TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 68, Issue I, March, 2025

DEVELOPMENT AND SIMULATION OF SERIAL ROBOTS USING MATLAB

Ernest-Andrei GROSZ, Marian BORZAN

Abstract: *In recent years the term of modularity is more and more present in robotics and mechatronics area and therefore the necessity to have the possibility to easily create and generate modular robots in an easier and faster way, have been increased. The purpose of the paper is to present the possibility to generate and simulate a serial robots environment having up to three robots each with up to three DOF. The proposed method aims to have a MATLAB user-oriented tool in which everyone has the possibility to generate robots, starting from a robotic schematic requested, avoiding the multiple troubleshooting, caused by the need to use multiple environments or having advanced knowledge of programing or specific tools knowledge.*

Keywords: *Modular robotics, Modularity, Serial Robots, Robotics Software, Kinematics, Automation Software.*

1. INTRODUCTION

Robotics is a field that focuses on the development and study of robots, encompassing the modeling, design, and manufacturing of robotic systems. Industrial robots are extensively employed in the automotive and electronics industries, as well as in other fields.[1] Modular robotic systems are those systems that are composed of modules that can be disconnected and reconnected in different arrangements to form a new system enabling new functionalities.[2] The term ‘‘modular robotics’’ refers to a family of robotic systems made of interconnected smaller units called ‘‘modules’’, joined together by docking interfaces.[3] Modularity offers significant functional and economic advantages over more traditional fixed-structure robots. The ability to reconfigure the morphology by rearranging the connectivity of their parts enables modular robots to adapt to changes in the environment.[3] Modularity is a way of decomposing a big and complex system or product into small, simple, independent, and manageable modular units that can be easily composed, decomposed, and replaced, which can be frequently found in both nature and industrial systems.[4]

Software process simulation modeling is increasingly being used to address a variety of issues from the strategic management of software development to supporting process improvements.[5] MATLAB is a powerful environment for linear algebra and graphical presentation that is available on a very wide range of computer platforms.[6] New product development these days makes heavy use of IT tools such as virtual simulation tools. The main motivation for introducing virtual simulation tools in new product development is to speed up development and reduce costs. Virtual simulation tools can do much more than just speed up product development and reduce its cost.[7] The ability to predict the maximal performance of an industrial robot executing non-deterministic tasks can improve process productivity through time-based planning and scheduling strategies. These strategies require the configuration and the comparison of a large number of tasks in real time for making a decision; therefore, an efficient task execution time estimation method is required.[8]

Received: 30.01.25; **Similarities**: 05.03.25: **Reviewed**: 16.02./24.02.25: **Accepted**:20.03.25.

In the past, robots were predominantly used in manufacturing to perform labor-intensive tasks that were hazardous to humans. As robot capabilities rapidly increased, they were used for routine tasks that required speed, agility, and dexterity, such as spot welding, stacking, drilling, cutting, and so on. Today, autonomous cars increasingly incorporate sensors and machine learning that increase the type of tasks that can be undertaken. Therefore, robots can be used for a growing number of activities within firms, including transporting goods, assessing quality, testing products, and so on. One of the main reasons why firms adopt robots is their performance in terms of productivity and therefore production. Usually, robots can work continuously and perform functions 24 hours a day.[9]

In this paper, the main goal is to develop a platform, in which the user is able to generate up to three serials robots, having up to a maximum number of three degrees of mobility and possibilities to configure modules according to user requirements, in a proposed environment.

2. INFORMATIONS

In recent years, an increasingly accentuated development of the modularity of robots has begun by generating them with the help of command-and-control software. There is a wide range of software that follows the generation and calculation of robots and the possibility of manipulating their individual components.

Software that has the possibility to perform such simulations is MATLAB Simscape using the Simscape Multibody Library in order to simulate the robots and also the software SolidWorks to generate robot bodies.

2.1. Scope and determination

Some of the current limitations regarding the modeling and generation of a new project were observed and identified. The need to use multiple software packages, together with additional libraries, and the need to understand the particularities of each one can led to inefficient use of resources and time.

The determination to realize the program comes from the need to realize the projects in as compact and well-structured intervals as possible, also from the need to be able to realize them in an easier way, not requiring a large range of software programs, or additional programs or specific library to be able to realize and complete a project. At the same time, from the need to have a program as friendly as possible in which it is not necessary to invest time in its way of working, or in the way of connectivity with other software, the focus being oriented towards the realization of projects, there are no administrative blocking points, caused by the tools used and also from the need to bring all the necessary information in one place, having the possibility of realizing a large project with a minimum and focused set of knowledge.

