Performance Comparison in the Optimisation of a Parallel Robot Using Particle Swarm Optimisation

Catalin Boanta and Cornel Brisan

***Abstract*—Parallel robots have many industrial applications due to their well-known advantages as high operational speeds, stiffness and accelerations. One the other hand, their workspace is reduced compared to the size of the elements of the robot. Frequently, the design of parallel robots implies a large amount of variables and nonlinear equations. This is why, a human designer generally applies optimisation algorithms in order to obtain specific properties of the robot. If the number of variables involved in the optimisation is too high, the required computational times may be extremely increased, aspect that for some applications is unacceptable. This is why, the aim of this paper is to analyse the performance comparison in terms of efficiency and computational times of an optimisation problem with several numbers of variables included in the optimisation. The variable define the geometrical characteristics of a parallel robot used for a solid waste selection system. Also, the optimisation problem is implemented using a heuristic algorithm, namely the Particle Swarm Optimization.**

***Index Terms*—Robots, optimisation, particle swarm optimisation, workspace.**

I. Introduction

Parallel robots have become more and more common in many industrial applications mostly due to their well-known advantages they provide as high stiffness, accelerations and operational speeds.

On the other hand, parallel robots have disadvantages, an example being the workspace which is reduced compared to the size of the robot. Moreover, the design of the parallel robots implies a lot amount of experience and, in many cases, optimisation algorithms are used in the process.

Since a parallel robot is defined by a high number of variables (the geometrical characteristics) that define the structure, the optimisation of such a robot may lead to very high computational times (of the order of days), aspects that is, usually, unacceptable.

This is why, the aim of this paper is to analyze the performance comparison in terms of efficiency and computational times of an optimisation problem of a parallel robot for different numbers of variables included in the optimisation. The optimisation problem in implemented using the Particle Swarm Optimisation algorithm and the objective function evaluates the correlation between the workspace of the robot with an imposed one.

The application of the parallel robot analyzed in this paper is in a Solid Waste Selection System. The robot disposes the solid waste from a moving conveyor, this is why, it is desired that the workspace of the robot to be suitable for this application.

The paper is organized as follows: The Section II present recent advances regarding the optimisation of robots using heuristic approaches, namely the Particle Swarm Optimisation, the Section III illustrate the robot to be optimized, the Section IV presents the optimisation setup and results and the Section V exemplifies the conclusions.

II. Literature Review

The implementation of the optimisation of the parallel robot and the performance comparison is carried out using the Particle Swarm Optimisation, an evolutionary population-based search algorithm.

The Particle Swarm Optimization Algorithm is inspired also from nature. It is a stochastic optimization technique and it simulates the behavior of birds flocking and fish schooling. This algorithm is an evolutionary algorithm that searches the best solution by updating the current population. The members of the swarm "communicate", each member being aware of the best position among them.

There are few references that use the Particle Swarm Algorithms to optimize a parallel robot. One example is [1] that proposes the use of this algorithm in a single object optimization: minimization of the stiffness over a cubic usable workspace of the structure of a parallel kinematic. Also, the paper carries out a comparison between the Particle Swarm Optimization and Genetic Algorithms. The results indicate the fact that this algorithm presents an improved overall behavior than Genetic Algorithms. Despite this fact, only a limited number of generations has been taken into consideration. Also, the comparison does not mention anything regarding the computational times of these two algorithms. The reference [2] illustrates another example of kinematic optimization using Particle Swarm Algorithm. A multi object optimization is carried out, taking into consideration both the global compliance and global conditioning index of a parallel structure. Particle Swarm Optimization has been used to optimize a parallel structure also in [3]. The maximization of the Global Condition Index along the workspace has been considered the performance criteria. There are no mentions of the reasons of using Particle Swarm Optimization. Also, a finite number of generations have been considered when the optimization has been carried out. The paper [4] proposes a design optimization of a planar parallel robot considering the energy consumption as the cost function

The optimisation of a parallel robot is a stochastic problem

Manuscript received October 14, 2019; revised April 1, 2020. This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCDI-UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0086/ 22/2018, within PNCDI III.

