CALCULATING TOOL FOR THE SPRING CONSTANT OF A SERPENTINE FLEXURE USED IN INERTIAL MEMS DEVICES

Petre NEGREA, Vlad Aurelian VĂDUVESCU

University of Craiova, Romania (pnegrea@elth.ucv.ro, vlad93v@yahoo.com)

DOI: 10.19062/2247-3173.2019.21.21

Abstract: *A Matlab/Simulink tool for calculating the constant of elasticity of a serpentine flexure is here presented. The tool is developed for four different equations, in terms of two axis, and even or odd number of meanders. Here are shown four types of microsuspension used in the MEMS devices, the component elements and structure of a serpentine flexure and the Matlab/Simulink models resulted from the mathematical equations. The simulations results, compared with those in the literature, confirms that the developed tool is correctly implemented.*

Keywords: *MEMS, microsuspension, serpentine flexure.*

1. INTRODUCTION

The tendency of systems miniaturization has led to the revolution of any product, but also brought with it some major problems, mainly from the manufacturing point of view.

Miniaturization of the systems led to new technologies such as MEMS (micro-electro-mechanical-systems) and NEMS (nano-electro-mechanical-systems). In the field of inertial navigation, from the point of view of aerospace, transport and naval engineering, miniaturized devices occupy an important place due to accelerometers and gyros, sensors that are used to detect and measure movements. MEMS actuators and sensors, due to their small size, brought the advantages of a low mass and volume, low energy consumption, greatly reducing the mass and size of a vehicle without sacrificing functionality. The major disadvantages are that they are susceptible to parasitic vibrations, mechanical and thermal shocks, and require a considerable amount of optimization calculation [1, 2, 3].

An important component in the study of inertial sensors, and in general of MEMS devices, is the elastic element. The microsuspensions (elastic element) used in the MEMS field can vary, both in shape and as the material of which they are composed, depending on their use [4].

2. COMMONLY USED ELASTIC STRUCTURES FOR THE INERTIAL SENSORS

The most commonly used elastic structures for inertial sensors are shown in Fig. 1 [4, 5]. Thus, four flexion patterns can be observed: a) fixed-fixed flexure; b) crab leg flexure; c) folded flexure; d) serpentine flexure. The four configurations present a seismic mass attached at the ends with different elastic elements.

The calculation for the elasticity constant of each configuration is rigorous and can be found in [4, 5].

According to the Fig. 1, can be observed an increase in structural complexity in terms of the number of components and their design, thus, the architecture of the model d) can provide a higher sensitivity from the input measurement point of view. Also, the rigidity of the serpentine flexure can be greatly adjusted, in the design phase, by changing the number of items, depending on the utility.

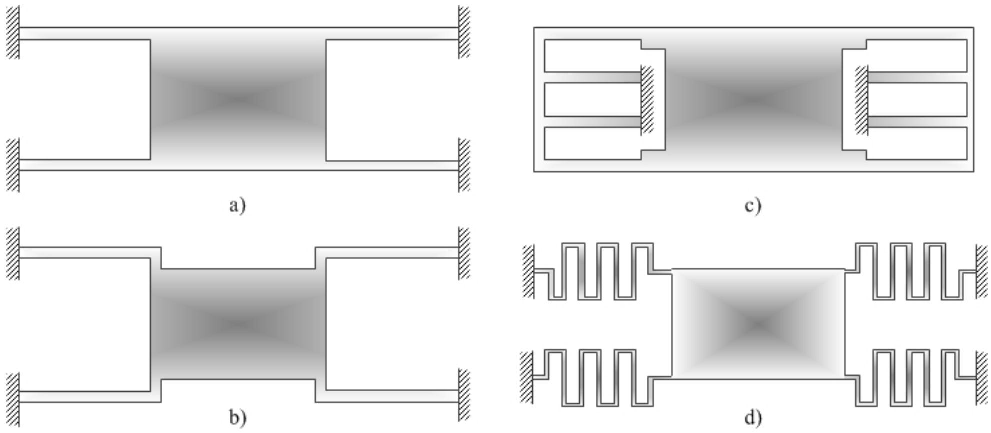

FIG. 1. Elastic structures for inertial sensors: a) fixed-fixed flexure; b) crab leg flexure; c) folded flexure; d) serpentine flexure

3. MATLAB/SIMULINK MODEL

In the calculus of the acceleration for an inertial accelerometer, an important role has the elasticity constant of the sensing element, this being directly related to the movement of the seismic mass [3]. The elastic force acting on the seismic mass is:

$$F_e = k \cdot x \tag{1}$$

where, F_e is the elastic force, k is the constant spring and x is the displacement of the proof mass.