The main goal of the project is to create a platform that will give the user the possibility to generate any kind of configuration for a serial robot with up to three degrees of mobility, having only fifth-class kinematic couples (rotation or translation). The need to create such a platform is to be able in to configure, generate, analyze and reconfigure as many robots as possible in the shortest possible time, making it more efficient, not only from the point of view of time, but also from the point of view of the resources necessary for the calculation and representation method to be able to fit the robots in the different projects.

With the advanced progress of technology, time becomes more and more compressed in daily work, therefore, in nowadays work projects, we want everything to have as much acceleration as possible in activity and evolution. For this reason, this project wants to create a friendly platform, easy to use and requires a minimum investment in understanding the way of working for any new member of a project, regardless of its degree of novelty or diversity. Offering the highest level of capability and performance with minimal knowledge of the platform program or implementation and usage, it provides a user-oriented usability perspective, providing one of the easiest ways to approach building and robot configuration. At the same time, the platform aims to create a graphic representation of the workspace, the possibility of positioning the robots inside it, on a real scale, in accordance with the design and implementation requirements of both kinematics and folding dimensions on a wide range of projects.

Considering the computing power and the multitude of knowledge dedicated to engineering that the MATLAB software component has, it will be used as the source program in the development of the software. The application will be developed on several layers, each of them having the possibility to be individually managed, giving the application modularity right from the start. The application will include manual code in its development. Also, features such as MATLAB Guide will be used to create graphic interfaces with which the end user will interact throughout the development of the projects.

2.2 Mode of operation of the proposed project

The first layer will be the start interface that will appear when the application is launched, as shown in Figure 1.

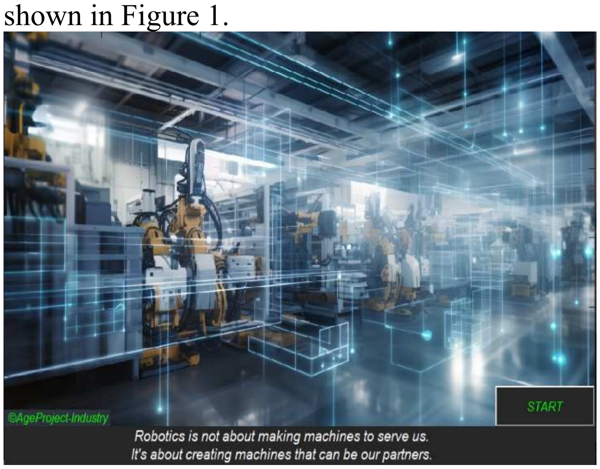

Fig 1. Pop-Up screen shown when the application is launched.

The platform will have several operating steps in which it will ask the user about the options he wants to enter, in order to generate the project. Figure 2 shows the operating method of the software, together with the steps followed by its user during the use of the platform and at the end of the generation of the serial robots.

Fig 2. The operating method of the developed software platform.

In an initial phase, the platform opens a pop-up in which the user is asked about the total number of robots needed for the project. Also here, the user is told that the number of robots the platform lets them choose varies from a minimum of one robot to a maximum of three robots. Otherwise, if the number is not valid or no number is entered, another pop-up window will be displayed, asking the user to enter a valid number. This pop-up window is shown in Figure3.

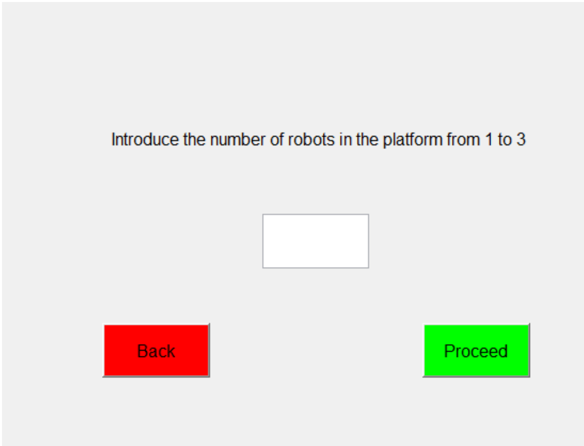

Fig 3. Pop-Up for selecting a total number of robots for their introduction into the platform.

In this mode of use, by introducing hints and asking the user about his preferences, the platform is sure that the user will reach the desired final result. In this way, the entire project will be developed, with a user-oriented approach.

After the user has been asked about the number of robots he wants to use in his project and he enters a valid number in the range of use of the platform, a new pop-up window will appear, asking him about the degrees of freedom of each robot that he will have introduced inside the workspace. The number of degrees of freedom that the user can choose is specified as being between a minimum of one and a maximum of three degrees of freedom for each robot. Similar to the previous pop-up, if the number entered by the user is not a valid one, located in the range of use, the platform will present a new pop-up in which the number chosen by user, is not a valid one and will be directed to the previous step to choose a valid number. The pop-up is shown in Figure 4.