Cătălin Boanta and Cornel Brișan are with the Technical University of Cluj-Napoca, Romania (e-mail: boanta_catalin@yahoo.com, cornel.brisan@mdm.utcluj.ro).

DOI: 10.7763/IJMO.2020.V10.753

due to the fact that several numerical values have to be considered into the optimization, values that are not known at this time of formulating the optimization problem of this parallel robot. Such examples of numerical values may be the Jacobian of the parallel robot and its determinant that are dependent of the configuration of the robot.

The Particle Swarm Optimisation has been investigated [5] regarding the effectiveness (i.e. the ability to find an optimal solution) and computational efficiency behaving better in solving non-linear unconstrained problems in comparison with the Genetics Algorithms

Considering all these aspects it may be concluded that Particle Swarm Optimisation is a suitable algorithm for optimisation problems of robots. Also, due to the high computational times of the algorithms, the Section IV illustrates a tradeoff between the number of variables involved in the optimisation, the computational times and the effectiveness and accuracy of the results.

III. Robot to be Optimized

The parallel robot analyzed in this paper is a subsystem of a Solid Waste Selection System. Solid waste produced by the inhabitants of cities from emerging economies has become one of the most relevant problem in the past years. The quantity of plastic, metal, paper or other type of solid waste is growing, aspect that may produce environmental issues [6] - [7].

This is why, an Automated Solid Waste Selection System would overcome the drawbacks of not recycling the solid waste. The concept of the system has been presented in the papers [8]-[10]. The system is composed by a Transportation System, Image Recognition and Sensorial System and a Robot Disposal System.

The Robot Disposal System is presented in the Fig. 1. The Robot 1 up to N places the Waste 1 up to N to the corresponding container.

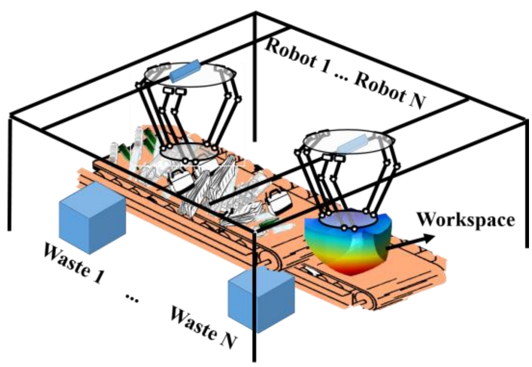

Fig. 1. Parallel robot to select unsorted waste from a conveyor [9].

In order to design the conveyor and the robotic system as optimal as possible, it has been considered that the required workspace of the robots has to be as wide as the conveyor the other dimensions (length, and height) are user imposed (as presented in the Fig. 2).

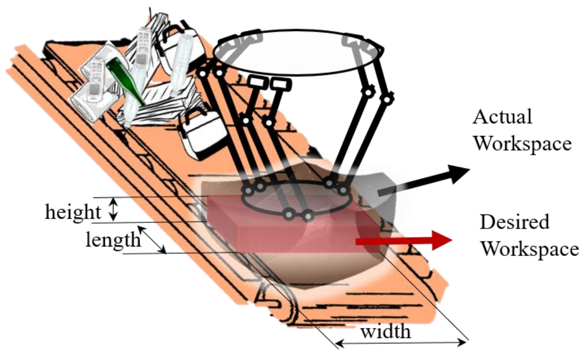

Fig. 2. Workspace of the robot (gray) and the required workspace(red).

The robot that compose the Robot Disposal System is a 6 DOF parallel manipulator with rotational actuators. The structure is composed by six RUS (rotational-universal-spherical) open kinematic loops that interconnect the fixed platform (on the top) to the mobile platform (on the bottom) with regard to the Fig. 2. The The robots that is optimized is a 6 DOF parallel robot with rotational actuators, composed by six identical rotational-universal-spherical kinematic open loops (RUS – from the bottom fixed plate to the top mobile plate). The architecture of the robot is presented in the Fig. 3.

