machines

Article

Integrating Trajectory Planning with Kinematic Analysis and Joint Torques Estimation for an Industrial Robot Used in Incremental Forming Operations

Sever-Gabriel Racz, Mihai Crenganiș, Radu-Eugen Breaz *, Alexandru Bârsan, Claudia-Emilia Gîrjob, Cristina-Maria Biriș and Melania Tera

Department of Industrial Machines and Equipment, Engineering Faculty, Lucian Blaga University of Sibiu, Victoriei 10, 550024 Sibiu, Romania; gabriel.racz@ulbsibiu.ro (S.-G.R.); mihai.crenganis@ulbsibiu.ro (M.C.); alexandru.barsan@ulbsibiu.ro (A.B.); claudia.girjob@ulbsibiu.ro (C.-E.G.); cristina.biris@ulbsibiu.ro (C.-M.B.); melania.tera@ulbsibiu.ro (M.T.)

* Correspondence: radu.breaz@ulbsibiu.ro

Abstract: Robot manufacturing involves continuous path control, which is now available for both robotic controllers and CAM software packages. However, CAM solutions are focused on generating the code for the robotic structure to follow the toolpath, without taking into consideration the dynamics and energy consumption. In this study, robot incremental forming was considered as the manufacturing process, and a simulation model, based upon Matlab-Simulink Simscape Multibody technology, was developed. The proposed model was fed with the trajectory information generated by the CAM program, and using an inverse kinematics function, it was able to generate the commands to drive the robotic structure on the technological toolpaths. The model was also used to study the dynamic behavior of the robot; external experimental data from a 3D force sensor were fed to the model to include the influence of the technological forces which appeared during the incremental forming process. Thus, using the proposed model in conjunction with the external CAM software, the influence of the workpiece position upon the joint torques could be estimated, opening the way for future optimization. The shortcomings of the model, mainly involving inaccurate information with regard to the physical properties of the robotic structure, were addressed by subtracting the dry-run joint torques from those obtained from the technological process.

Keywords: industrial robots; incremental forming; robotic machining; inverse kinematics; robot dynamics; workpiece placement

check for updates

Citation: Racz, S.-G.; Crenganiș, M.; Breaz, R.-E.; Bârsan, A.; Gîrjob, C.-E.; Biriș, C.-M.; Tera, M. Integrating Trajectory Planning with Kinematic Analysis and Joint Torques Estimation for an Industrial Robot Used in Incremental Forming Operations. *Machines* **2022**, *10*, 531. https://doi.org/10.3390/machines10070531

Academic Editor: Antonios Gasteratos

Received: 20 May 2022
Accepted: 21 June 2022
Published: 30 June 2022

1. Introduction

Industrial robots are mainly used for operations which require point-to-point motion control (e.g., pick and place, spot welding, simple assembly tasks, or labelling). However, the use of industrial robots for machining operations which require continuous path control (e.g., milling or deburring) are becoming more and more widespread [1–5].

Aspects of robot machining reported in the literature in various synthesis papers were reviewed [6–8]. Robot dynamics issues, mainly low stiffness of the robotic structure and machining vibrations [9,10], are considered to be major drawbacks of robot machining. Robot posture [11] and workpiece placement optimization [12] have been reported as viable approaches to improve the robot stiffness.

Machining quality, which includes both machining accuracy and surface quality, is also considered a major problem that hinders large scale implementation of robotic machining [7]. Reducing the cutting forces [13] and identifying the error sources [14] are both methods that have been employed to overcome these problems.

Monitoring and compensation approaches, for example the implementation of new control strategies such as force-control [15] and advanced position control [16], have also been studied to further improve the results of the robotic machining process.

Another issue which is also mentioned in the literature is related to the energy efficiency of robotic machining processes. The work reported in [17] indicates path planning strategies are an effective measure for the energy efficient use of robots.

Programming industrial robots, namely generating the code to control the motion of the end-effector along the toolpath, is identified in [6] as a major challenge in robotic machining. A relatively recent work, [18], states that generally, commercially available CAD/CAM software packages do not provide post-processors for industrial robots. However, this situation has changed dramatically in the last few years; nowadays many vendors do offer effective CAD/CAM solutions for robot programming.

Another manufacturing process which requires continuous path control is single point incremental forming (SPIF) [19,20]. It requires the coordination of the motion of the tool (punch) in a vertical direction (Z axis) with the planar motion of the sheet metal workpiece (on the XY plane). This process allows flexible manufacturing of sheet metal parts by plastic deformation without the need to use a die. Because SPIF in not yet widely accepted at an industrial level, specialized technological equipment is available only within research facilities, thus multi-axis CNC machine tools and industrial robots are generally used instead. Performing SPIF by means of industrial robots [21–24] is subject, in principle, to the same issues that are found in robotic milling. However, certain differences are manifested in regard to the processing forces (which are much more difficult to estimate, and usually higher, for SPIF), and in the expected manufacturing accuracy—the requirements for robotic milling are much higher than those for SPIF, where the spring-back phenomenon drastically influences the dimensional accuracy of the processed parts.

Virtual reality and digital twins, some of the main pillars of the Industry 4.0 approach, are also considered valuable tools for programming industrial robots, even when point-to-point continuous toolpaths are involved [25]. Testing the program and movement cycles of an industrial robot in the cyber world, before implementing it in the real world at the workshop level, can reduce the time and cost of the implementation, while keeping the process continually optimized. Digital twins have also been proposed to address the issues related to human–robot interaction for collaborative robots, a special class of industrial robots [26].

Some research work reported in the literature deals with enabling the development of industrial robots with learning capabilities by means of employing digital twins using knowledge graphs [27]. Other authors have focused on developing universal methodologies to create digital twins for industrial robots [28].

Estimating the forces acting upon the end-effector of an industrial robot is also an important topic of current robotic research. Some very interesting results are presented in [29] and [30], where a virtual force sensor, based upon the dynamic model of the robot, was used for this purpose.

Analyzing the results presented in the literature in regard to robotic operations requiring continuous path control, we identified the following research directions for the proposed approach:

- As stated in [8], commercially available CAM software packages are focused on path planning and code generation, but are not able to take the static and dynamic effects generated by the forces within the process into consideration. Thus, users are not aware at any time of the variation in the joint torques, or how these torques can influence the efficiency of the machining process;
- New control strategies applied to existing controllers can be effective, but usually their practical implementation requires supplementary equipment, which increases the overall costs. If the manufacturing accuracy is not critical, as in robotic SPIF, other methods for improving the robotic manufacturing processes which can be also implemented at shop floor level with minimum costs should be considered;

- Workpiece placement optimization has great potential as a solution for improving the efficiency of the machining process, with regard to resistant torques. New path planning software solutions allow the user to change the toolpaths very quickly. It is expected that changing the workpiece position in the robot workspace could influence the dynamics of the robotic structure in regard to the required joint torques, thus influencing the energy consumption;
- Robotic machining requires solutions which can be applied rapidly at workshop level, using both the existing robotic structures and the given machining layouts.

Thus, the approach targeted in this study was to build a CAE model and validate it experimentally, in order for it to be used as an auxiliary tool by the CAM software. Consequently, while the CAM will generate the machining code, the proposed CAE model will provide dynamic information, such as the joint torque values. Any change in the machining conditions (for example, the workpiece position) will be met by qualitative information about how this change will influence the joint torques.

It should be stated here that the proposed solution does not currently target either workpiece position optimization or dynamic robot toolpath optimization.

2. Materials and Methods

Our approach relied on an experimental layout designed for a robotic incremental forming process, based upon a KUKA KR 210 industrial robot (KUKA Deutschland Gmbh, Augsburg, Germany).

A Matlab-Simscape Multibody (The MathWorks Inc., Natick, MA, USA) first generation dynamic model of a KUKA KR 210 robot was developed to allow the user to study the behavior of the robot during the machining process when under the influence of the machining forces, to determine the joint torques, and to calculate the energy consumption of the system.