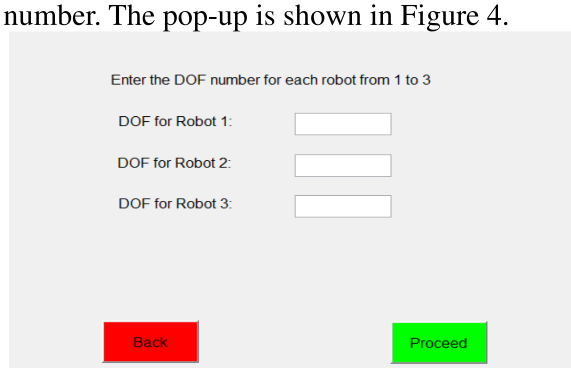

Fig 4. Pop-up for entering the number of degrees of freedom for each individual robot.

If the number of robots entered is not the maximum number allowed by the platform, it will only display the filled fields for the specific number of robots. After entering the number of degrees of freedom related to each robot, in the next step, translation or rotation kinematic coupling types will be selected for each individual robot along with the Ax, Ay and Az coordinates specific to the individual kinematic coupling, as shown in Figure 5.

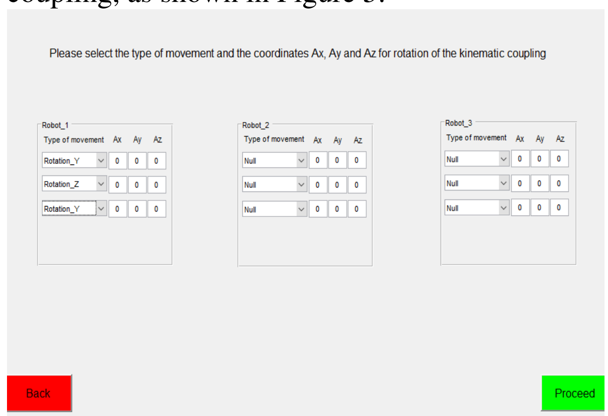

Fig 5. Pop-up for selecting the type of movement and entering the Ax, Ay and Az coordinates for the kinematic couples.

In order to use the platform, it is necessary to carry out an analysis of the desired movement, along with a schematization of the robot's movements, attaching a coordinate system [T0]...[Tn], [n ∊ ℕ| n ≥ 0, n ≤3] to each element. In order to realize this, the robotic schematic reference form Figure 6 is proposed.

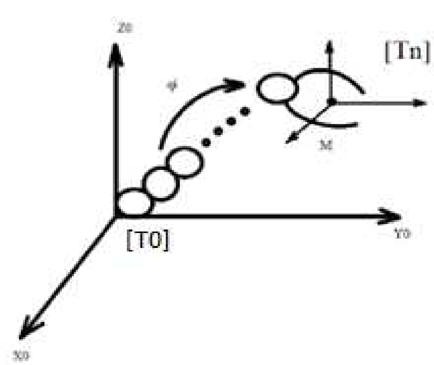

Fig 6. Attachment of coordinate systems, robot elements.

The relative motion between two consecutive elements can be described by a 4X4 homogeneous matrix called a transfer/transformation matrix that has the form:

$$A_i=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Where a_{14}, a_{24}, a_{34} represent the coordinates of the origin with the higher index in rank with the coordinate system with the lower index.

$$a_{11}=\cos(x_i x_i+1)\ ;\ a_{12}=\cos(x_i y_i+1)\ ;a_{13}=\cos(x_i z_i+1) \quad (2)$$

$$a_{21}=\cos(y_i x_i+1)\ ;\ a_{22}=\cos(y_i y_i+1)\ ;a_{23}=\cos(y_i z_i+1) \quad (3)$$

$$a_{31}=\cos(z_i x_i+1)\ ;\ a_{32}=\cos(z_i y_i+1)\ ;a_{33}=\cos(z_i z_i+1) \quad (4)$$

$$A_i=\begin{bmatrix} R_i & P_i \\ 0 & 1 \end{bmatrix} \quad (5)$$

R_i=[3X3]- Orientation sub-matrix

P_i=[3X1]- Position sub-matrix

The platform will ask the user what kind of movement the kinematic couples have in the origin of the current schematization, starting from the kinematic couple attached to the element in the system [T0] to the kinematic couple attached to the system [Tn]. The program will have predefined matrices for each type of kinematic couple as follows.