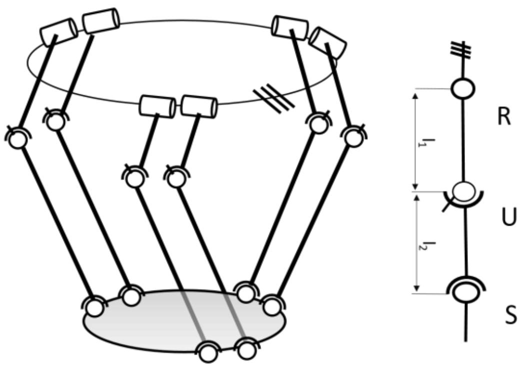

Fig. 3. Parallel robot to be optimized [9].

IV. Optimization Setup

The aim of the paper is to analyze a performance comparison regarding the effectiveness and computational times between the optimisations of the parallel robot with several numbers of variables to be optimized. The robot is optimized to achieve a workspace similar to an imposed one, as presented in [11] and in the Fig. 2.

Since a symmetric configuration of a parallel robots leads to better kinematic performance, 6 variables have been considered to be influencing the design of the robot, namely, the radiuses of the fixed and mobile platform (R and r, as seen in the Fig.4), the length of the mobile elements(l_1 and l_2, as presented in the Fig. 3) and the ratios between two consecutive angles of positioning the rotational joint on the fixed platform and on the mobile platform (as presented in the Fig. 4 and the equation.

$$ratio_d = \frac{\beta_1}{\beta_2} = \frac{\beta_3}{\beta_4} = \frac{\beta_5}{\beta_6} \tag{1}$$

$$ratio_u = \frac{\alpha_1}{\alpha_2} = \frac{\alpha_3}{\alpha_4} = \frac{\alpha_5}{\alpha_6} \quad (2)$$

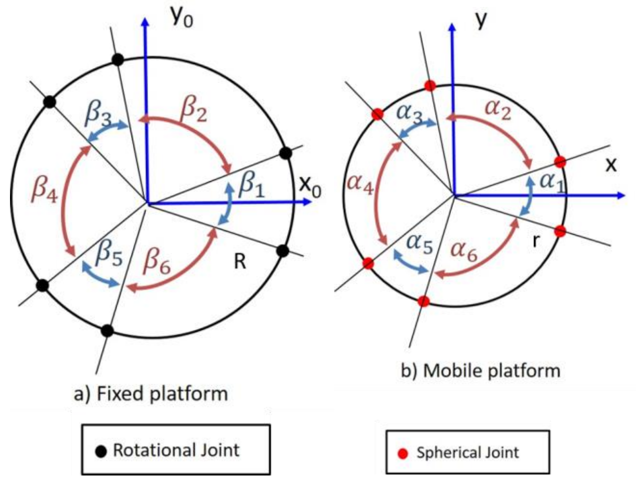

Fig. 4. Mobile and fixed platform (top view) [9].

Therefore, the number of variables that define each configuration of the robot is 6. In order to realize a comparison between effectiveness and computational times, three cases will be considered

1. Two design variables,
2. Four design variable
3. Six design variables.

A. Two Variables Setup

The design vector of the optimisation contains, in this case, the variables that influence the most the workspace of the parallel robot i.e. the length of the first and second elements l_1 and l_2 of the open rotational-universal-spherical open loops that interconnect the fixed and mobile platform. The design vector is given in the eq. (3).

$$x_{2\text{vars}} = (l_1, l_2) \quad (3)$$

The constraints imposed for the length of the elements are presented in the eq. (4).

$$\begin{aligned} &l_1 > 0.01[m] \\ &l_2 > 0.01[m] \\ &l_1 < 0.5[m] \\ &l_2 < 1.5[m] \end{aligned} \quad (4)$$

The matrix form of the equations (that is implemented in Matlab) is given in the eq. (5), where $A_{2\text{vars}}$ and $b_{2\text{vars}}$ are the matrices from the eq. (6) and (7).

