

AUTOMATION OF PARTIAL TASKS IN THE DESIGN OF ROBOTIC ARMS

ZDENEK ZEMAN, MILAN MIHOLA, JIRI SUDER, ADAM BOLES LAVSKY

VS B - Technical University of Ostrava, Department of Robotics, Ostrava, Czech Republic

DOI: 10.17973/MMSJ.2022_03_2021130

e-mail: zdenek.zeman@vsb.cz

The design of robotic arms is a demanding process, especially in terms of the demands placed on developers' knowledge and experience and the time required for the design. Therefore, the development aimed to create procedures for possible automation of some of the individual tasks of the robotic arm design process, which are further applied in the development of a software tool for the automation of partial tasks in the design of robotic arms RobotDesigner. The benefits and functions of RobotDesigner are demonstrated in the example of optimization of a simple manipulator with 5 degrees of freedom for a given manipulation task. The optimization results show that the use of this software tool significantly speeds up and streamlines the sub-tasks of design and optimization of arms. By using iterative processes and artificial intelligence, using similar software tools and connecting them to CAD systems, it will not only be possible in the future to fully automate the robot arm design process but also to generate and compare different structures and design solutions to find the most optimal for the task, which the robot will perform.

KEYWORDS

Automation, Equipment design, Robotics, Mechatronics

1 INTRODUCTION

When designing mechatronic devices, great emphasis is placed on minimizing designs costs with the shortest possible design duration and flexibility. Due to these requirements, the design processes of mechatronic devices take the path of partial or complete automation. For this purpose, various software tools are being created, the task of which is to speed up and simplify certain parts of the design of new devices. These software tools reduce the duration of the design and often, due to their partial or complete autonomy, the knowledge requirements placed on the designers.

Currently, there are a number of software tools designed for the design of standardized machine parts, such as bolts, bearings, gears, etc., either in the form of stand-alone applications (e.g. MitCalc [MITCalc 2021]) or as part of CAD systems (e.g. KISSsoft [KISSsoft 2021]). The calculations according to which parts are designed and inspected using these software tools are in most cases based on generally known standards and procedures relating to the given types of machine parts (ANSI, DIN, ISO, etc.). For the design of non-standard parts such as drive units, some manufacturers provide online software tools for their selection. Still, the applicability of these tools is very limited and is only related to the products of one manufacturer. Apart from the online software tools, the manufacturers only state the procedures for selecting the part in the form of instructions in the catalogue. The execution of the inspection calculation itself and the selection of the part is then left to the developer.

Therefore, the research in this area follows the path of creating custom software tools that combine the design and selection of standardized and non-standardized parts from various manufacturers for the possibility of the complete design of components of various devices. These software tools solve various parts of the design of the resulting device, which then connect to each other and form complete autonomous units, where the individual software tools automatically transfer inputs and outputs to each other, and the result is a finished designed device or part of the device. Reddy and Sridhar presented a tool for parametric modeling of bolts, nuts and bearings [Reddy 2015]. Reddy and Venkatachalapathi created a tool for flexible modeling of box assemblies for industrial batteries using a knowledge base [Reddy 2018]. In both cases, the tools use the SolidWorks Application Programming Interface (API).

API stands for Application Programming Interface and refers to an application programming interface in computer science. When designing 3D models, it is used in connection with the creation of parameterized models, but it can also be used to create motion and strength analyzes, collision detection in the system, etc. To design components using this environment, the user fills in several required input values via GUI of the created software tool, and the application then calculates, designs, and creates a model of the resulting designed part/node according to the specified requirements. Furthermore, the designer can work with the models as if he had modeled them himself from the beginning. The advantage is a very fast design and the ability to use applications with minimal knowledge of the CAD system.

By linking these design and modeling processes, it would be possible to further create software applications that can design complete equipment, optimize designed structures, or generate artificial new structures using artificial intelligence by specifying the appropriate boundary conditions of the task for which the device is designed.

2 METHODOLOGY

RobotDesigner is a software tool created for the automation of partial tasks in the design of robotic arms. In its current form, the software is able to automatically optimize existing structures for a given trajectory or, in collaboration with genetic algorithms [Pastor 2021] design kinematic structures based on a given trajectory and assist designers in designing new robotic arm structures. The design processes of the robotic arms automated by this application are divided into three parts, preliminary design of the cross-section of the robot arms, design of the drive units, and creation of a simplified 3D model. The following subchapters discuss the methodology according to which these individual parts of the design process are addressed in the application.

2.1 Preliminary design of the cross-section of the robot arms

The preliminary design of the cross-section of the robot arms is achieved using known analytical methods. Their application can determine the angle of rotation and deflection of an arm loaded with different combinations of forces, bending moments and torques. The complexity of these methods depends on the number and location of entanglements (supports), the combination and places of action of individual loads, and the cross-sectional shape of the proposed arm.

The principle of the design of the cross-section of the robot arms is taken from the RobotArmDesigner software developed for this task. This software was developed for later use in the mentioned RobotDesigner software. Algorithms for the design of the cross-section of the robot arms use the calculation according to the

Castiglian theorem, based on deformation energies [Hearn 1997, Kurrer 2008]. Equations (1) and (2) apply to calculating the angle of rotation φ and deflection w according to this theorem.

$$\varphi = \frac{1}{E \cdot I} \int_0^l M(x) \cdot \frac{\partial M(x)}{\partial M} \cdot dx \quad (1)$$

$$w_b = \frac{1}{E \cdot I} \int_0^l M(x) \cdot \frac{\partial M(x)}{\partial F} \cdot dx \quad (2)$$

where E is the modulus of elasticity in tension [MPa], I is the moment of inertia of the cross-section [mm⁴], $M(x)$ is the moment [N·mm], $\partial M(x)/\partial M$ is the partial derivative of the moment according to the moment acting at the calculated location, $\partial M(x)/\partial F$ is the partial derivative of the moment according to the force acting at the calculated point and l is the length over which the integration takes place.