For the translational couple along the x axis, the homogeneous transfer matrix will be generated:

$$A_{i,i+1}^{T,x}=\begin{bmatrix} 1 & 0 & 0 & q_i \\ 0 & 1 & 0 & a_y \\ 0 & 0 & 1 & a_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

For the translational couple along the y axis, the homogeneous transfer matrix will be generated:

$$A_{i,i+1}^{T,y}=\begin{bmatrix} 1 & 0 & 0 & a_x \\ 0 & 1 & 0 & q_i \\ 0 & 0 & 1 & a_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

For the translational couple along the z axis, the homogeneous transfer matrix will be generated:

$$A_{i,i+1}^{T,z}=\begin{bmatrix} 1 & 0 & 0 & a_x \\ 0 & 1 & 0 & a_y \\ 0 & 0 & 1 & q_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

For the rotation couple around the x axis, the homogeneous transfer matrix will be generated:

$$A_{i,i+1}^{R,x}=\begin{bmatrix} 1 & 0 & 0 & a_x \\ 0 & cq_i & -sq_i & a_y \\ 0 & sq_i & cq_i & a_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

For the rotation couple around the y axis, the homogeneous transfer matrix will be generated:

$$A_{i,i+1}^{R,y}=\begin{bmatrix} cq_i & 0 & sq_i & a_x \\ 0 & 1 & 0 & a_y \\ -sq_i & 0 & cq_i & a_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

For the rotational torque around the z axis, the homogeneous transfer matrix will be generated:

$$A_{i,i+1}^{R,z}=\begin{bmatrix} cq_i & -sq_i & 0 & a_x \\ sq_i & cq_i & 0 & a_y \\ 0 & 0 & 1 & a_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

In the next step, geometric parameters Ax, Ay, Az will be requested for the positioning of the kinematic couplings as follows:

- The geometric parameters are given for the first kinematic couple that realizes the movement between the systems [T0] and [T1].

- For the following kinematic couplings, the geometric parameters are required in relation to the kinematic coupling attached to the preceding element.

- The geometric parameters of the final effector will be completed in relation to the coordinate system [Tn] attached to element n.

In the next step, the platform asks about the kinematic mode desired to be to perform, be it direct or inverted kinematics as shown in Figure7.

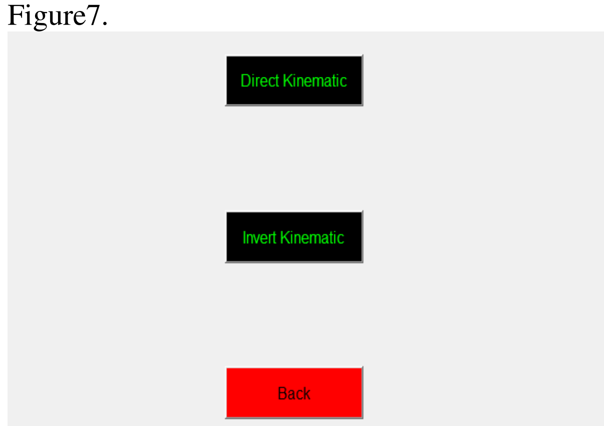

Fig 7. Pop-up for selection of the kinematic mode.

In the next step in direct kinematic mode, the initial positions of the couplers can be calculated using the “Calculate” button shown in Figure 8.

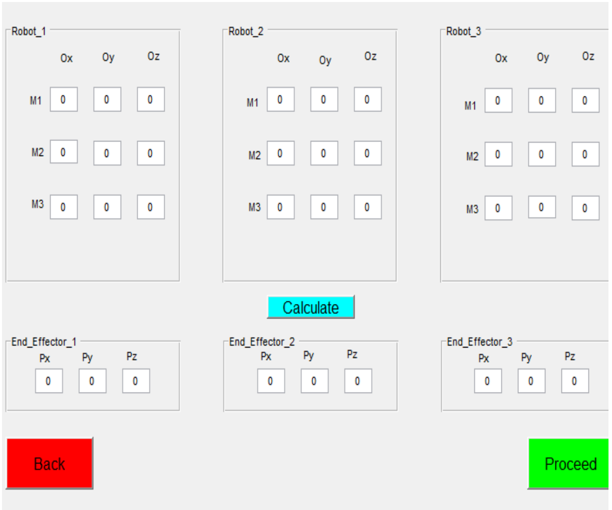

Fig 8. Pop-Up for the direct kinematic mode selected used for calculation of the kinematic couples.

The matrix Pn is generated, the matrix of the coordinates of point M in relation to the system [Tn], along with the end effector (EF) which represents the characteristic point of the robot attached to the end effector.

$$P_n = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (12)$$

To establish the parking position, the generalized coordinates Q for each kinematic couple will have value 0 by default, later having the possibility to change its value through the user interface.