The construction of the Simscape model was based upon the 3D model of the robot, provided by KUKA Company on its website, in stp format. The 3D model was imported in Solidworks software (Dassault Systemes, Vélizy-Villacoublay, France), and specific constraints were introduced. Consequently, the 3D model was converted into a functional assembly, separating the main functional units of the robot—the robot base, the six axes of the robot, and the incremental forming working unit as the end-effector.

The next step involved the export of the 3D assembly from Solidworks to Simscape, by use of the Simscape Multibody Link Plug-In provided by Mathworks installed in an external CAD application (Solidworks). The result of the export was an xml file, which was subsequently imported into Simscape Multibody, generating a combined Simulink/Simscape simulation diagram.

The simulation diagram was completed using a Matlab function block, which modelled the inverse kinematics of the robotic structure.

A specific toolpath was selected for the processed part, and the program for following this toolpath was generated by SprutCAM software (SprutCAM Tech., Limassol, Cyprus). The code generated by the CAM solution (i.e., the positions and orientation of the end-effector) was provided as input for the inverse kinematic function block, which in turn outputted the joint angles, which were sent to the Simscape model.

Based upon the joint angle values, the Simscape model was able to calculate the torques within the robot joints. The processing forces, measured using an external 3D force sensor fitted on the end-effector, were fed into the model, and the differences between the torques encountered during the dry-run regime and the working regime were calculated.

In addition, using the internal capabilities of the robot controller, the currents through the motor joints were measured, both in the dry run regime and the working regime. The evolution of the currents was used to qualitatively validate the torques calculated by means of simulation.

Finally, the differences in the joint torques for three different positions of the workpiece in the robot's workspace were compared. The position of the workpiece within the robot workspace was changed to test how this influenced the values of the joint torques.

A flowchart of the method is presented in Figure 1.

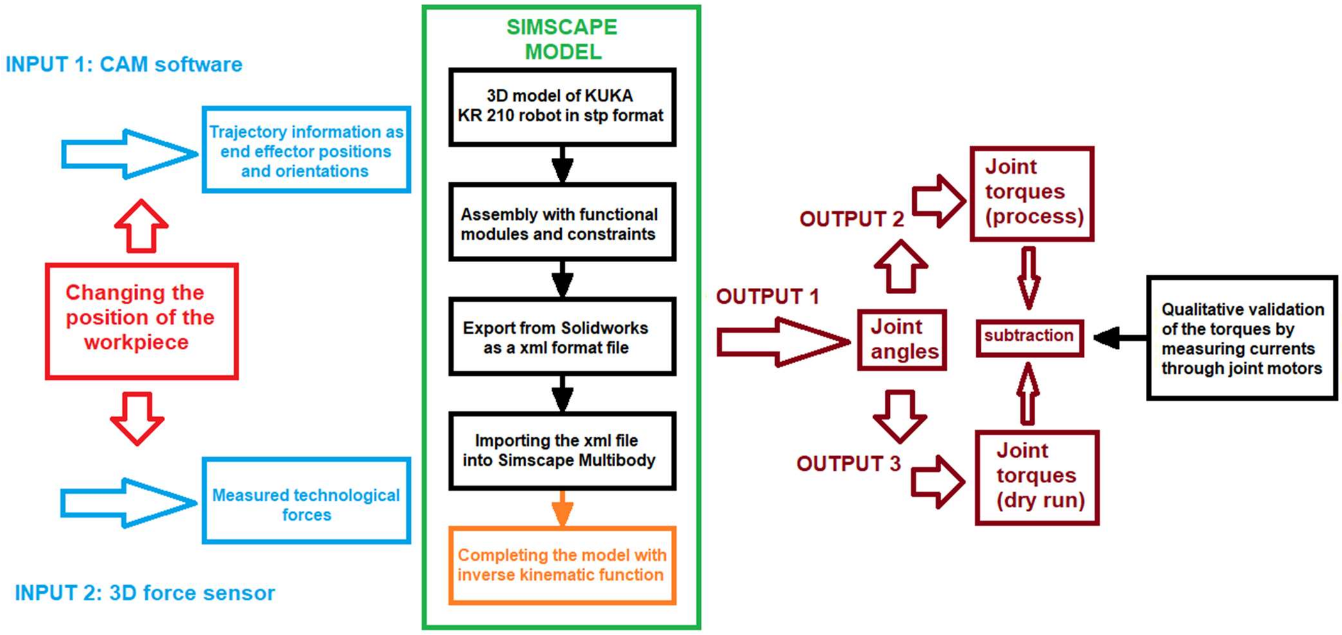

INPUT 1: CAM software
Trajectory information as end effector positions and orientations
Changing the position of the workpiece
Measured technological forces
INPUT 2: 3D force sensor
SIMSCAPE MODEL
3D model of KUKA KR 210 robot in stp format
Assembly with functional modules and constraints
Export from Solidworks as a xml format file
Importing the xml file into Simscape Multibody
Completing the model with inverse kinematic function
OUTPUT 1
Joint angles
OUTPUT 2
Joint torques (process)
OUTPUT 3
Joint torques (dry run)
subtraction
Qualitative validation of the torques by measuring currents through joint motors

Figure 1. Flowchart of the proposed approach.

2.1. KUKA KR 210 Kinematic

Within the field of robot kinematics, two fundamental types of problems can be distinguished: direct kinematic problems and inverse kinematic problems.

For the direct problem of positioning, if the geometric characteristics of the kinematic chain and the laws of variation of the generalized coordinates are considered known, one must determine of the laws of variation of the absolute position, and the absolute orientation of the characteristic point or the position of the final effector. Consequently, solving the forward kinetics problem for a robot means finding the position and orientation of the end-effector in relation to the base reference system, and knowing the joint positions and the geometric link parameter values.

For inverse kinematic problems, the geometric characteristics of the kinematic chain, as well as the laws of variation of the different absolute kinematic parameters, are considered known, and one must determine the relative kinematic parameters of the kinematic chain, or the relative movement between two consecutive elements of the robot. Developing the mathematical model and solving the inverse kinematics of a serial robot is a difficult and challenging task. To solve inverse kinematics using analytical methods is possible only for a simplified mechanical structure, and only if the robot has a particular position.

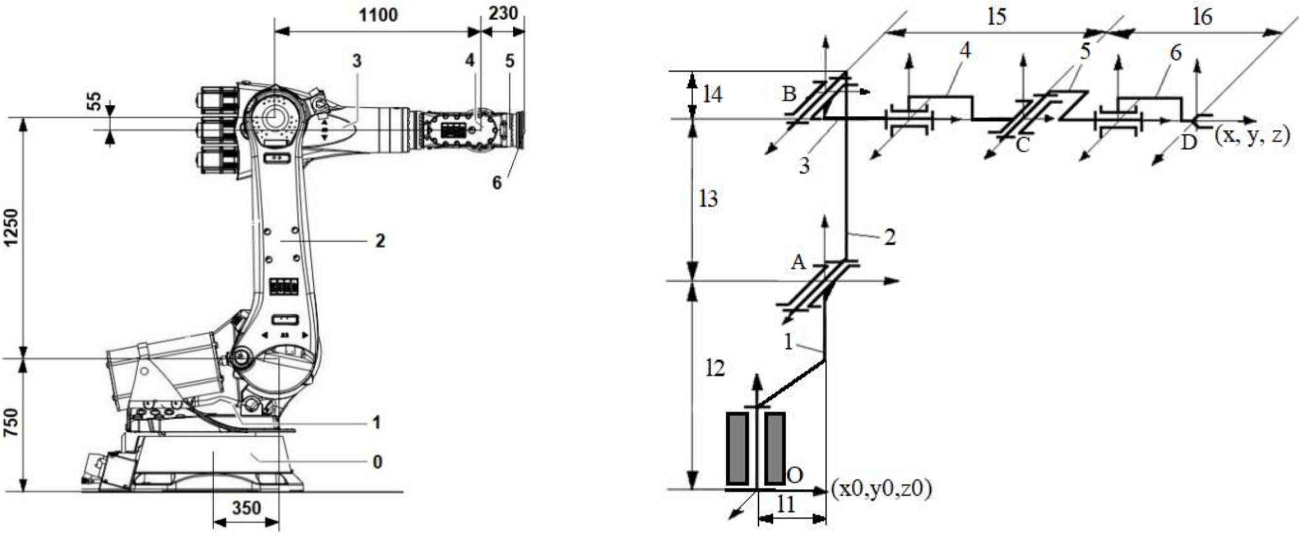
Obtaining the inverse kinematics equations is an essential part of understanding the robot's motion, and it is a preliminary step in modelling the robot's dynamics and control.