In the design of the arms of industrial robots with an angular structure, the individual arms are considered in the preliminary design as embedded beams, loaded at their ends by forces and moments from the weight of the suspended members or the load caused by the weight of the arms. Fig. 1 shows the possible loading of the designed arm with force F , bending moment T_b and continuous load q .

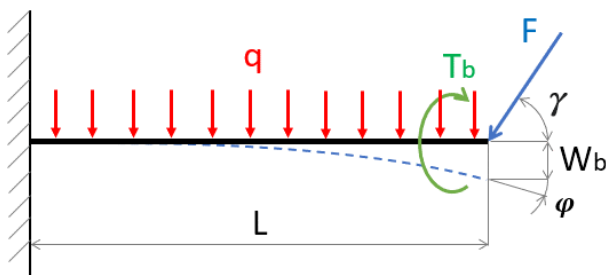


Figure 1. Load on embedded beam

The magnitude of the moment $M(x)$ at the point of embedding, based on which it is possible to determine the angle of rotation and deflection at the endpoint of the arm, can then be calculated using equation (3).

$$M(x) = F \cdot \sin(\gamma) \cdot L + T_b + q \cdot \frac{L^2}{2} \quad (3)$$

for $x \leq L$

The continuous load q represents the weight of the designed arm. In the case of the preliminary design of arm profiles, the dynamic effects are calculated based on its weight, and the magnitude of the continuous load q is therefore based on the density of the beam material ρ [kg · m³], its cross-sectional area S [m²] and gravity acceleration g [kg · s⁻²], which is calculated by the sum of all accelerations at the selected location of the beam and in the direction of the investigated deflection of the proposed arm, the relation (4) is used to calculate it.

$$q = \rho \cdot S \cdot \sum a_i \quad (4)$$

The adopted design algorithm (Fig. 3) selects the cross-section of the arm from the values of the maximum required allowed deflection. It also compares the value of the calculated bending stress with the allowable value for the selected arm material. Thus, the algorithm gradually reads the dimensions of cross-sections from the database from the smallest to the largest and checks whether they meet the specified parameters. The database from which suitable profiles are selected contains standardized round, square, rectangular, inter-circular and thin-walled square / rectangular profiles for steel and aluminium.

The maximum load values for a given designed part of the arm are obtained from the motion analysis on the skeletal/existing model of the selected kinematic structure, which is gradually supplemented by the designed drive units and load-bearing elements. The design of the arm cross-section is then performed for the combination of forces in the X, Y and Z axes, with the largest resulting force. The designed cross-section of the arm is further checked for the combinations of forces with the one largest force in the individual axes (various other combinations of forces with the largest force in the X, Y and Z axis).

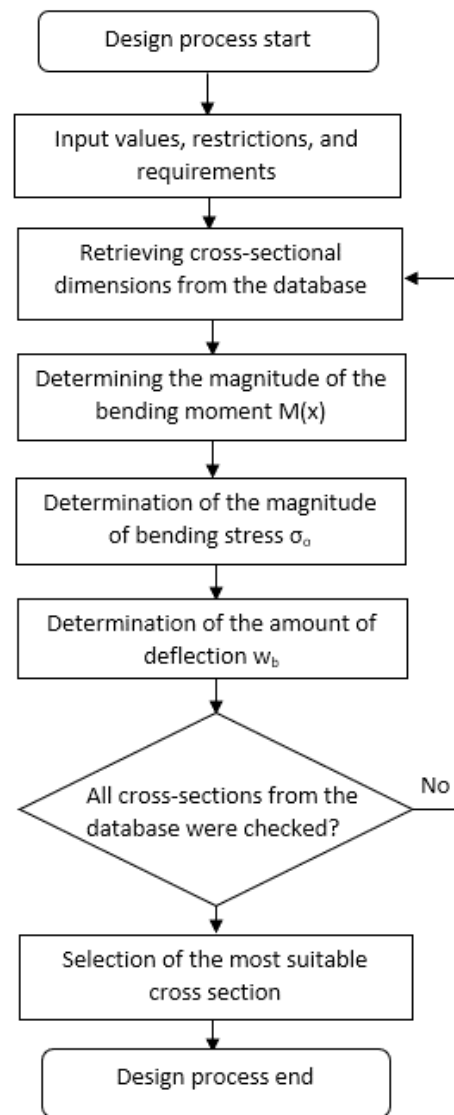


Figure 2. Algorithm for preliminary design of arm cross-section dimensions

In this way, the algorithm finds suitable dimensions for different types of cross-sections, for which it then selects the cross-section which has the lowest weight and which at the same time corresponds to the maximum permissible dimensions defined by the user. In this way, it is possible to work on a more suitable type of cross-section and its dimensions than would be the case if the design were completed at the moment when the first cross-section of the arm was found in the database, with which it is possible to achieve less deflection than required. The resulting deflection from the cross-section of the arm is then added to the deflection from the drive unit, which is calculated during its design.