For the simulation of the robot, homogeneous transfer matrices will be generated, these being calculated with the following formula:

$$A_{0,n} = \prod_{i=o}^{n-1} A_{i,i+1} \quad (13)$$

The position of the characteristic point attached to the final effector in relation to the system [T0], expressed with the help of transfer matrices, will be found using the relation:

$$P_{0,n} = \prod_{i=0}^{n-1} A_{i,i+1} \cdot P_n \quad (14)$$

In order to calculate the type of kinematic coupling and its type of movement, rotation on X, Y or Z or translation on one of the mentioned axes, a generation function was realized, a function that depends on the geometric parameters Ax, Ay and Az together with the coordinate generalized Q. The code for function is presented in Figure 9.

```
function A = Functie_A (Q,Lx,Ly,Lz,i);
% Q
% Lx = Ax
% Ly = Ay
% Lz = Az
% i case selection number



    if i == 1
        A=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]; % unit matrix
    end
    if i == 2
        A=[1 0 0 Lx; 0 cos(Q) -sin(Q) Ly; 0 sin(Q) cos(Q) Lz; 0 0 0 1]; % Rotation on X
    end
     if i == 3
        A=[cos(Q) 0 sin(Q) Lx; 0 1 0 Ly; -sin(Q) 0 cos(Q) Lz; 0 0 0 1]; % Rotation on Y
     end
     if i== 4
        A=[cos(Q) -sin(Q) 0 Lx; sin(Q) cos(Q) 0 Ly; 0 0 1 Lz; 0 0 0 1]; % Rotation on Z
     end
     if i == 5
        A=[1 0 0 (Q+Lx); 0 1 0 Ly; 0 0 1 Lz; 0 0 0 1]; % Translation on X
     end
     if i == 6
        A=[1 0 0 Lx; 0 1 0 (Q+Ly); 0 0 1 Lz; 0 0 0 1]; % Translation on Y
     end
     if i== 7
        A=[1 0 0 Lx; 0 1 0 Ly; 0 0 1 (Q+Lz); 0 0 0 1]; % Translation on Z
     end
end
```

Fig 9. The used script for calculation function for kinematic couples.

In the generation step, the platform will open a window in which the generalized coordinates will be entered and with the help of which the function will be calculated, and the robot will be generated and displayed in the graphic interface shown in Figure 10.

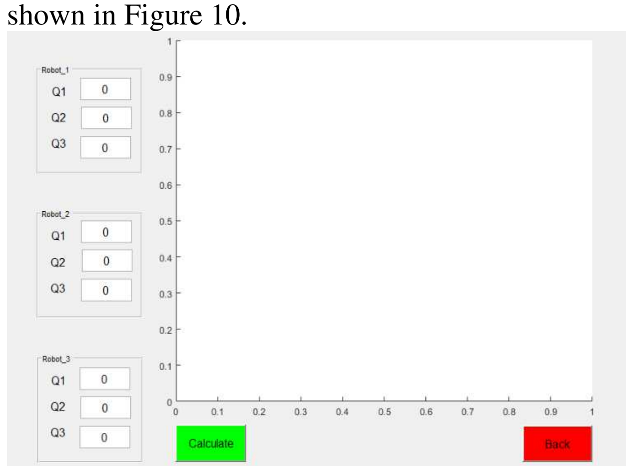

Fig 10. Pop-up for robotic generation interface.

3. RESULTS

In order to validate platform calculations, the example of the $R_z R_y R_y$ robot as shown in Figure 11 is proposed.

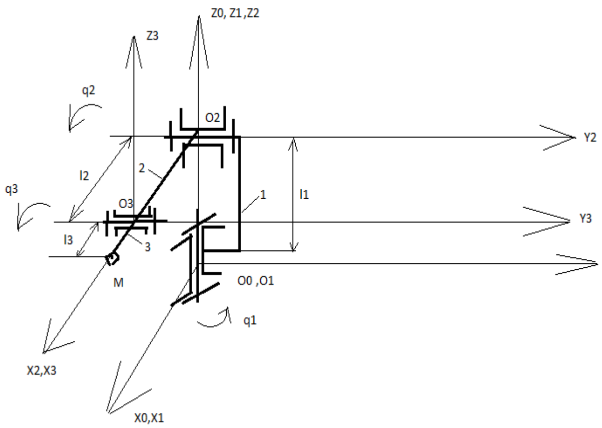

Fig 11. Schematic of the RRR robot that will be analyzed.

The defining matrices of the robot will be written as follows:

$$A_1=\begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

$$A_2=\begin{bmatrix} c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

$$A_3=\begin{bmatrix} c_3 & 0 & s_3 & l_2 \\ 0 & 1 & 0 & 0 \\ -s_3 & 0 & c_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$$P_M^3=\begin{bmatrix} l_3 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (18)$$

Where:

$$l1=20cm \quad (19)$$

$$l2=30cm \quad (20)$$

$$l3=40cm \quad (21)$$

To use calculation software for validation, the Mathcad program was used, the initial values being as follows in Figure 12.