In this study, a method for obtaining the inverse kinematics equations was proposed, and the mathematical model was then tested using the Matlab-Simulink function block.

2.1.1. Direct Kinematics

Knowing the relative parameters for the kinematic chain, and the shape of the homogeneous transfer matrices between elements, respectively, for the coordinate systems attached to each kinematic element, one can determine the total transfer matrix between the end effector of the robot and the absolute coordinate system.

The kinematic structure for the KUKA KR 210-2 robot is presented in Figure 2.



(**a**) (**b**)

Figure 2. KUKA KR 210-2 robot simplified structure: (**a**) Overall dimensions; (**b**) Kinematic diagram.

The direct kinematics of the KUKA KR 210-2 robot was developed considering its Denavit–Hartenberg (DH) parameter set. Based on the DH parameters for each kinematic link and each robot joint, the homogenous transfer matrices for the entire robotic structure were developed. Using the Euler angles, the orientation of the end-effector, expressed in relation to the robot base, was obtained. Since the direct kinematics of standard six-degrees-of-freedom industrial robots are well-known, this data will not be presented in this paper.

2.1.2. Inverse Kinematics

The solution for the inverse kinematic problem for the KUKA KR 210-2 robot is presented in Appendix A. In order to validate the kinematic model of the robot, a Simulink diagram was built, and is presented in Figure 3, which integrates the Simscape model (noted as a virtual KUKA robot—limited by a green border) and the inverse kinematic block (limited by an orange border).

The Simulink diagram shown in Figure 4 reflects the following:

- The input to the diagram is the trajectory information (the end-effector positions and information, and the X, Y, Z, roll, pitch, and yaw angles generated by the CAM program). Based upon these imposed trajectory values, the inverse kinematic block outputs the joint angles (θ_1, θ_2, θ_3, θ_4, θ_5 and θ_6 values);
- The virtual KUKA robot uses the joint angles to drive the robotic structure along the programmed toolpath, and finally issues the measured position and orientations of the end-effector at the output “Body element 6”;
- As shown in Figure 4, the imposed trajectory data (generated by the CAM program) are identical with the verified trajectory data (calculated by the Simscape model);
- Thus, the Simscape model (virtual KUKA robot) was validated from a kinematic point of view;
- It should be noted that for the proposed applications, all trajectories were generated while keeping the tool axis fixed (i.e., the roll, pitch and yaw angles were zero), in order to ensure the readings of the 3D sensor were not affected.

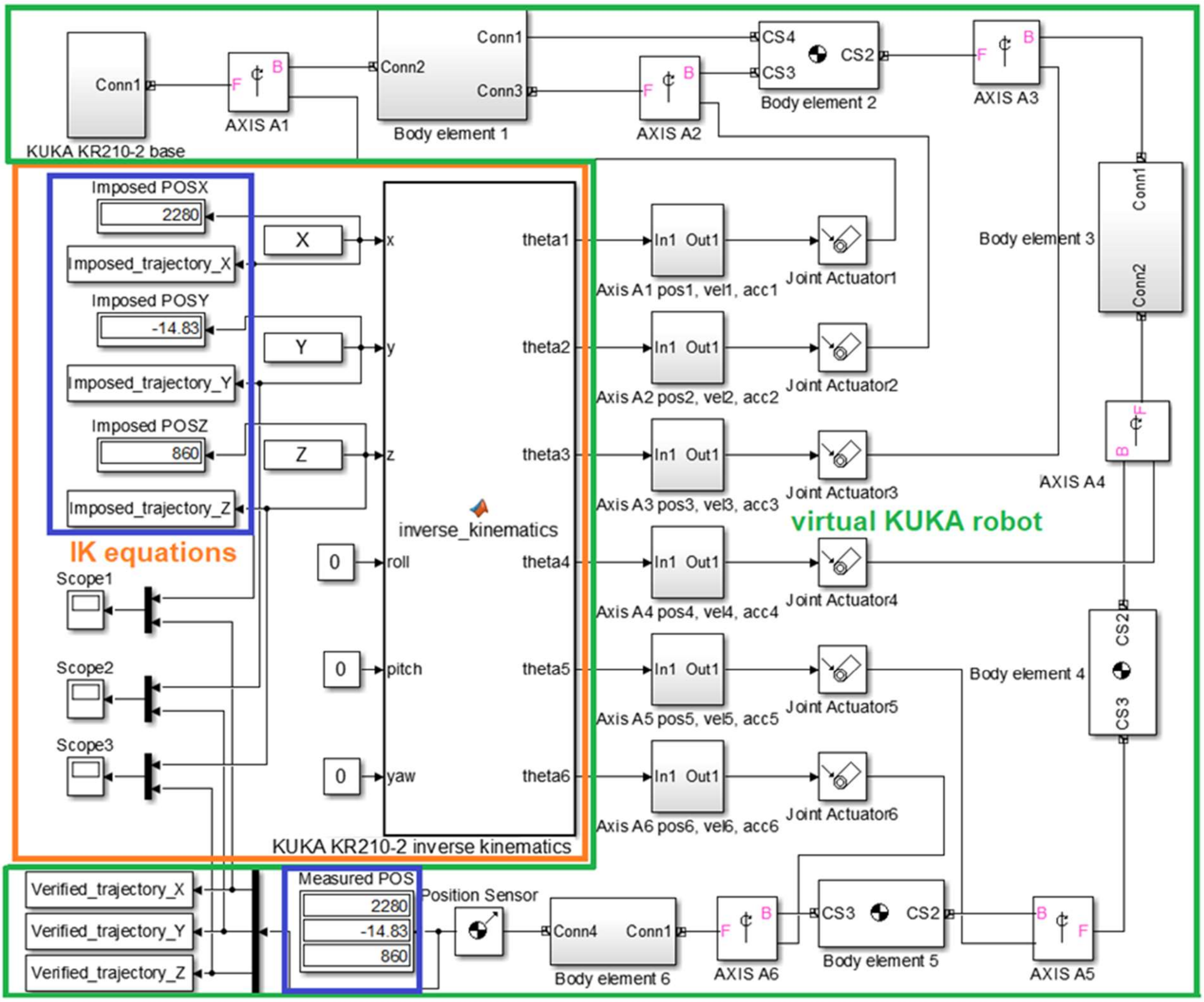


Figure 3. Simulink diagram used for validating the robot kinematic.

2.2. The Dynamic Model of the KUKA KR 210 Robot in Matlab-Simscape

The starting point for creating the dynamic model of the KUKA KR 210 robot in Matlab-Simulink Simscape Multibody was the 3D model of the robot, in .stp (.step) or igs (.iges) format. This model is made available by KUKA on its website (www.kuka.com/ accessed on 17 November 2021).

The next step involved opening and editing the 3D model of the robot in Solidworks. The robot model was defined as an assembly, isolating the component parts of the assembly (robot axes) and establishing the constraints necessary for the assembly to function properly.

The choice of the CAD program in which the editing was done was conditioned by the existence of a translator from CAD to Simscape.

The components of the robotic assembly (robot axes), together with the working unit for incremental forming (which integrates the 3D sensor), are shown in Figure 4.