The designed arm does not have to be loaded in only one plane. In such a case, it is necessary to distribute the force and moment effects into suitably designed, mutually perpendicular planes passing through the deflection line of the beam. The design of the beam cross-section dimensions would then proceed again according to the algorithm in Figure 2. Only the control of the maximum deflection would be performed in all of the planes, and then the calculation of the total deflection would be performed.

2.2 Design of drive units

When designing industrial robots and manipulators, the use of compact electric drive units is taken into account. These power units are calculated according to the instructions given by their manufacturers. The design of the drive unit is performed from the point of view of the course of the load by the input torque, the moment of inertia and the course of the input speed. The designed drive units are then subjected to inspection from the point of view of the force load on the outlet flange by radial and axial force and tilting moment, resp—their courses. Figure 3 shows the procedure for checking the drive unit Spinea DS series [Spinea 2021], according to its manufacturer's instructions. At this time, the DrivePicker software can design compact drive units from drive units Spinea DS and Harmonic Drive Canis Drive with the possibility of adding more in the future.

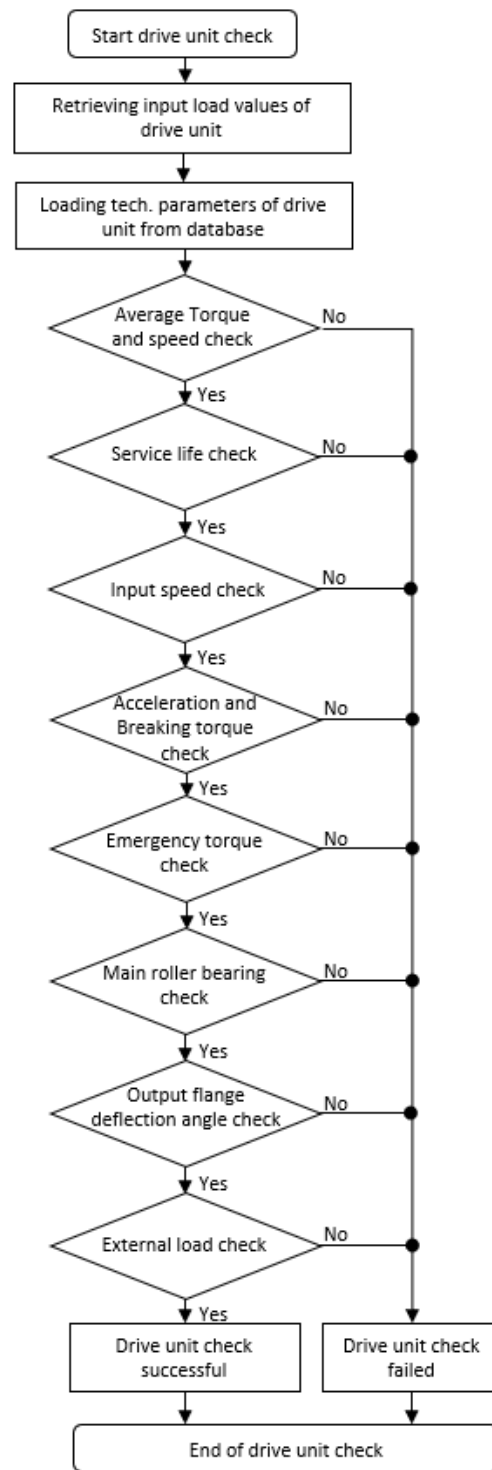


Figure 3. Algorithm for checking the drive unit of the Spinea DS series

The computing core of the developed DrivePicker software [Zeman 2021] is used to design the drive units. The adopted algorithms first process user-entered input data and calculate the maximum, minimum and average values of output torque, speed and radial and axial forces from them, these data are once again taken from motion analysis of the line model of the selected kinematic structure, which is gradually supplemented by the proposed elements.

Then the technical data of drive units are retrieved from the database, from which the design algorithm will select the most suitable one from each series of each manufacturer. The first algorithm uses the interval halving method to reduce the number of drive units from which the most suitable is selected

to a maximum of five units. The algorithm then finds the resulting unit among these remaining units by checking one by one until it finds the smallest one that matches the entered input values. This unit is then added to the list of suitable drive units, and the whole procedure is repeated for another series from another manufacturer.

The list of suitable drive units, always one from each series, then passes through an output filter, where their size and weight are compared. The resulting suitable drive unit is then the one with the lowest weight and the smallest possible size. If more units of the same size and weight are found, which differ only in, for example, the gear ratio, the values of max. torque and speed are further compared, and a unit is selected that has a higher max. torque at speeds of 20% higher than the required minimum.

2.3 3D model creation

All 3D models from which the robotic arm is assembled are stored in a knowledge database in two forms. The first form is 3D models of available power units, effectors, etc. In most cases, these are models taken from the manufacturers of the components. These models are divided into parts that can be combined into an assembly corresponding to a real component. The corresponding mass and moment of inertia are assigned to these parts, according to the actual values. The models are further supplemented by reference elements in the form of coordinate systems and axes, with the help of which it is possible to insert these adopted models into the assemblies of the proposed devices. Figure 4 shows the model of the CanisDrive drive unit supplemented by reference elements.