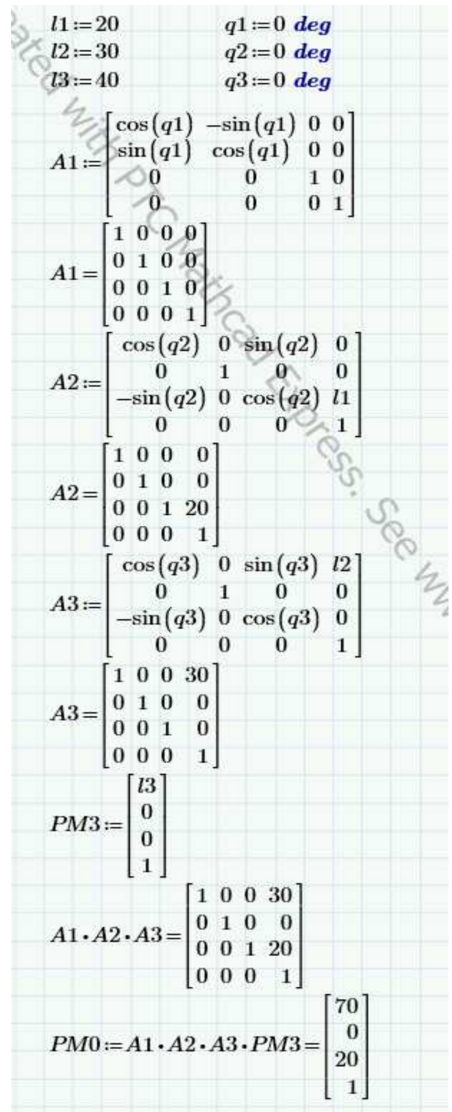

Fig 12. Mathematical validation for initial values using Mathcad Software.

In order to carry out the exercise, the following series of figures from Figure 13 to Figure 17 presented will show how to enter the data required by the platform in order to generate the RzRyRy robot up to the initial position compared with the mathematical results visible in Figure 12.

4. CONCLUSIONS

According to what was analyzed in the current stage of development in the field of robot generation, was observe a cumbersome, slow development with a multitude of possible errors introduced from the early stages, in which a multitude of tools, software programs together with additional libraries together with a high level of knowledge in handling them are necessary in order to generate robots for kinematic analysis and their visual representation.

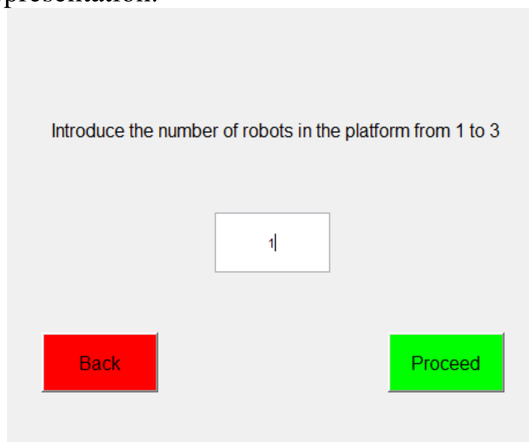

Fig 13. Pop-Up for selecting the total number of robots for their introduction into the platform introducing one RRR robot.

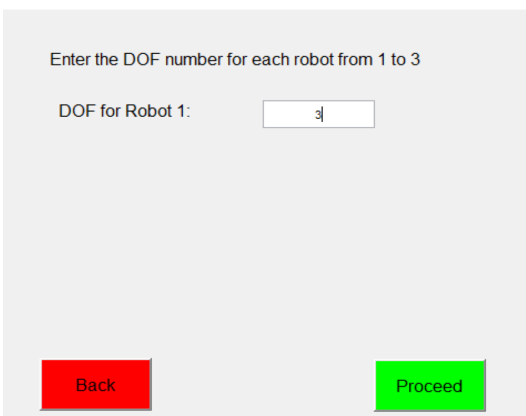

Fig 14. Pop-up for entering the number of degrees of freedom introducing three DOF for RRR robot.

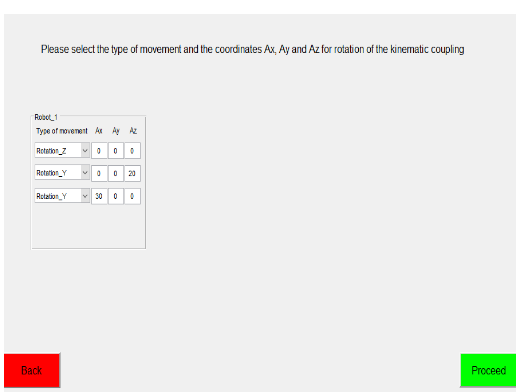

Fig 15. Selection of the type of movement and entering the Ax, Ay and Az coordinates for the kinematic couples for RRR robot.