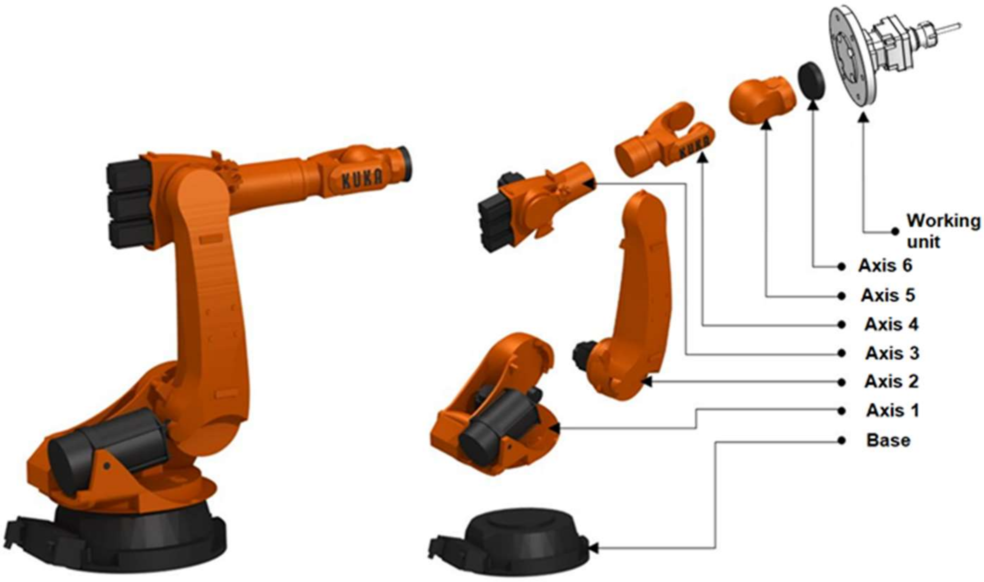


Figure 4. Main components of the robotic assembly (robot axes).

A screenshot with the robotic assembly in Solidworks before exporting to Simscape is shown in Figure 5.

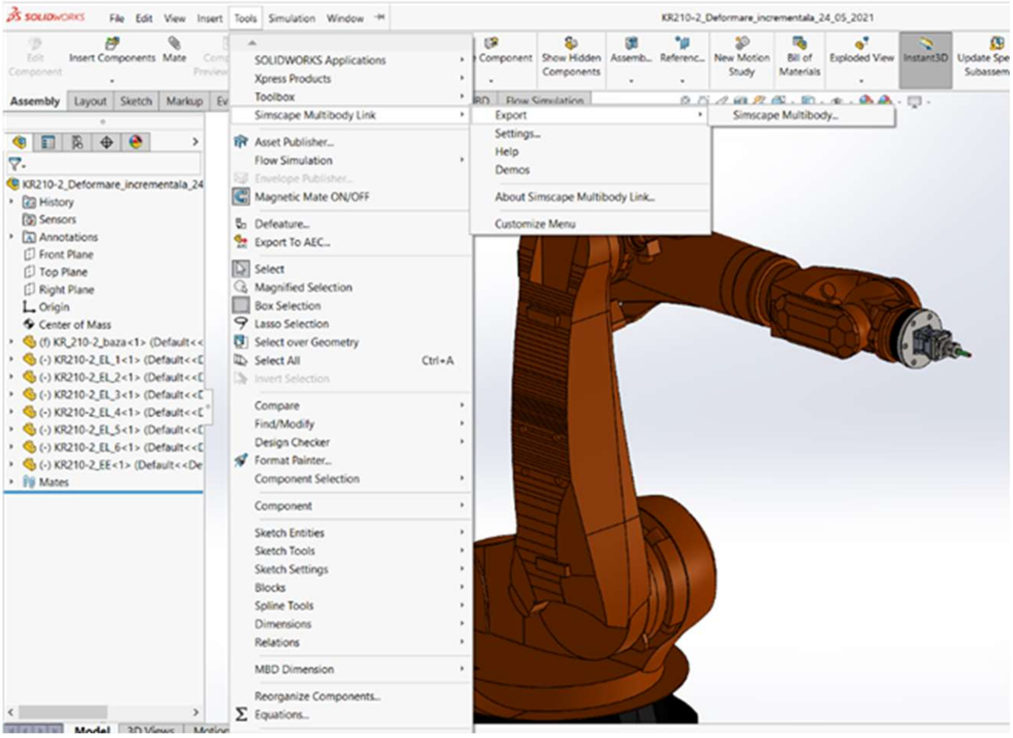


Figure 5. Exporting a CAD model from SolidWorks to Matlab-Simulink Simscape Multibody.

As a result of the export, an xml file was generated, which was named, for example, “KR210-2.xml”. This xml file was imported into Matlab-Simulink with the command:

$$\text{smimport ('KR210-2')} \quad (1)$$

Following import, a Simulink file (KR210_2.slx) and a data file (KR210_2_DataFile.m) were generated.

The combined Simulink–Simscape simulation diagram (the .slx file) is presented in Figure 6, while a screenshot of the graphical simulation obtained by running the simulation diagram is presented in Figure 7.

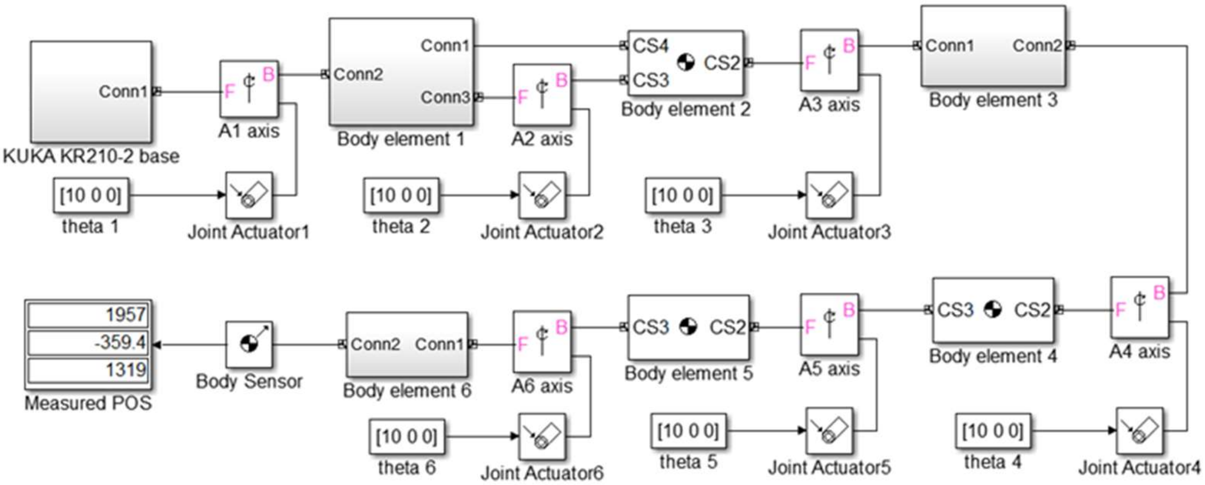


Figure 6. Combined Simulink–Simscape diagram (KR210_2.slx file).

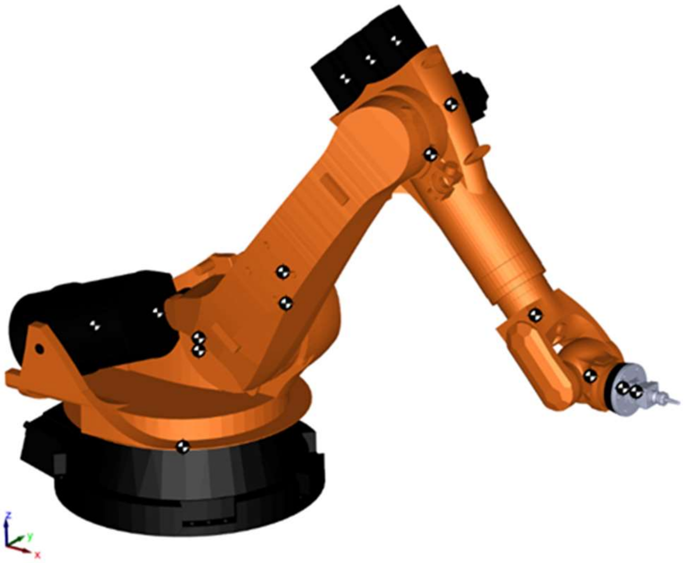


Figure 7. Screenshot from the simulation graphical interface of Simscape—Mechanics Explorer.