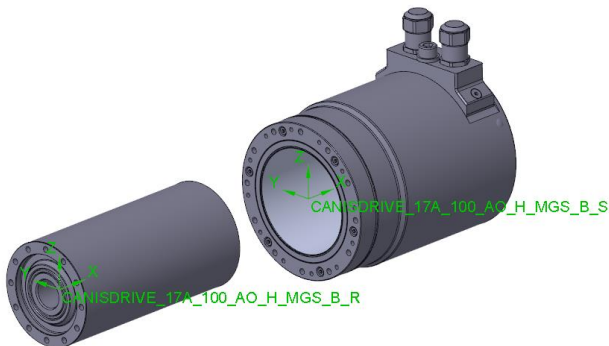


Figure 4 3D model with reference elements

In the second form, 3D models are prepared in the form of codes. These codes are prepared using the API tools available in SolidWorks CAD software, thanks to the possibility of recording the process of creating 3D models using the "Record / Pause Macro" function. This also means that one needs to have SolidWorks CAD software installed to run the software tool.

The "Record / Pause Macro" function records the individual actions leading to the creation of the desired 3D model. To be able to modify the code in the required way at a later stage, it is necessary to modify the individual dimensions of the model while creating it. It is suitable to proceed according to the tree structure of the model from the first element to the last. In this way, information on the dimensions of individual parts of the created model is arranged relatively clearly in one part of the code. Then the material or other properties are assigned to the model. The model created in this way is then saved. The last step is to end the function for recording the macro and then save it. To be able to use the code created in this way to create a parameterized 3D model, the values for the individual

dimensions are replaced by variables into which the required dimensions of the model are inserted before generation.

The same approach is used to create assemblies. Inserting elements into an assembly and creating links between them is done by suitable code sequences. The first sequence of code ensures that the model of the element that is to be inserted into the assembly is opened. The second sequence of code is used by the owner to insert this model into the assembly. The third sequence then closes the model window. If the developer inserts a model of another element, all he needs to do is change the model's name and the path to its location for the code designed in this way. The process of inserting individual elements into the assembly can therefore be automated with the help of several lines of code and a list of elements with links to their location and reference elements, through which the models are bound.

Currently, the Robot Designer software database contains models for Festo DHPS series effectors, models of Spinea DS and Harmonic Drive Canis Drives and models of basic load-bearing elements for creating 3D models of angular structures (basic models of brackets, flanges, arm cross-sections). The resulting 3D model is built automatically after entering the type of effector, drive units in the joints and the number and lengths of individual arm links. With the exception of power units, the design of which is connected with the creation of the model, these parameters are entered into the software manually according to the given kinematic structure.

3 ROBOTDESIGNER SOFTWARE TOOL

RobotDesigner is a software tool developed to design parts of the supporting subsystems of industrial robots and manipulators optimized for specific tasks. The primary purpose of the software is to reduce the design time of these robotic arms by automating individual design processes. Using this software tool, it is very easy to optimize structures through the iteration process and thus achieve better robot parameters to fulfil the given task. In its current form, the software is limited only to designing and optimizing angular robot structures with 3-6 degrees of freedom. Figure 5 shows a block diagram of the RobotDesigner software tool architecture.

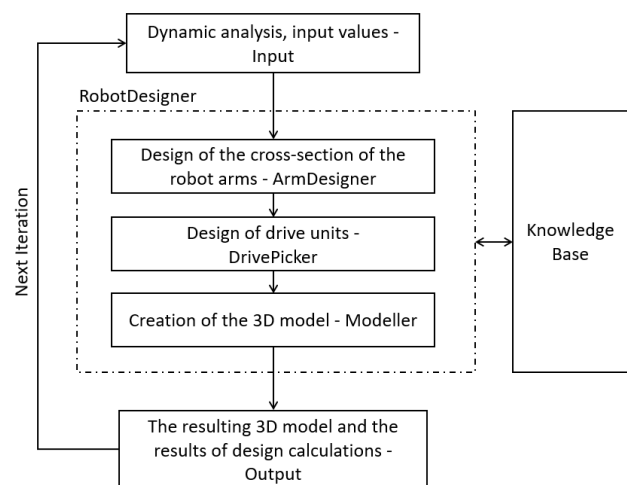


Figure 5 RobotDesigner software tool architecture

The user works with the application via its graphical user interface (GUI), which is divided into four basic parts (see Fig. 6). The first part is a panel with buttons for creating/saving/opening of a project. This panel is located at the top of the window. After pressing the button for creating a project, a dialogue box will appear in which the user selects what type of project he wants to create, either the design of the drive unit, the design of the cross-section of the arm (beam) or the complete design of the robot arm, ie. its supporting subsystem and manipulator. Then, based on the selected project type, a window is displayed in which the user enters the input values of the given design. The second part is the panel, located in the left part of the window, then used to display the basic parameters of the already designed arm/drive unit/beam. The third part is the panel located in the middle of the window is used to display the created 3D model of the designed part/robot, and the last fourth part is the panel on the right side of the screen is used to measure distances on the displayed 3D model, where the user selects in the panel with a 3D model of the surface/edge and using the capture measurement button located at the bottom of the fourth panel, displays the dimension of the selected distance.