$$A_{2\text{vars}} \cdot x^t_{2\text{vars}} < b_{2\text{vars}} \quad (5)$$

$$A_{2\text{vars}} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6)$$

$$b_{2\text{vars}} = \begin{bmatrix} -0.1 \\ -0.1 \\ 0.5 \\ 1.5 \end{bmatrix} \quad (7)$$

B. Four Variables Setup

In this case, the design vector of the optimisation contains the length of the first and second elements of the RUS open loop, l_1 and l_2, and the values $ratio_d$ and $ratio_u$, from the eq. (1) and (2), that define the angles of positioning of the rotational and spherical joints on the fixed and mobile platform. Therefore, de design vector has the following form:

$$x_{4\text{vars}} = (l_1, l_2, ratio_u, ratio_d) \quad (8)$$

The constraints imposed in this case are presented in the eq. (9):

$$\begin{aligned} &l_1 > 0.01[m] \\ &l_2 > 0.01[m] \\ &l_1 < 0.5[m] \\ &l_2 < 1.5[m] \\ &ratio_u > 0.1 \\ &ratio_u < 10 \\ &ratio_d > 0.1 \\ &ratio_d < 10 \end{aligned} \quad (9)$$

In order to be implemented in Matlab, the constraints are written in matric form, as in the eq. (10), where $A_{4\text{vars}}$ and $b_{4\text{vars}}$ are the matrices from the eq. (11) and (12)

$$A_{4\text{vars}} \cdot x^t_{4\text{vars}} < b_{4\text{vars}} \quad (10)$$

$$A_{4\text{vars}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$b_{4\text{vars}} = \begin{bmatrix} -0.1 \\ -0.1 \\ 0.5 \\ 1.5 \\ -0.1 \\ 10 \\ -0.1 \\ 10 \end{bmatrix} \quad (12)$$

C. Six Variables Setup

Beside the dour variable from the previous setup, this configuration takes into consideration the radiuses of the mobile and fixed platforms, r and R. In this case, the design vector is given by the eq. (13).

$$x = (l_1, l_2, ratio_u, ratio_d, r, R) \tag{13}$$

The constraints for the 6 variable design vector is given by the eq. (14).

$$\begin{aligned} & l_1 > 0.01[m] \\ & l_2 > 0.01[m] \\ & l_1 < 0.5[m] \\ & l_2 < 1.5[m] \\ & ratio_u > 0.1 \\ & ratio_u < 10 \\ & ratio_d > 0.1 \\ & ratio_d < 10 \\ & r > 0.01 \\ & R > 0.01 \\ & r < 0.5[m] \\ & R < 1.5[m] \end{aligned} \tag{14}$$

In order to be implemented in the Matlab environment the constraints are written in matric form, as in the eq. (15), where $A_{6\text{vars}}$ and $b_{6\text{vars}}$ are the matrices from the eq. (16) and (17).

$$A_{6\text{vars}} \cdot x^t_{6\text{vars}} < b_{6\text{vars}} \tag{15}$$

$$A_{6\text{vars}} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$b_{6\text{vars}} = \begin{bmatrix} -0.01 \\ -0.01 \\ 0.5 \\ 1.5 \\ -0.1 \\ 10 \\ -0.1 \\ 10 \\ -0.01 \\ -0.01 \\ 0.5 \\ 1.5 \end{bmatrix} \tag{17}$$

D. Objective Function

The objective function evaluates the correlation between an user imposed desired workspace and the actual workspace of the robot. In this case, the workspace of the robot has to include an imposed cubical space, as seen in the figure below.

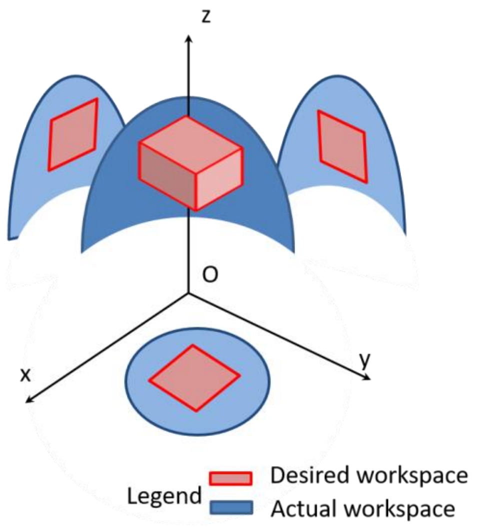

Fig. 5. Imposed Workspace and actual workspace of the robot.

