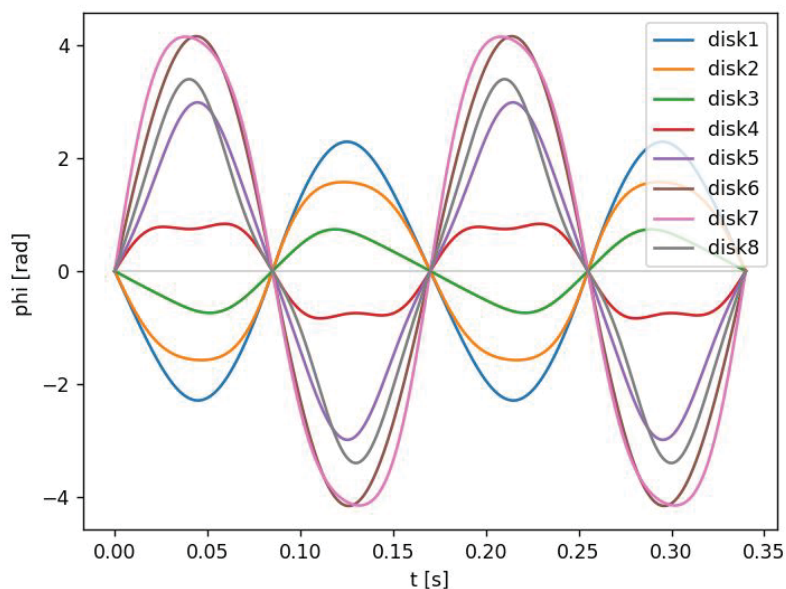


Fig. 4: Steady-state response.

## 2.6 Non-periodic force - impulse

We want to analyse the response of our system, when there is an acting impulse moment  $M_2(t) = M_0 \tanh(-2\frac{t}{t_0})$ .

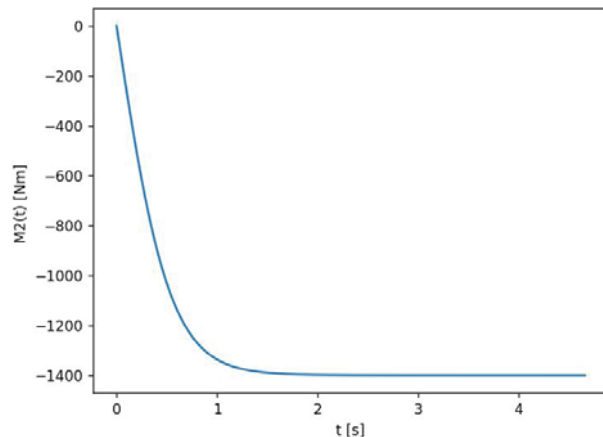


Fig. 5: Function  $M_2(t)$ .

We are solving this problem by the use of convolution integral [5]:

$$x(t) = \int_0^t m_2(\tau)g(t-\tau)d\tau \quad (22)$$

In the case of no damping, function  $g(t)$  looks like:

$$g(t) = \frac{1}{\omega_0} \sin(\omega_0 t) \quad (23)$$

We have to transform vector  $M_2$  into the modal coordinates in the same way as in the previous case of periodic function, eq. (20). Eq. (22) then looks like

$$x(t) = \int_0^t \{M_{2,mod}\}(\tau)g(t-\tau)d\tau \quad (24)$$

The last step is a transformation of modal coordinates back into physical, using the relation (9), and plotting a graph.

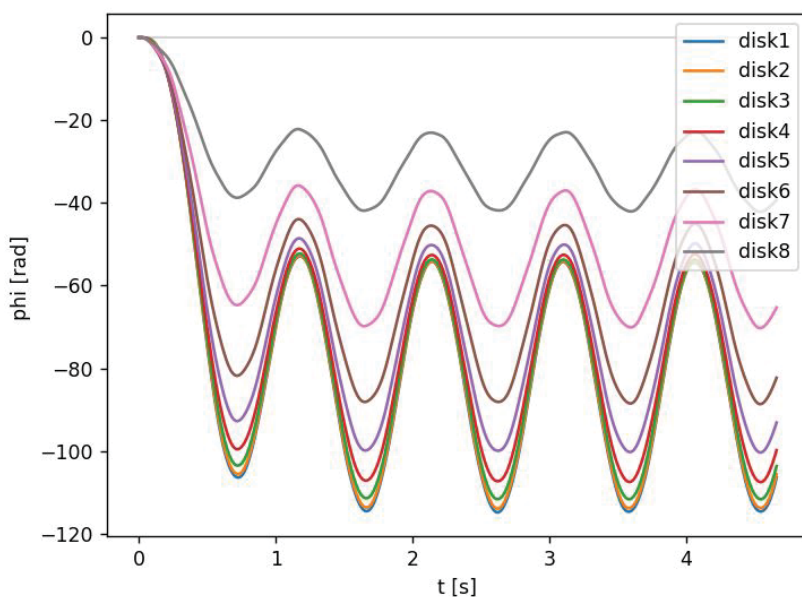


Fig. 6: Response of the system after impulse.

### 3 Conclusion

This project consists of two main parts. First, we study theoretical background of the problem and derive all the equations needed. Once that was done, we solved them with Python, creating a Jupyter Notebook. The most important packages we used were NumPy and Matplotlib.pyplot. With Numpy we created vectors and matrices and performed various operations: summation of arrays, dot products of two arrays, transposition of matrix, computation of eigenvalues and eigenvectors, sorting of values, and many more. We used Matplotlib.pyplot for plotting some graphs, for creating figures, containing several subplots and for creating an interactive plot. All the results we were able to predict from the theory were confirmed with calculations, for example: modal mass and stiffness matrices should be diagonal and they actually are. All plotted graphs also make sense, so we can conclude that our work was done correctly.

### Acknowledgement

Main author, N. Kumer, works on this project during her stay at CTU. Her main goal was to understand the theoretical background of the problem, because Mechanics is her major. Beside that she also gained a lot of knowledge about Python, and she will use it in the future for solving similar problems.

N. Kumer would like to thank the other two co-authors for all the support and help.

The related jupyter notebook can be obtained on request from authors (matous.cejnek@fs.cvut.cz).

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS18/177/OHK2/3T/12.

### Literature

- [1] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90, 2016.
- [2] Travis E Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007.
- [3] Niyazi Ari and Makhamadsulton Ustazhanov. Matplotlib in python. In *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, pages 1–6. IEEE, 2014.
- [4] Singiresu S. Rao and Fook Fah Yap. *Mechanical vibrations*, volume 4. Prentice hall Upper Saddle River, 2011.
- [5] M. Boltežar. Nihanje linearnih mehanskih sistemov zaradi udarne motnje. 2004.





**Selected article from**

**Tento dokument byl publikován ve sborníku**

**Nové metody a postupy v oblasti přístrojové  
techniky, automatického řízení a informatiky 2019  
New Methods and Practices in the Instrumentation,  
Automatic Control and Informatics 2019  
27. 5. – 29. 5. 2019, Zvíkovské Podhradí**

**ISBN 978-80-01-06617-1**

Web page of the original document:

<http://iat.fs.cvut.cz/nmp/2019.pdf>

Obsah čísla/individual articles:

<http://iat.fs.cvut.cz/nmp/2019/>

Ústav přístrojové a řídicí techniky, FS ČVUT v Praze, Technická 4, Praha 6