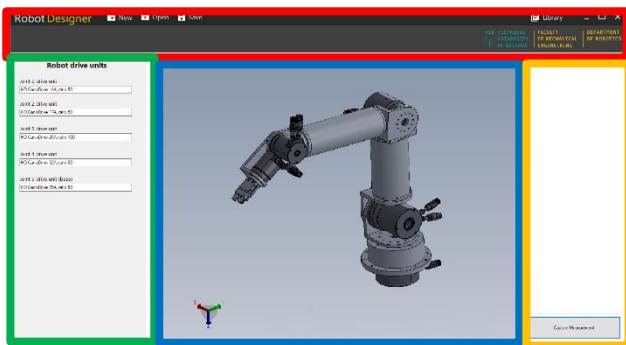


Figure 6 RobotDesigner GUI

The dialogue box for entering input values for the robot arm design project can be seen in Figure 7. For each arm, forces and moments act from previously designed parts of the robotic arm. The design starts from the end of the robotic arm, resp. From the object of manipulation and ends with the base of the proposed device. The input values for the design of the arm parts are the waveforms of the output torque, the output speed of rotation and the axial and radial forces acting in each joint of the arm for a given time. This data can be obtained by exporting from the analyzes of robotic arms that we want to optimize, performed in CAD software. The data is entered into the application via a properly formatted file. The input data is stored in a column format in a precisely defined order (time, output torque, output speed, radial force, axial force). Supported input file formats are .txt, .xml and .csv files. Other input values for the design of the robot arm are the moment of inertia of the load in each joint, the minimum required service life of the drive unit bearings and in the case of beam elements, also the length and maximum cross-sectional dimension of the arm. These values are entered manually by the user via the application UI.

The dialog box 'Design Robot' contains the following fields and controls:

- Project name:** Test_project
- Working Directory:** C:\Users\Zdeněk\Desktop\RobotDesigner_v01 (with a Select button)
- Robot base:**
 - Input data file:** C:\Users\Zdeněk\Desktop\RobotDesigner_v01\data5.txt (with an Add file button)
 - Load inertia:** 3.2 kgm2
 - Bearing life:** 10000 h
- Arm 3:**
 - Input data file:** C:\Users\Zdeněk\Desktop\RobotDesigner_v01\data4.txt (with an Add file button)
 - Load inertia:** 2.8 kgm2
 - Bearing life:** 10000 h
- Arm 2:**
 - Input data file:** C:\Users\Zdeněk\Desktop\RobotDesigner_v01\data3.txt (with an Add file button)
 - Load inertia:** 2.6 kgm2
 - Arm length:** 550 mm
 - Bearing life:** 10000 h
 - Max. arm size:** 100 mm
- Arm 1:**
 - Input data file:** C:\Users\Zdeněk\Desktop\RobotDesigner_v01\data2.txt (with an Add file button)
 - Load inertia:** 2.2 kgm2
 - Arm length:** 350 mm
 - Bearing life:** 10000 h
 - Max. arm size:** 100 mm
- Gripper arm:**
 - Input data file:** C:\Users\Zdeněk\Desktop\RobotDesigner_v01\data1.txt (with an Add file button)
 - Load inertia:** 1 kgm2
 - Bearing life:** 10000 h

Buttons: Next, Cancel

Figure 7 Dialog for entering input data of the Design Robot project

After entering all values and confirming the dialogue with the Next button, the robot arm design starts. Based on the principles mentioned in the previous chapters, the software first selects a drive unit in each joint. Furthermore, the software will design the optimal cross-sections of the arm beams and their size. Finally, according to the dimensions of the selected beams, the arms' lengths, and the size of the drive units, the software generates a complete 3D model of the robot arm. The result is this complete 3D model and data on the type of power units used in its joints (see Figure 8).

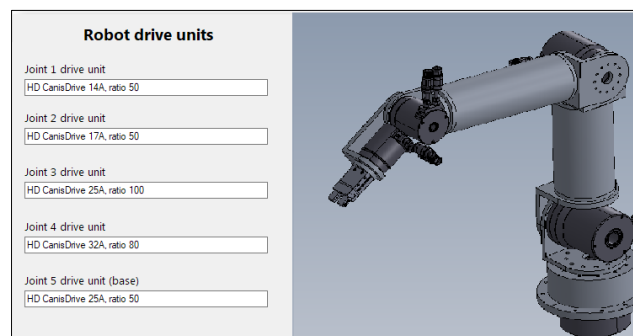


Figure 8 Results

4 EXPERIMENT

The use of the Robot Designer software tool is demonstrated to optimize a robotic arm with 5 degrees of freedom. In its current form, the software cannot fully automatically design new structures but is able to automatically optimize existing structures. To this end, a simple model of the robotic arm was created in SolidWorks software (see Figure 9), which serves as a basis to be optimized and which was subjected to dynamic analysis to obtain input values of torque, speed, and forces acting in individual joints.

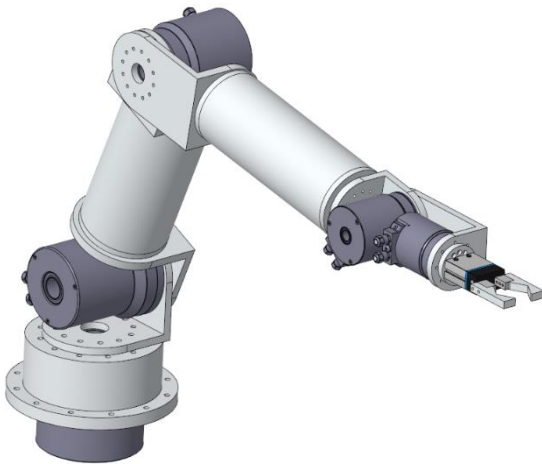


Figure 9. Kinematic model of the robot arm

A dynamic analysis was performed on the model on the trajectory, which can be seen in Figure 10. The maximum speed of the endpoint of the arm reached 0.6 m/s. The robotic arm will be optimized for this specified task (see Figure 10).