If an acceleration acts on the seismic mass, for the simple configuration [seismic mass → elastic element], due to the Newton's 2nd law, in eqilibrum can be written:

$$m \cdot a = k \cdot x \tag{2}$$

So, it can be observed that in the determination of acceleration, spring constant plays an important role.

For this purpose, the paper presents a Matlab/Simulink calculation tool for the spring constant of the serpentine flexure. A schematic representation of the serpentine is shown in Fig. 2, where in a) is illustrated the overall configuration with four springs attached to the proof mass, that intuitively hints the fact that they can be adjusted in number of its elements, and in b) is detailed one typical spring [4, 5, 6, 7, 8].

The model is composed of several meanders, each meander has the same length *a* and width *b*, more precisely each meander is composed of 2 segments of different lengths, vertically *b*, horizontal *a*. Each segment of the meanders, according to right picture from the fig. b) is of width w_a for the horizontal segment and w_b for the vertical segment, and thickness *h* [4].

The coordinate system for which the calculation was performed was considered as the *x*-axis along the segment *b*, positive upwards, the *y*-axis along segment *a*, positive from the fixed end to the seismic mass, and the *z*-axis along the thickness *h*, perpendicular to the two axis.

The spring constants for which the Matlab/Simulink model has been created are only those in the *x* and *y* directions.

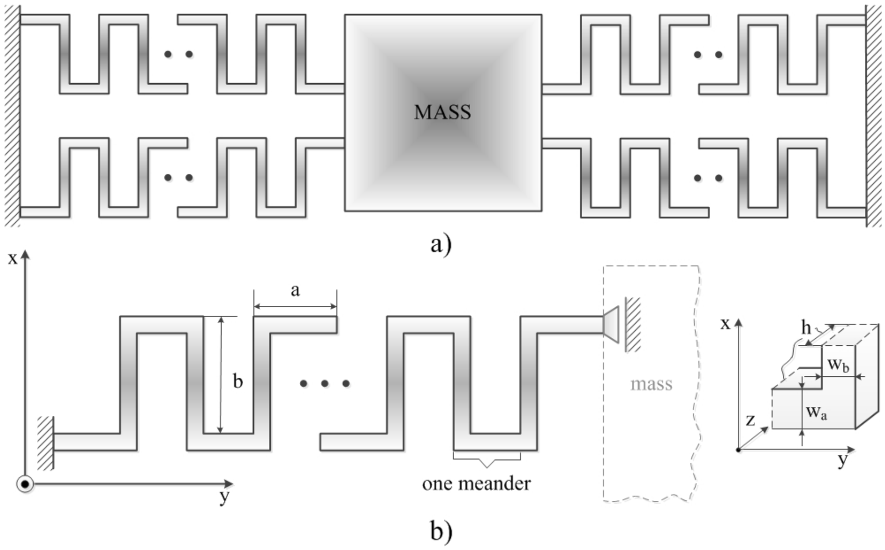

FIG. 2 Serpentine schematic representation: a) overall configuration with four springs; b) one typical spring

The general equation of the spring constant changes according to the parity of the number of meanders. Thus, according to [4], for even *n*:

$$k_x = \frac{48EI_{z,b}\left[(3\overline{a}+b)n-b\right]}{a^2n\left[(3\overline{a}^2+4\overline{a}b+b^2)n^3-2b(5\overline{a}+2b)n^2+(5b^2+6\overline{a}b-9\overline{a}^2)n-2b^2\right]}, \tag{3}$$

$$k_y = \frac{48EI_{z,b}\left[(\overline{a}+b)n^2-3bn+2b\right]}{b^2\left[(3\overline{a}^2+4\overline{a}b+b^2)n^3-2b(5\overline{a}+2b)n^2+(5b^2+6\overline{a}b-9\overline{a}^2)n-2b^2\right]} \tag{4}$$

where, k_x, k_y are the spring constants in the *x* and *y* directions, *n* is the number of meanders, *E* is the Young modulus and,

$$\overline{a} = \frac{I_{z,b}\cdot a}{I_{z,a}}; \tag{5}$$

$$I_{z,b} = \frac{h\cdot w_b^3}{12}; \tag{6}$$

$$I_{z,a} = \frac{h\cdot w_a^3}{12}; \tag{7}$$

with $I_{z,a}$, $I_{z,b}$ the moment of inertia.