At this point, the simulation diagram, as shown in Figure 6, was able to output the coordinates of the end-effector (working unit) using the joint angles as input (direct kinematics).

The simulation diagram was completed with various elements (eg., sensors and actuators) so that it could simulate the dynamic behavior of the robotic structure. Thus, the values of the technological forces were measured and saved by the data acquisition system (QuantumX—MX840A, Hottinger Brüel & Kjaer GmbH, Darmstadt, Germany) working in tandem with the 3D force sensor (PCB 3-component force sensor—PCB261A13, PCB Piezotronics, Depew, NY, USA), Figure 8.

The data acquisition system allowed the simultaneous recording of the values of the three components (Fx, Fy, and Fz) of the deformation force, working at an acquisition frequency of 50 Hz. The software package used for data acquisition and analysis was Catman, from HBM (Hottinger Brüel & Kjaer GmbH, Darmstadt, Germany).

The calculation of the real values of the components of the deformation forces was made by means of the interdependence relationship between the value of the received electrical signal and the magnitude of the force, established with the help of the calibration curves.

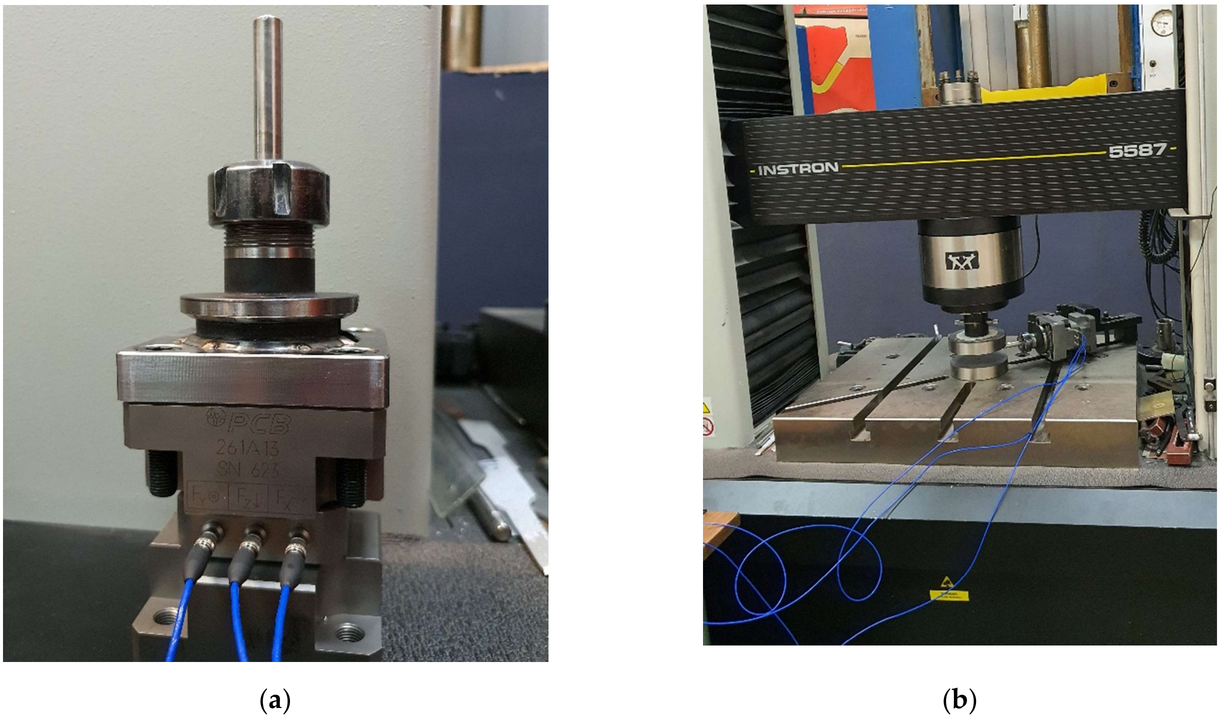


Figure 8. 3D force sensor: (**a**) Integration with the end effector; (**b**) Calibration using INSTRON 5587 tensile testing machine.

The calibration of the sensor was achieved by loading it with a certain force in each direction (X, Y, and Z) on the tensile test machine INSTRON 5587 (Instron, Norwood, MA, USA).

The diagram presented in Figure 9 shows the subsequent feed with the imposed trajectory (coming from the CAM program), which was further converted by the Simscape model into motion data (angle, speed, and acceleration for each joint), and its outputs of the torques in each joint.

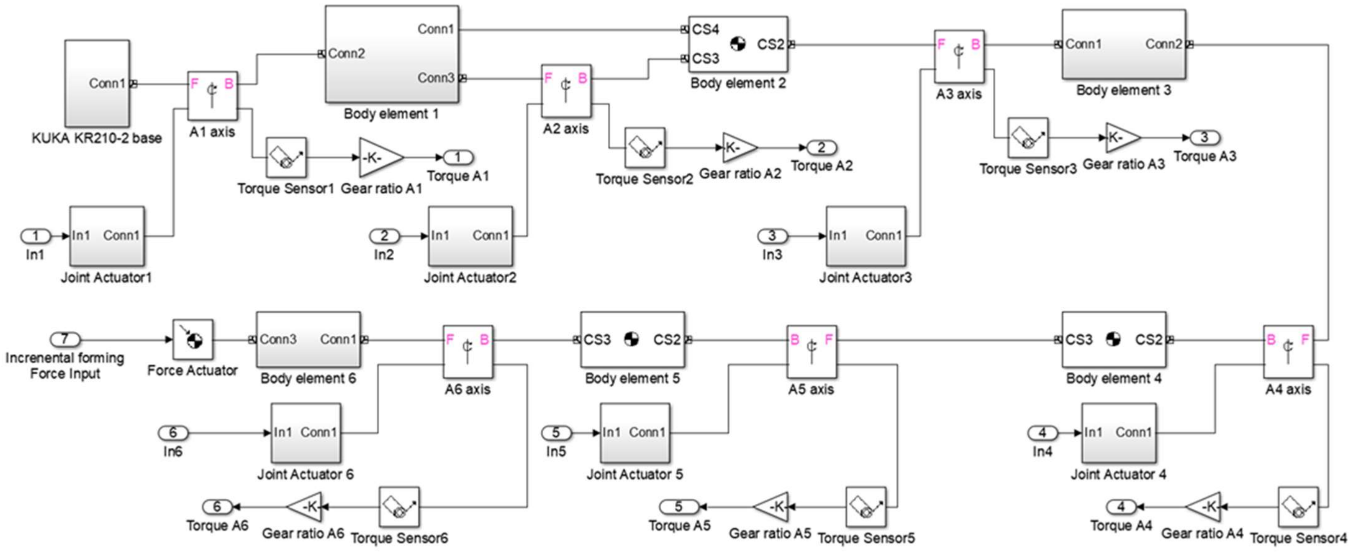


Figure 9. Simulation diagram using force and motion as inputs and joint torques as outputs.

It is evident that the simulation diagram shown in Figure 10 relies on inputs for the joint angles (inverse kinematics resolution for an imposed trajectory). The inputs were sourced from the CAM program, which generated the code for driving the robot on the toolpaths, and were generated as absolute positions and orientations of the end-effector. Thus, the simulation diagram requires the inverse kinematics block to drive the robot on an imposed trajectory, as presented in Figure 10.