Fig 16. Direct Kinematic Mode calculation for RRR robot.

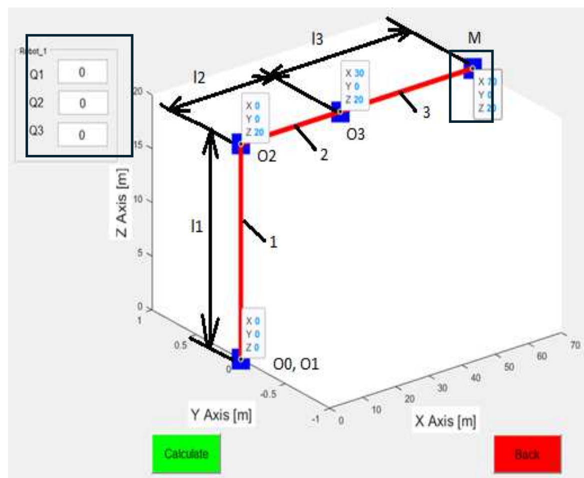

Fig 17. Running the platform until the RzRyRy robot is generated.

Using the MATLAB programming environment, the paper proposes the development of an application, oriented towards the end user for the generation of a workspace, the possibility of generating up to three robots whose morphology can have up to three degrees of mobility. Using this application, users will no longer need to use other additional software or other libraries to use the application, all the information being present inside the application.

The application has a friendly interface with the user, in which he is guided step by step in relation to the action he must do and the information he must enter in order to reach the right result, namely the generation of a preferential number of robots. After the configuration that complies with the basic requirements needed in the applied projects and in which the workspace of both the robotics and the work environment can be configured according to the needs of each.

In the next stage, after the generation of the robots is completed, users will have work programs for the functionality of the robots, graphics to use for the accelerations, speeds and intersections of the robots, together with trajectory calculations and the possibility of generating and simulating these inside a graphic interface with the user. By using this platform, a much faster result will be reached in terms of the time invested. Users have the advantage of being able to focus on the project, not on the administrative part of the connectivity of the tools, knowledge of the libraries and will be able to benefit from a platform to guide them in carrying out the projects in which they are involved. Also, having the possibility of implementing the entire workspace in a single environment and having the possibility of high-precision calibration, design errors will be easily identifiable, thus avoiding in the initial steps of the project possible costs, together with the unpleasant situations that followed of these. The application also brings the possibility of optimizing and improving the quality and performance of current development and research systems in virtual robotics.

REFERENCES

[1] Jia, Jiguang, and Xuan Sun. "*Structural Optimization Design of a Six-Degrees-of-Freedom Serial Robot with Integrated Topology and Dimensional Parameters*." Sensors 23.16 (2023): 7183.

[2] Yim, Mark, David G. Duff, and Kimon D. Roufas. "*PolyBot: a modular reconfigurable robot.*" Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065). Vol. 1. IEEE, 2000.

[3] Moubarak, Paul, and Pinhas Ben-Tzvi. "*Modular and reconfigurable mobile robotics."* Robotics and autonomous systems 60, no. 12 (2012): 1648-1663.

[4] Zhang, Chao, et al. "*Modular soft robotics: Modular units, connection mechanisms, and applications.*" Advanced Intelligent Systems 2.6 (2020): 1900166.

[5] Kellner, Marc I., Raymond J. Madachy, and David M. Raffo. "*Software process simulation modeling: why? what? how?* " Journal of Systems and Software 46.2-3 (1999): 91-105.

[6] Corke, Peter I. "*A computer tool for simulation and analysis: the Robotics Toolbox for MATLAB.*" Proc. National Conf. Australian Robot Association. 1995.

[7] Baba, Yasunori, and Kentaro Nobeoka. "*Towards knowledge-based product development: the 3-D CAD model of knowledge creation.*" Research policy 26, no. 6 (1998): 643-659.

[8] Righettini, Paolo, Roberto Strada, and Filippo Cortinovis. "*Neural network mapping of industrial robots' task times for real-time process optimization.*" Robotics 12, no. 5 (2023): 143.

[9] De Backer, K., et al. (2018), "*Industrial robotics and the global organisation of production*", OECD Science, Technology and Industry Working Papers, No. 2018/03, OECD Publishing, Paris.