For odd *n*:

$$k_x = \frac{48EI_{z,b}}{a^2n\left[(\overline{a}+b)n^2-2bn+2b\right]}, \tag{8}$$

$$k_y = \frac{48EI_{z,b}\left[(\bar{a}+b)n-b\right]}{b^2(n-1)\left[(3\bar{a}^2+4\bar{a}b)n+3\bar{a}^2-b^2\right]}. \tag{9}$$

Using the equations (3-9) a Matlab/Simulink model was created and calculates the spring constant on two axes, *x* and *y*. Given that the equations differ according to the parity of the number of meanders and the fact that there are two axes, 4 distinct blocks were created (Fig. 3). For the respective blocks, grouped into one (Fig. 4) that represents the final calculation tool, a mask was implemented (Fig. 5) to introduce the constructive parameters of the analyzed structure. The user can easily enter these parameters manually, without having to introduce the equations.

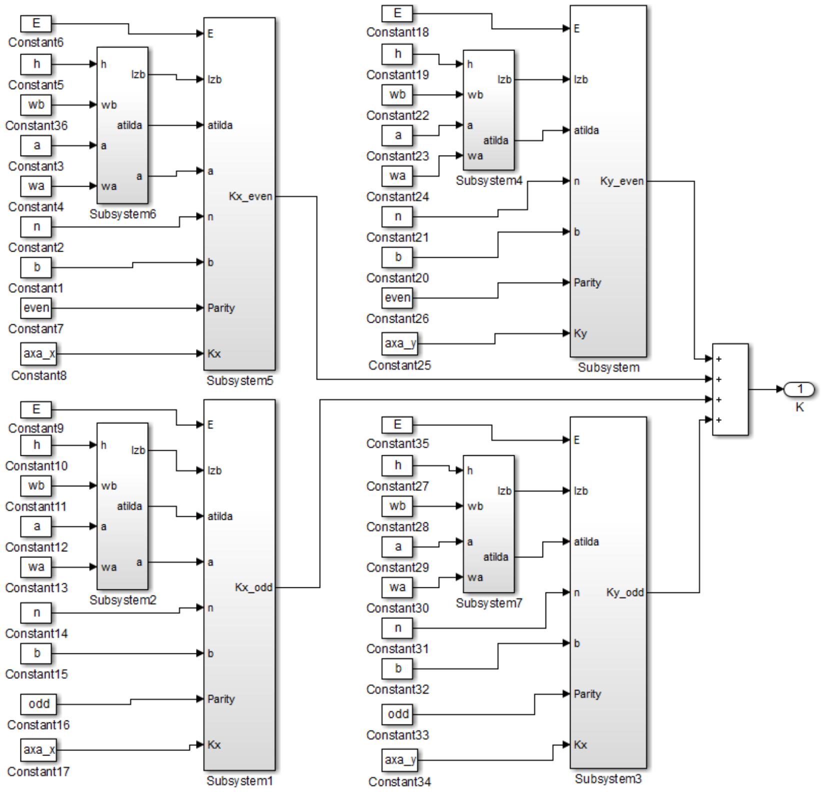

FIG. 3 Matlab/Simulink model for the serpentine flexure

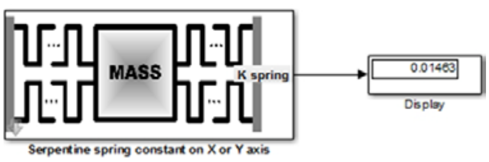

FIG. 4 Serpentine flexure Matlab/Simulink block

Also, with the number of meanders entered, the block automatically selects, for even or odd, the computing equation. The tool has a background process that allows this switch, from odd to even and vice versa. In Fig. 6 is shown a block diagram for the k_x equation.

The model was simulated using a set of parameters given in other specialized works [4, 6, 7]. After the simulations, identical results were obtained, which confirms that the developed calculation tool is correctly implemented.