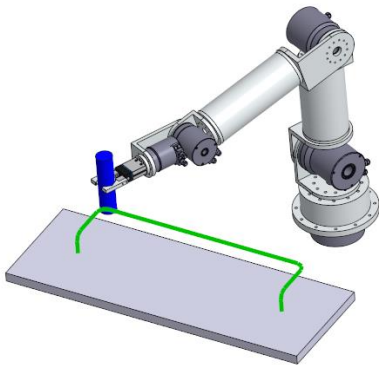


Figure 10. Selected manipulation task

The individual values obtained from the analysis of the values of torque, speed and radial and axial forces acting in each joint were exported from Cad to a text file, which was used as input to the Robot Designer software tool. The joints are numbered in order from the location closest to the end effector (Joint 1) to the base of the robotic arm (Joint 5). The courses of the mentioned values in the form of graphs can be seen in Figures 11-15.

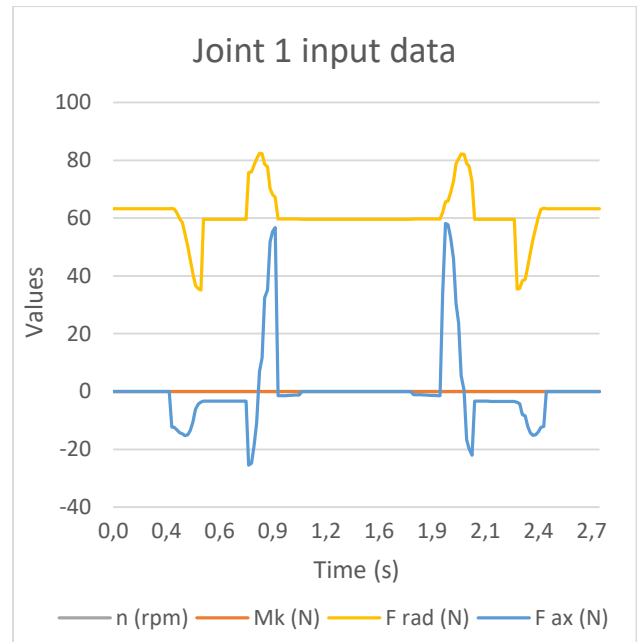


Figure 11. Joint 1 input data graph

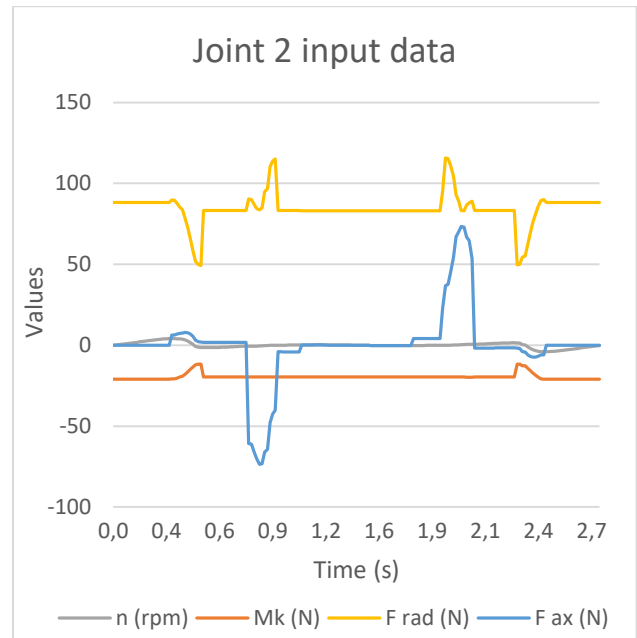


Figure 12. Joint 2 input data graph

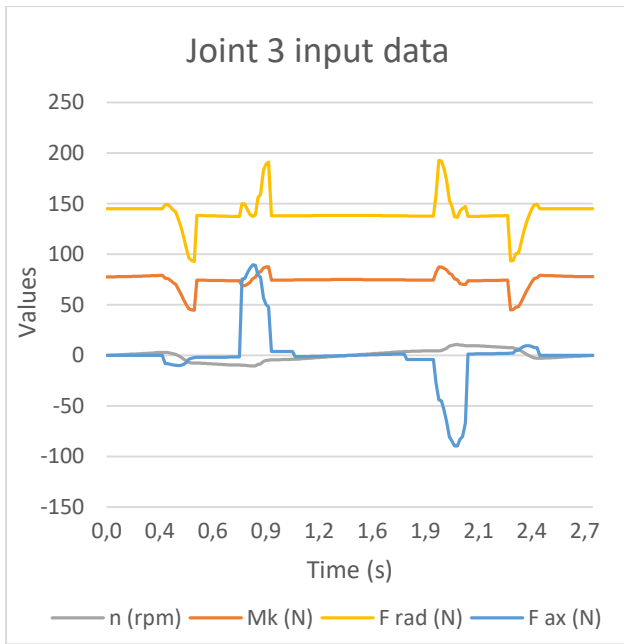


Figure 13. Joint3 input data graph

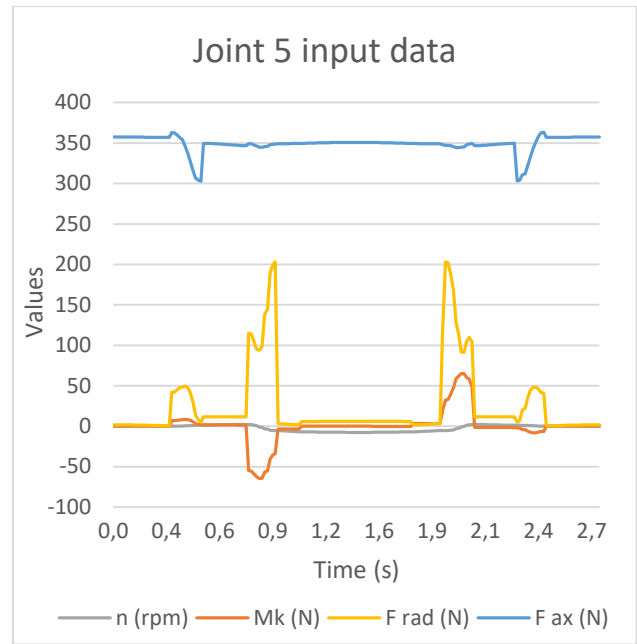


Figure 15. Joint 5 input data graph

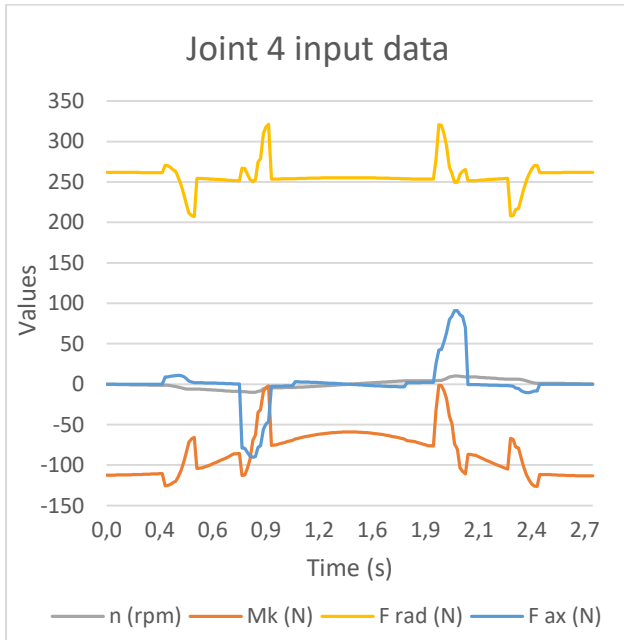


Figure 14. Joint 4 input data graph

After loading these values into the Robot designer application, it was necessary to specify additional parameters such as moments of inertia of loads for individual joints, required service life of drive units, lengths of arm beams and maximum cross-sectional size of arm beams. The moments of inertia of loads were obtained again from the basic 3D model in the CAD system SolidWorks. A value of 30,000 hours was selected as the minimum required bearing life for all drive units. The robotic arm includes two brackets between the joints 2-3 and 3-4, the length of the other arms depends on the size of the drive units. The length of the arm with the beam between the joints 2-3 (Arm 2) is 519 mm. The arm with the beam between the joints 3-4 (Arm 3) then has a length of 449 mm. The maximum cross-sectional size of these two beams was not limited in any way.

After finding and reaching all the necessary input values, it was possible to start the design of the optimized arm. The result of this design is a 3D model, which can be seen in Figure 16. The basic parameters of the designed, optimized robot and their comparison with the original model can then be seen in Table 1.