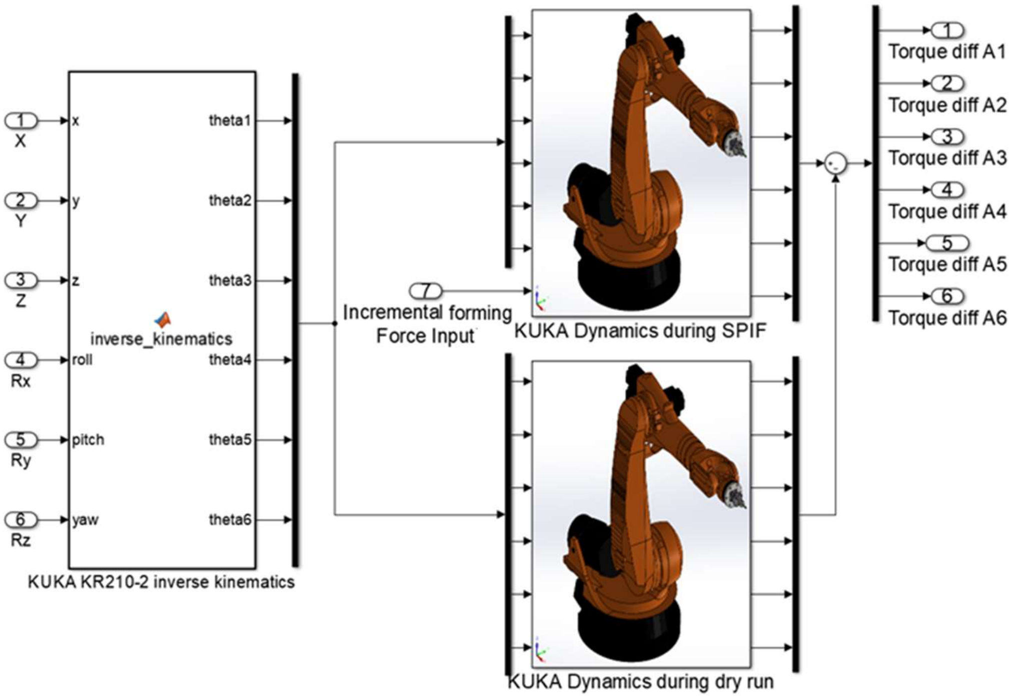


Figure 10. Final simulation diagram for calculating the differential joint torques.

Moreover, because the mechanical properties of the robotic structure (i.e., the center of mass and the moment of inertia) were not precisely known, the simulation diagram shown in Figure 10 was structured in a differential manner. Consequently, the differential joint torques were calculated by subtracting the values obtained during the dry run (with no technological force data needed) from the values obtained during the incremental forming process. By use of this subtraction method, only the dynamic effects due to the process (i.e., the additional joint torques) were considered, and the influences of the center of mass and moment of inertia of the robotic structure (which are not accurately known) were removed from the system.

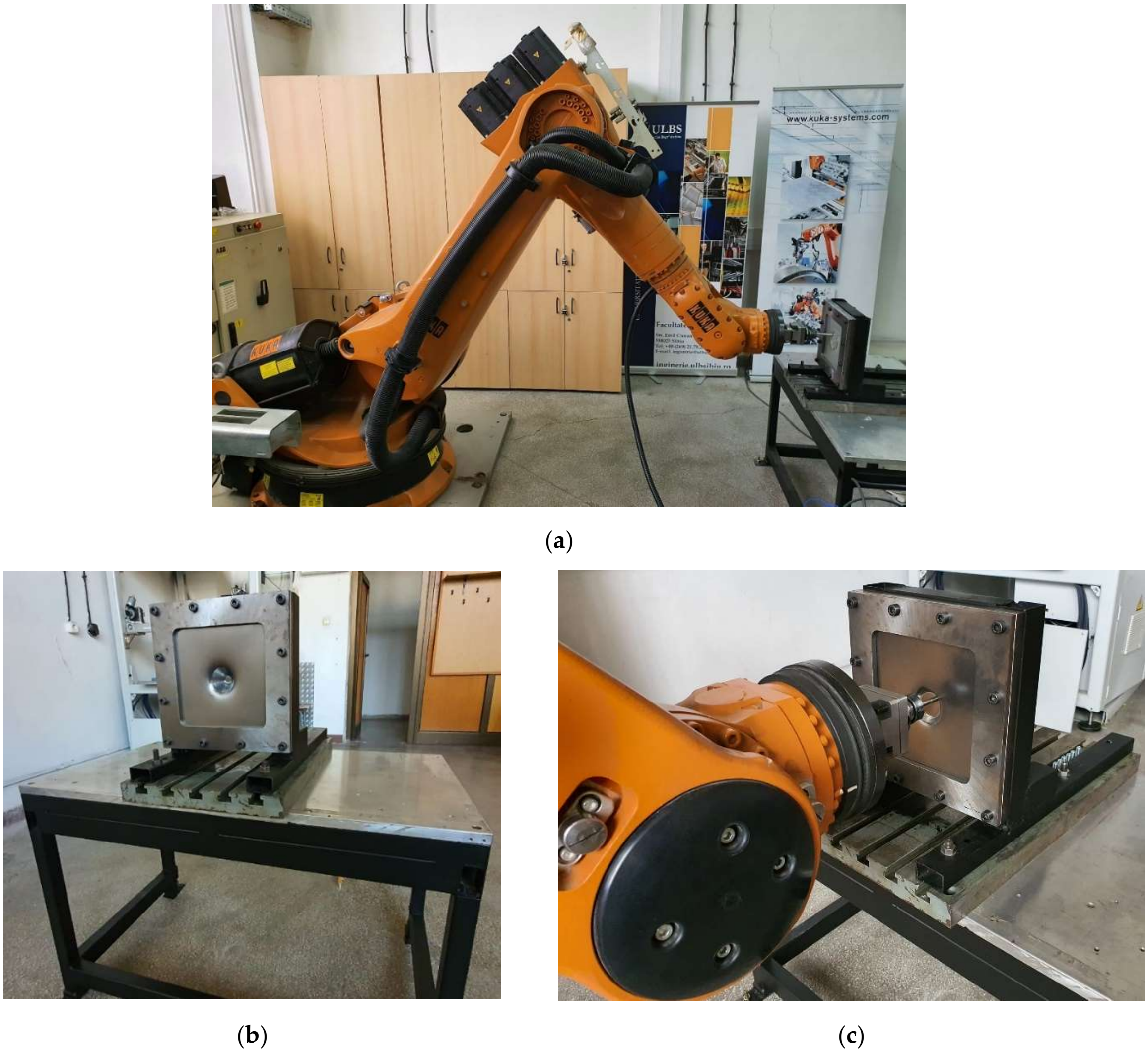
3. Results

Figure 11 presents the experimental layout composed of the KUKA KR 210-2 robot equipped with the incremental forming unit, 3D sensor, data acquisition system, and workpiece fixing system.

The code for driving the end-effector of the robot along the processing toolpaths while avoiding collisions and singularities was generated by means of a CAM software package, SprutCAM. The CAM program required a kinematic model of the robot, together with a 3D model of the workpiece fixing system, in order to be able to generate the code.

The kinematic model of the robot was adapted by the users, who added the 3D model of the end-effector to the KUKA KR 210 robot model existing in the library of the CAM software. The 3D model of the fixing system was developed entirely by the users.

The kinematic model of the robot and the 3D model of the workpiece fixing system, essentially acting as digital twins of the system, are presented in Figure 12.



(**a**)

(**b**)

(**c**)

Figure 11. Experimental layout for incremental forming: (**a**) Overall view of the layout; (**b**) workpiece fixing system; (**c**) Image taken during incremental forming process.