[10] Tsai, Cheng-Han, Eduin E. Hernandez, Xiu-Wen You, Hsin-Yi Lin, and Jen-Yuan Chang. "*RoboTwin Metaverse Platform for Robotic Random Bin Picking*." Applied Sciences 13, no. 15 (2023): 8779.

[11] Chaturvedi, Devendra K. *Modeling and simulation of systems using MATLAB and Simulink*. CRC press, 2017.

[12] Becker, Markus C., Pasquale Salvatore, and Francesco Zirpoli. "*The impact of virtual simulation tools on problem-solving and new product development organization.*" Research policy 34, no. 9 (2005): 1305-1321.

[13] Quintana, Virgilio, Louis Rivest, Robert Pellerin, Frédérick Venne, and Fawzi Kheddouci. "*Will Model-based Definition replace engineering drawings throughout the product lifecycle? A global perspective from aerospace industry.*" Computers in industry 61, no. 5 (2010): 497-508.

[14] Wang, Yijie, Yushan Zhou, Lai Wei, and Ruiya Li. "*Design of a Four-Axis Robot Arm*

System Based on Machine Vision." Applied Sciences 13, no. 15 (2023): 8836.

[15] Grosz, E. A., & Borzan, M. (2025). *"Modular Robotics Configurator: A MATLAB Model-Based Development Approach".* Applied System Innovation, 8(1), 21.

[16] Grosz, E.A., Borzan, M., Trif, A. (2024). "*Development of a robotic cell through the matlab simscape environment."* Academic Journal of Manufacturing Engineering, 22(3).

[17] Grosz, E. A., Lapusan, C., & Brisan, C. (2020, December). Aspects regarding the development of a gripper with variable geometry. In *IOP Conference Series: Materials Science and Engineering* (Vol. 997, No. 1, p. 012076). IOP Publishing.

[18] Karupusamy, S.; Maruthachalam, S.; Veerasamy, B. *Kinematic Modeling and Performance Analysis of a 5-DoF Robot for Welding Applications. Machines* 2024, 12, 378.

[19] Carbonari, L.; Palpacelli, M.C.; Callegari, M. *Inverse kinematics of a class of 6R collaborative robots with non-spherical wrist.* Robotics 2023, 12, 36.

[20] Noskieviˇc, P.; Walica, D. *Design and realisation of the simulation model of the Stewart platform using the MATLAB-Simulink and the Simscape Multibody Library*. In Proceedings of the 2020 21th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 27–29 October 2020; IEEE: Piscataway, NJ, USA; pp. 1–5.

[21] Mohapatra, S.; Srivastava, R.; Khera, R. *Implementation of a two wheel self-balanced robot using MATLAB Simscape Multibody.* In Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 25–28 February 2019; IEEE: Piscataway, NJ, USA; pp. 1–3.

[22] Hazem, Z.B.; Ince, R.; Dilibal, S. Joint *Control Implementation of 4-DOF Robotic Arm Using Robot Operating System.* In Proceedings of the 2022 International Conference on Theoretical and Applied Computer Science and Engineering (ICTASCE), Ankara, Turkey, 29 September–1 October 2022; IEEE: Piscataway, NJ, USA; pp. 72–77.

DEZVOLTAREA SI SIMULAREA ROBOTILOR VIRTUAL UTILIZAND MATLAB

Abstract: *În ultimii ani termenul de modularitate este din ce în ce mai prezent în domeniul roboticii și mecatronicii și de aceea a crescut necesitatea de a avea posibilitatea de a crea și genera cu ușurință roboți modulari într-un mod mai ușor și mai rapid. Scopul lucrării este de a prezenta posibilitatea de a genera și simula un mediu de roboți seriali având până la trei roboți, fiecare cu până la trei DOF. Metoda propusă urmărește să aibă un tool MATLAB orientat către utilizator în care fiecare utilizator să aibă posibilitatea de a genera roboți, pornind de la o schemă robotică solicitată, evitând multiplele depanări, cauzate de necesitatea utilizării mai multor medii sau având cunoștințe avansate de programare sau cunoștințe de tool-uri specifice.*

Cuvinte cheie: *Robotică modulară, modularitate, roboți seriali, software robotic, cinematică, software de automatizare.*

Ernest-Andrei GROSZ, Technical University of Cluj-Napoca, Department of Manufacturing Engineering, Faculty of Machine Building, Ernest.Grosz@campus.utcluj.ro, Str. Bucegi 11A, Cluj-Napoca, 0756099558

Marian BORZAN, Prof. Dr. Ing, Technical University of Cluj-Napoca, Department of Manufacturing Engineering, Faculty of Machine Building, Marian.Borzan@tcm.utcluj.ro, Str. Prof. Emil Petrovici, nr. 11, 0264 401647