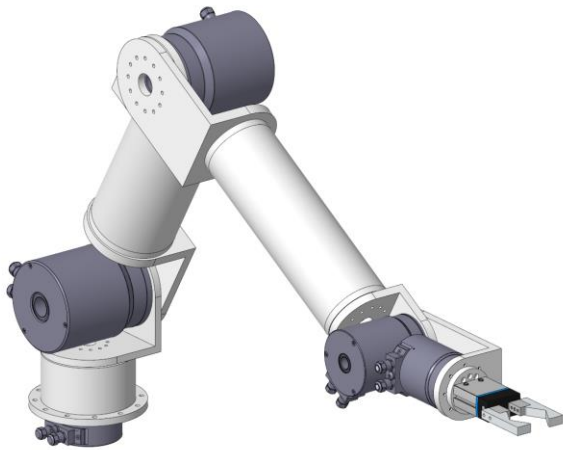


Figure 16. Kinematic model of an optimized robot arm

Basic parameters of designed manipulator		
Values	Original model	Optimized model
Joint 1 drive unit type	HD Canis Drive 14A (Ratio 100)	HD Canis Drive 14A (Ratio 50)
Joint 2 drive unit type	HD Canis Drive 17A (Ratio 100)	HD Canis Drive 17A (Ratio 50)
Joint 3 drive unit type	HD Canis Drive 25A (Ratio 100)	HD Canis Drive 25A (Ratio 100)
Joint 4 drive unit type	HD Canis Drive 32A (Ratio 100)	HD Canis Drive 25A (Ratio 100)
Joint 5 drive unit type	HD Canis Drive 40A (Ratio 100)	HD Canis Drive 17A (Ratio 100)
Arm 2 profile section	Ø 120 x 5 mm	Ø 100 x 3 mm
Arm 2 profile mass	1,43 kg	0,76 Kg
Arm 3 profile section	Ø 140 x 5 mm	Ø 120 x 3 mm
Arm 3 profile mass	1,46 Kg	0,66 kg
Manipulator total mass	39,23 Kg	24,32 Kg

Table 1. Optimization results

Comparing the selected parameters of the original and optimized model, we can see that during the optimization using software, there was a significant reduction in the total weight of the manipulator from 39.23 Kg to 24.32 Kg. This weight reduction has been achieved by replacing the drive units in the fourth and fifth joints with smaller units in terms of size and weight, which are of sufficient power to fulfil the chosen task of the manipulator. Furthermore, the cross-section of the robot arms was optimized for the profile of arm 2 from a diameter of 120 mm and a thickness of 5 mm to a diameter of 100 mm and a thickness of 3 mm. Similarly, the I profile of the arm 3 was replaced, which was optimized from a diameter of 140 mm and a wall thickness of 5 mm to a diameter of 120 mm and a wall thickness of 3 mm. According to the newly designed drive units and arm profiles, the dimensions of the brackets, flanges and base were also adjusted, which further reduced the total weight of the manipulator.