The fitness function evaluates the ratio between the actual workspace of the robot and the desired workspace. The value of the fitness function is given by the eq.

$$fitness = \frac{WS_{desired} - WS_{imposed}}{WS_{desired}} \tag{18}$$

where $WS_{desired}$ and $WS_{imposed}$ are the volumes of the desired and imposed workspaces.

In ideal case, the two workspaces from the Fig. 4 are identical and the fitness value is null.

In the case presented in this paper, the length (L), width (W) and height (H) of the desired workspace are:

$$[L, W, H] = [0.3, 0.3, 0.1][m] \tag{19}$$

V. Results and Conclusions

The optimisation presented in the previous section has been implemented in Matlab using the Particle Swarm Optimisation, available at [11]. For all the cases (2,4 and 6 variables) the swarm size of the Particle Swarm Optimisation has been set from 10 up to 250 members (with a step of 10). Therefore, for each case, the algorithm has run 25 times.

This way, the Particle Swarm Optimisation algorithm has been evaluated regarding the effectiveness (i.e. the ability to reach an optimal solution), computational time and influence of the swarm size upon the results.

The Fig. 6 illustrates the value of the fitness function with regard to the number of the population size for all the three cases (2, 4, and 6 variables). From the figure, it is clear that a higher number of variables leads to a better result of the fitness function for all the swarm sizes. Nevertheless, a higher number of swarm population does not assure that the fitness value is lower (for example, the fitness value for a population of 150 for 6 variables is higher than a the fitness corresponding to a population of 10 members.

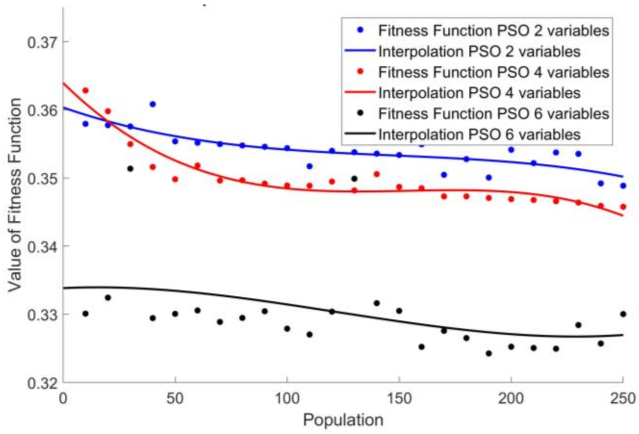

Fig. 6. Fitness function compared to population size for 2,4 and 6 variables design vector.

Moreover, there are particular cases (as 30 or 130 members in the population, 6 variables) in which the fitness value for a higher number of variables in the design vector has a larger fitness value than in a lower number (a higher fitness value implies an inferior optimisation result).

The Fig. 6 express graphically the demanded computational time with regard with the swarm size and number of variables of the design vector. As results from the image, a higher swarm size implies a higher computational times, for each size of the design vector. Also, for the same size of the swarm, the computational times increases significantly from 2 up to 6 variables design vector.

Comparing the Fig. 6 and 7 parallel, one can conclude that a higher number of design variables does not implies in all the cases a higher computational time (see the 2 and 4 design variables, Fig. 7) but it may have an important result upon the fitness value.

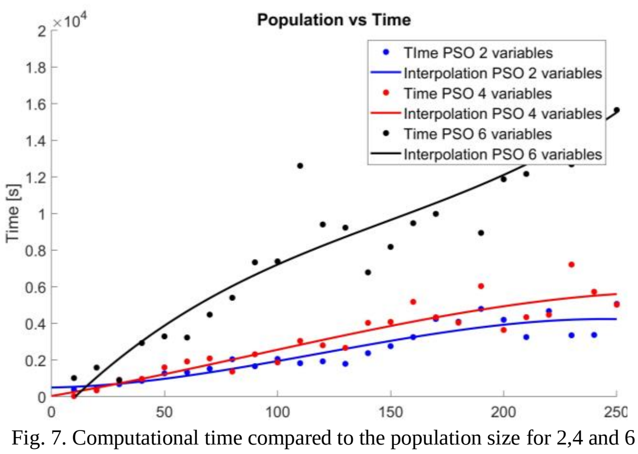

Fig. 7. Computational time compared to the population size for 2,4 and 6 variables design vector.