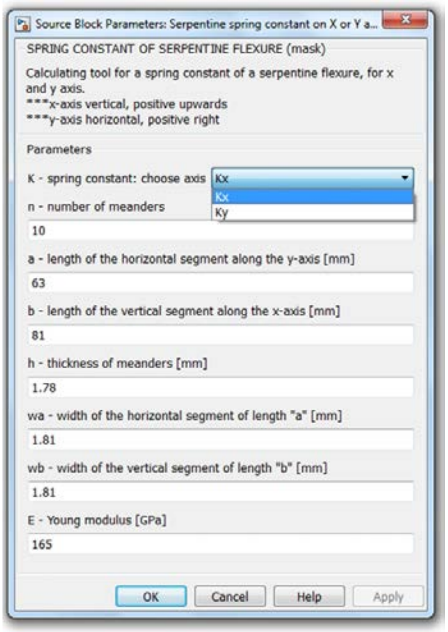

FIG. 5 Parameters settings

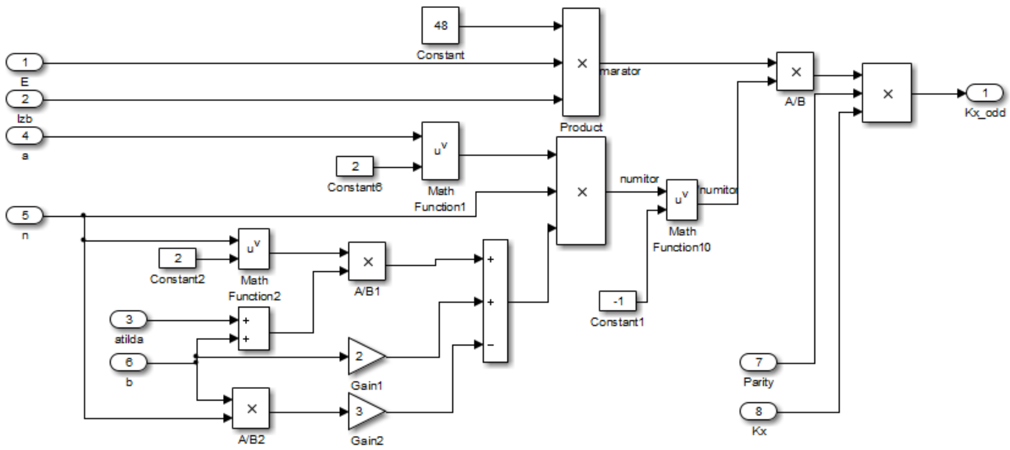

FIG. 6 K_x Matlab/Simulink block diagram

CONCLUSIONS

In the field of microsuspensions, different architectures are used, but a particular attention is focused on the serpentine configuration. This paper presents a calculating tool for the spring constant on two axis of a serpentine flexure, commonly used in inertial MEMS devices. Given the equations, a Matlab/Simulink model was created, which allows the user to manually introduce the constructive parameters and showing the final results and which can be used in other block diagrams. Because the equations differ, if the meanders number is odd or even and the computing is on two axes, 4 blocks were initial created, eventually included in one, which, due to a software written in Matlab, switches automatically, depending on the number of meanders and the selected axis. For a set a parameters given by the specialized literature, the simulation confirms that the implemented tool works correctly.

REFERENCES

[1] Sunil Kumar, *Design and Fabrication of Micromachined Silicon Suspensions*, Ph.D. Thesis, University of California, Irvine 2000;

[2] Wong Wai Chi, *Analysis of the Suspension Beam in Accelerometer for stiffness constant and resonant frequency by using Analytical and Numerical Investigation*, Master Science Dissertation, University Sains Malaysia, 2007;

[3] T.L. Grigorie, S*isteme de navigatie inertiala strap-down*, SITECH, Craiova, Romania, 2007

[4] G.K. Fedder, *Simulation of Microelectromechanical Systems*, Ph.D. Thesis, University of California at Berkeley, 1994;

[5] Nicolae Lobontiu and Ephrahim Garcia, *Mechanics of Microelectromechanical Systems*, 2005 Springer Science + Business Media, Inc., Print ©2005 Kluwer Academic Publishers

[6] Giuseppe Barillaro, Antonio Molfese, Andrea Nannini, and Francesco Pieri, *Analysis, Simulation and Relative Performances of two kinds of Serpentine Springs*, Journal of Micromechanics and Microengineering, February 2005;

[7] Winne Jerry, P.Anitha Saraswathi, *A novel design of serpentine structure for enhanced performance of MEMS based pressure sensors*, International Journal of Instrumentation, Control and Automation (IJICA), ISSN: 2231-1890, Volume-2, Issue-1, 2013;

[8] Dimitrios Peroulis, Sergio P. Pacheco, Kamal Sarabandi and Linda P.B. Katehi, *Electromechanical considerations in developing low-voltage RF MEMS Switches*, IEEE Transactions on microwave theory and techniques, VOL. 51, NO. 1, JANUARY 2003.