If we wanted to further achieve the most optimal state of the manipulator, we would have to perform further iterations of optimization, where we would replace the original model with an optimized one and perform identical motion analysis to obtain new input data for the software tool. We would repeat this until the original and optimized model were identical. Within the iteration, the software is also able to optimize the whole structure gradually in individual segments from the effector to the base, which reduces the total optimization time, but the final result is the same as when we optimize the whole structure at once. The software currently optimizes entire structures within one iteration to obtain data on changes in the proposed robot, which can then be used to teach neural networks to create estimated possible models according to the load in the joints.

The Robot Designer software tool is currently being expanded with the possibility of generating different types of manipulators. In the future, it is planned to design optimized robotic arms fully automatically using the iteration process where the involvement of artificial intelligence for simultaneous generation and comparison of different structures and kinematics of arms for use on task and to find the optimal one in terms of using different automatically generated parts.

5 CONCLUSIONS

The previous chapters described the development of a software tool for the automatic design of RobotDesigner arms. The methodology was described, according to which the individual parts of the arm design were automated. The work with the GUI software was described. Its function was verified on the task of mass optimization of the robot arm with 5 degrees of freedom to perform the given experimental manipulation task. As part of further development, it is planned to design optimized robotic arms fully automatically using an iteration process, and to consider the use of artificial intelligence to simultaneously generate and compare different arm structures and kinematics for use on a given task and to find the optimal one for using different automatically generated parts. For the possibility of complete automation, it is planned to perform dynamic analyzes in the future to obtain load values using the SolidWorks CAD API by linking the generated models to already moving skeletons, on which the analyzes will then be performed. Furthermore, the detection of collisions between individual parts of the robot and obstacles in the workspace will be solved. The API interface of the SolidWorks system also offers the possibility to perform

strength analysis on individual generated parts, where it is possible to change using macros as well as the dimensions of models and boundary conditions for strength analysis (loading, fastening, etc.). In this way, it would be possible to quickly inspect all parts of the robot in terms of strength and optimize them in size and shape.

ACKNOWLEDGMENTS

This article has been elaborated under support of the project Research Centre of Advanced Mechatronic Systems, reg. no. CZ.02.1.01/0.0/0.0/16_019/0000867 in the frame of the Operational Program Research, Development and Education. This article has been also supported by specific research project SP2022/67 and financed by the state budget of the Czech Republic.

REFERENCES

- [Hearn 1997] Hearn E.J. Mechanics of Materials 1. 3th ed. Oxford: Butterworth-Heinemann, 1997. ISBN 9780750632652
- [Kurrer 2008] Kurrer K.E. The History of the Theory of Structures: From Arch Analysis to Computational Mechanics. Berlin: Ernst & Sohn, 2008. ISBN 978-3433018385
- [KISSsoft 2021] KISSsoft AG. [online]. Copyright ©2021 KISSsoft AG [cit. 16.02.2021]. Available from: <https://www.kisssoft.com/en>
- [MITCalc 2021] MITCalc - Mechanical, Industrial and Technical Calculations. [online]. Available from <https://www.mitcalc.com/>
- [Reddy 2015] Esanakula, Jayakiran & Sridhar, Naga venkata & Rangadu, Vootukuri. (2015). Knowledge – Based

Parametric Modeling for Bolts, Nuts and Bearings using SolidWorks. International Journal of Applied Engineering Research. 10. 16111-16120.

- [Reddy 2018] E. Jayakiran Reddy, N. Venkatachalapathi, V. Pandu Rangadu. (2018). Development of an approach for Knowledge-Based System for CAD modelling. Materials Today: Proceedings. ISSN 2214-7853
- [Spinea 2021] SPINEA - Excellence in motion [online]. Available from: <https://www.spinea.com/>
- [Zeman 2021] ZEMAN, ZDENEK, MILAN MIHOLA a JIRI SUDER. DESIGN OF ALGORITHMS FOR AUTOMATIC SELECTION OF DRIVE UNITS FOR MECHATRONIC DEVICES. MM Science Journal [online]. 2021, 2021(2), 4362-4370 [cit. 2021-9-27]. ISSN 18031269. Available from: doi: 10.17973/MMSJ.2021_6_2021011
- [Pastor 2021] Pastor, R., Bobovsky, Z., Huczala, D., Grushko, S. Genetic Optimization of a Manipulator: Comparison between Straight, Rounded, and Curved Mechanism Links. Applied sciences. 2021. 11(6), 2471.

CONTACTS:

Ing. Zdenek Zeman
VSB - Technical University of Ostrava, Department of Robotics
17. listopadu 2172/15, Ostrava, 708 00, Czech Republic
+420 597 321 209, zdenek.zeman@vsb.cz, www.fs.vsb.cz/354