Therefore, the paper has realized a performance comparison in the optimisation of a parallel robot used for a Solid Waste Selection System, using the Particle Swarm Optimisation. As expected, a higher number of design variables, leads to a better fitness value, but has an important increasing effect upon the computational times. On the other hand, a higher number for the size of the swarm, does not implies a better fitness value.

As a future outlook, the research presented in this paper may be enhanced by comparing the Particle Swarm Optimisation with other optimisation algorithms. This way, one can identify the best optimisation algorithm for the design of robots.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

CB conducted all research and wrote the paper; CB analyzed the data, the experimental stage and the English grammar. All authors had approved the final version.

References

[1] Q. Xu and Y. Li. "Stiffness optimization of a 3-dof parallel kinematic machine using particle swarm optimization," *Robotics and Biomimetics*, 2006.

[2] D. Zhang and W. Bin. "Kinematic analysis and optimization for 4PUS-RPU mechanism," in *Proc. 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 330-335. IEEE, 2015

[3] Z. Zhang, H. Yu, and Z. Du, "Design and kinematic analysis of a parallel robot with remote center of motion for minimally invasive surgery," in *Proc. 2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 698-703, 2015.

[4] S. Kucuk, "Energy minimization for 3-RRR fully planar parallel manipulator using particle swarm optimization," *Mechanism and Machine Theory*, vol. 62, 2013, pp. 129- 149.

[5] R. Hassan, C. Babak, O. De Weck, and G. Venter, 'A comparison of particle swarm optimization and the genetic algorithm," in *Proc. 1st AIAA Multidisciplinary Design Optimization Specialist Conference*, pp. 18-21, 2005.

[6] D. C. Wilson, L. Rodic, A. Scheinberg, C. A. Velis, and G Alabaster, "Comparative analysis of solid waste management in 20 cities," *Waste Man-agement & Research*, vol. 30, no. 3, pp.237-254, 2012.

[7] A. K. Jha, S. K. Singh, G. P. Singh, and P. K. Gupta, "Sustainable municipal solid waste management in low income group of cities: A review," *Tropical Ecology*, vol. 52, no. 1, pp.123-131, 2011.

[8] O. Hancu, C. R. Rad, C. Lapusan, and C. Brisan, "Aspects concerning the optimal development of robotic systems architecture for waste sorting tasks, in *Proc. IOP Conference Series: Materials Science and Engineering*, vol. 444, no. 5, 2018.

[9] C. Boanta and C. Brisan, "Optimization of a robot used for a solid waste selection system," *International Journal of Modeling and Optimization*, vol. 9, no. 6, pp. 357-361, 2019.

[10] C. Boanta, M. Margaritescu, A. Verl, and C. Brisan, "Design concept of an automated solid waste selection system," in *Proc. International Symposium on Robotics*, Munich, Germany, 2018

[11] C. Boanta and A. Csiszar, "Optimal design of a parallel structure used as a haptic interface," *Mechanism and Machine Theory*, 2017.

[12] S. Chen, Particle swarm optimization. [Online]. Available: http://www.mathworks.com/matlabcentral/ fileexchange/25986 Distributed under BSD license

Catalin Boanta finished his master thesis in mechatronics at Technical University of Cluj Napoca, in 2016. Them he continues his studies, curently being in his fourth year of Ph.D. studies in the field of industrial engineering.

Cornel Brisan was researching in Duisburg, Germany as an Alexander von Humboldt fellow. From 2003 he is a full professor at the Technical University of Cluj Napoca, Department of Mechatronics and Machine Dynamics.