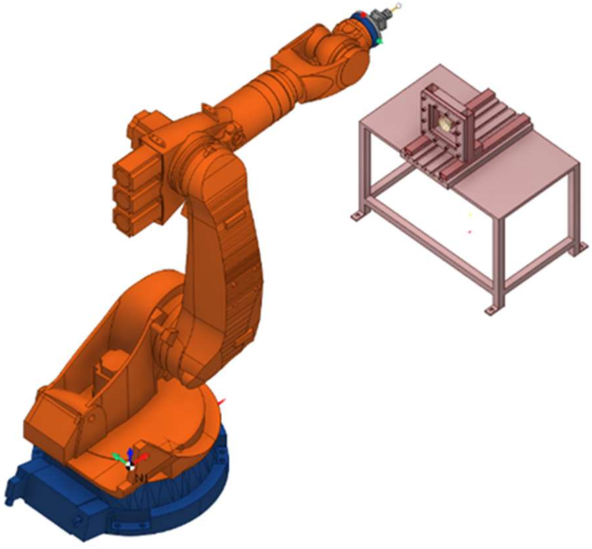


Figure 12. Kinematic model of the KUKA KR 210-2 robot and workpiece fixing system used by the CAM software package.

To test the proposed approach, a cone-shaped frustum part was processed from a sheet metal workpiece by means of incremental forming. The workpiece material was DC04 steel, and the thickness of the sheet was 0.8 mm.

The geometric characteristics of the processed part are presented in Figure 13.

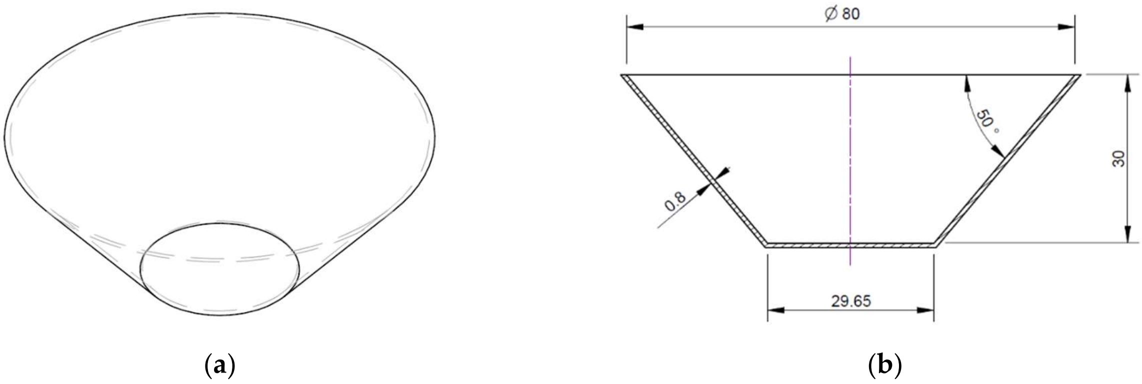


Figure 13. Test part processed by incremental forming: (**a**) 3D model of the part; (**b**) sectional dimensions of the part.

In order to ensure a uniform distribution of the stresses on the surface of the part, and to avoid lead-in and lead-out defects, which could become stress concentrators and consequently could favor the occurrence of cracks, a continuous spiral spatial toolpath, with a maximum step of 1 mm, was used to process the part. The toolpath used for processing the test part is presented in Figure 14.

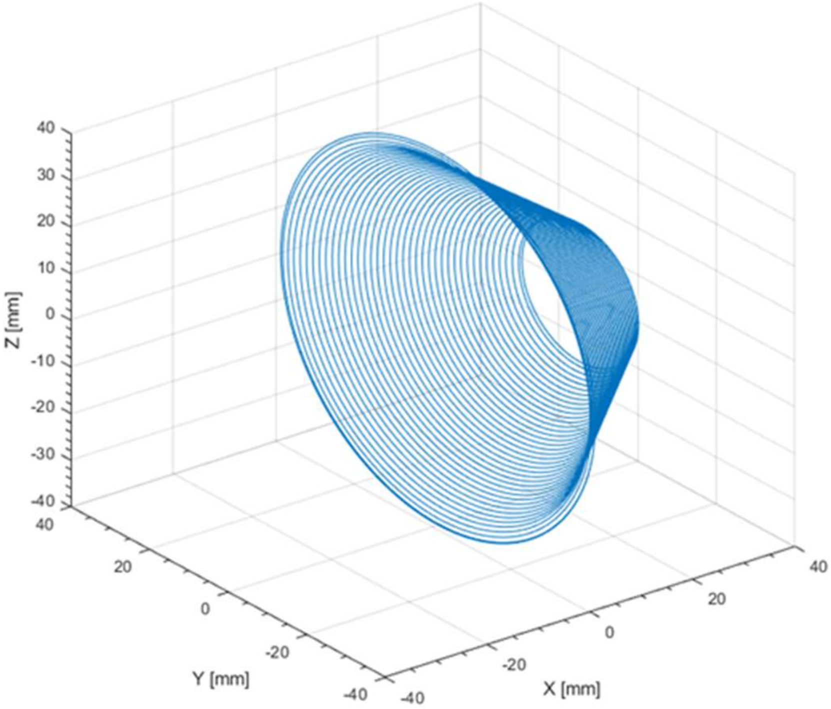


Figure 14. Continuous spiral spatial toolpath used to process the test part.

It is noticeable that only the position of the end-effector changes while the toolpath is followed. The process is presented in Figure 15. The orientation of the end-effector re-mained unchanged because the axis of the tool was kept fixed, in order to ensure the readings generated from the 3D sensor were not affected.

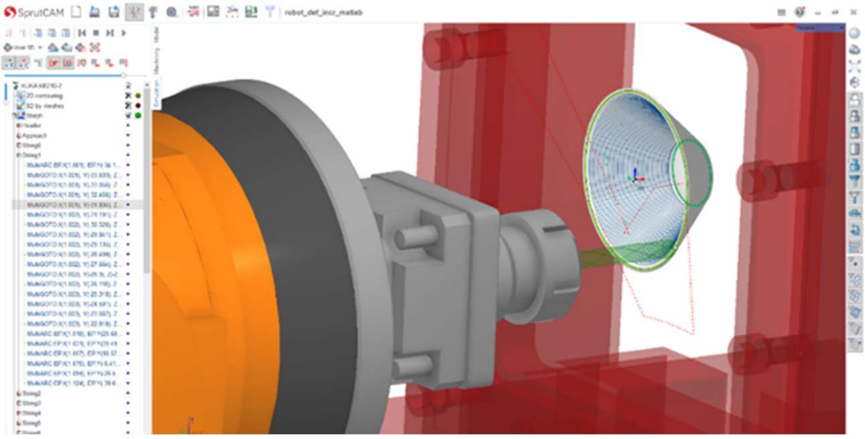


Figure 15. Continuous spatial spiral toolpath used to process the test part.

Figure 16 presents a screenshot from the simulation of the processing of the part in the CAM software. It should be noted that there are no dedicated CAM software packages for incremental forming, and therefore a milling operation was considered as an alternative. Thus, the punch used for incremental forming was replaced with a spherical mill, and the simulation shows the material being removed from between the workpiece and the final part, while in the real process the material was only redistributed. However, these assumptions did not affect the shape of the toolpath presented in Figure 14, and thus the code generated by the CAM software could be used to drive the end-effector (the punch) on the toolpath to process the test part by means of incremental forming.

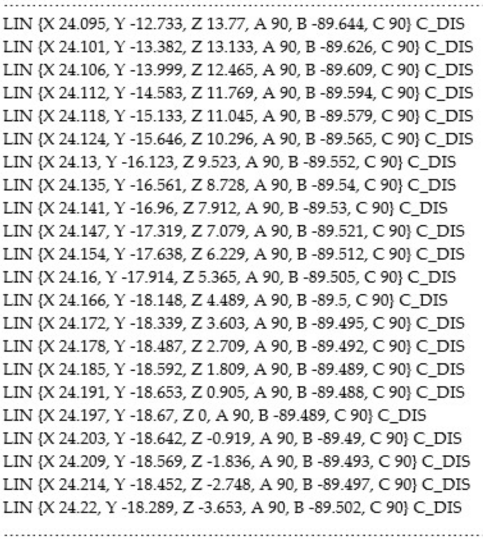

```
...........................................................................
LIN {X 24.095, Y -12.733, Z 13.77, A 90, B -89.644, C 90} C_DIS
LIN {X 24.101, Y -13.382, Z 13.133, A 90, B -89.626, C 90} C_DIS
LIN {X 24.106, Y -13.999, Z 12.465, A 90, B -89.609, C 90} C_DIS
LIN {X 24.112, Y -14.583, Z 11.769, A 90, B -89.594, C 90} C_DIS
LIN {X 24.118, Y -15.133, Z 11.045, A 90, B -89.579, C 90} C_DIS
LIN {X 24.124, Y -15.646, Z 10.296, A 90, B -89.565, C 90} C_DIS
LIN {X 24.13, Y -16.123, Z 9.523, A 90, B -89.552, C 90} C_DIS
LIN {X 24.135, Y -16.561, Z 8.728, A 90, B -89.54, C 90} C_DIS
LIN {X 24.141, Y -16.96, Z 7.912, A 90, B -89.53, C 90} C_DIS
LIN {X 24.147, Y -17.319, Z 7.079, A 90, B -89.521, C 90} C_DIS
LIN {X 24.154, Y -17.638, Z 6.229, A 90, B -89.512, C 90} C_DIS
LIN {X 24.16, Y -17.914, Z 5.365, A 90, B -89.505, C 90} C_DIS
LIN {X 24.166, Y -18.148, Z 4.489, A 90, B -89.5, C 90} C_DIS
LIN {X 24.172, Y -18.339, Z 3.603, A 90, B -89.495, C 90} C_DIS
LIN {X 24.178, Y -18.487, Z 2.709, A 90, B -89.492, C 90} C_DIS
LIN {X 24.185, Y -18.592, Z 1.809, A 90, B -89.489, C 90} C_DIS
LIN {X 24.191, Y -18.653, Z 0.905, A 90, B -89.488, C 90} C_DIS
LIN {X 24.197, Y -18.67, Z 0, A 90, B -89.489, C 90} C_DIS
LIN {X 24.203, Y -18.642, Z -0.919, A 90, B -89.49, C 90} C_DIS
LIN {X 24.209, Y -18.569, Z -1.836, A 90, B -89.493, C 90} C_DIS
LIN {X 24.214, Y -18.452, Z -2.748, A 90, B -89.497, C 90} C_DIS
LIN {X 24.22, Y -18.289, Z -3.653, A 90, B -89.502, C 90} C_DIS
...........................................................................
```

Figure 16. Fragment of the code for driving the KUKA KR 210-2 onto the imposed toolpath.

A short fragment of the code generated by the CAM software solution is presented in Figure 16. It was generated using the absolute position and orientation of the end effector, together with the force values from the 3D sensor, as inputs for the Simulink-Simscape simulation diagram.

The circular interpolation was de-activated from the CAM software settings to obtain only linear segments within the G-code. It should be noted that this fact does not affect the accuracy of the toolpath, but does increase the size of the code. This setting was necessary to achieve the inverse kinematic data absolute position and orientation (input for Simulink-Simscape simulation diagram) in linear segments (point-to-point format), because Matlab is not able to directly process circular interpolation data. One can also notice from Figure 16 that the orientation of the end-effector (angles A, B, C) does not practically change—there are slight modifications in the B angle, but these are smaller than 0.1°, and thus can be neglected.

To test the influence of the workpiece position upon the joint torque values, three different positions were set on the *X* axis, as presented in Figure 17. The offset between the initial position (2432.5 mm from the robot origin) and positions 2 and 3 (±625 mm) was chosen as this was the largest possible discrepancy between the initial experimental layout and the active width of the working table (625 mm).

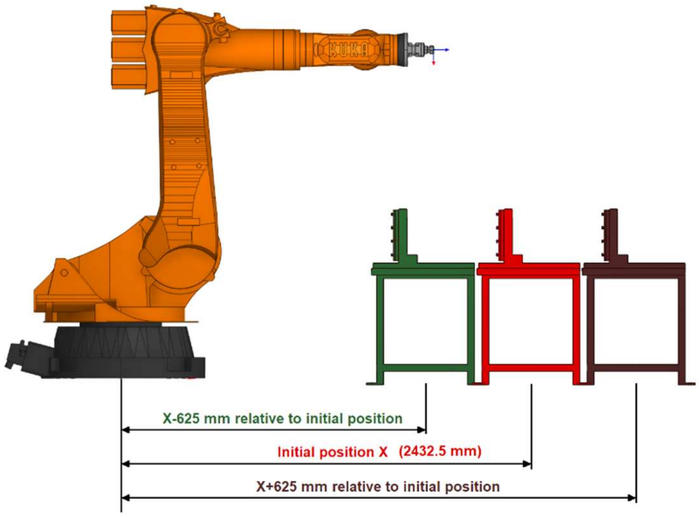

X-625 mm relative to initial position
Initial position X (2432.5 mm)
X+625 mm relative to initial position

Figure 17. Three test positions along *X*-axis.

The processing forces measured using the 3D sensor, presented in Figure 18, were fed as inputs to the Simulink-Simscape diagram to determine the differential joint torques. It can be seen that the data acquisition system of the 3D sensor stored the measured values as a data file, which was further processed for graphical representation (Figure 18), and for data processing (.mat file) in Matlab. The measured force values are provided in a Supplementary File, together with MATLAB/Simulink-Simscape files, and toolpath code. A video file showing an example of part being processed is also provided as Supplementary Material.

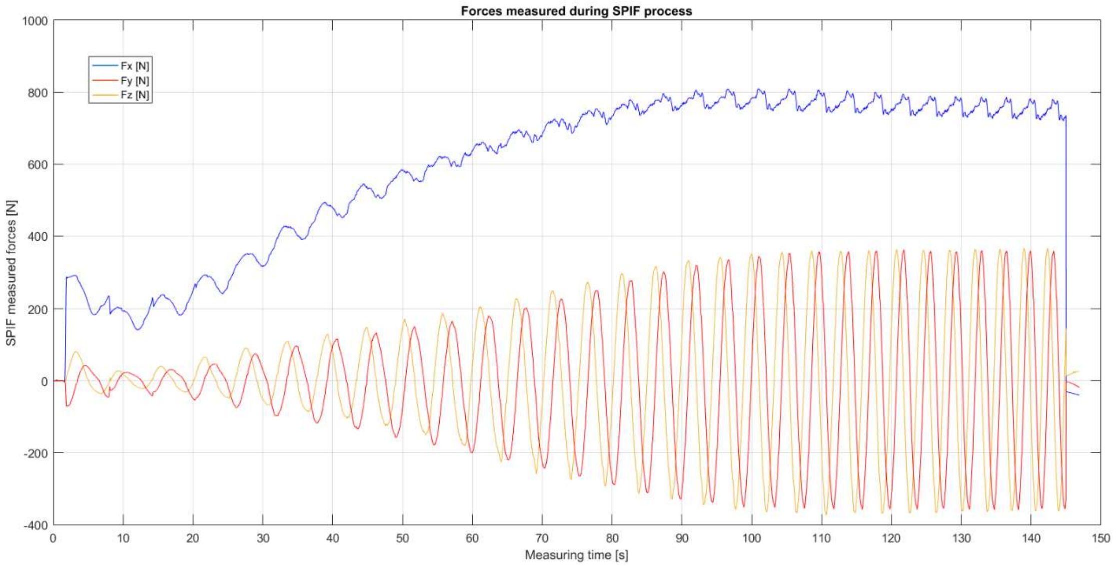
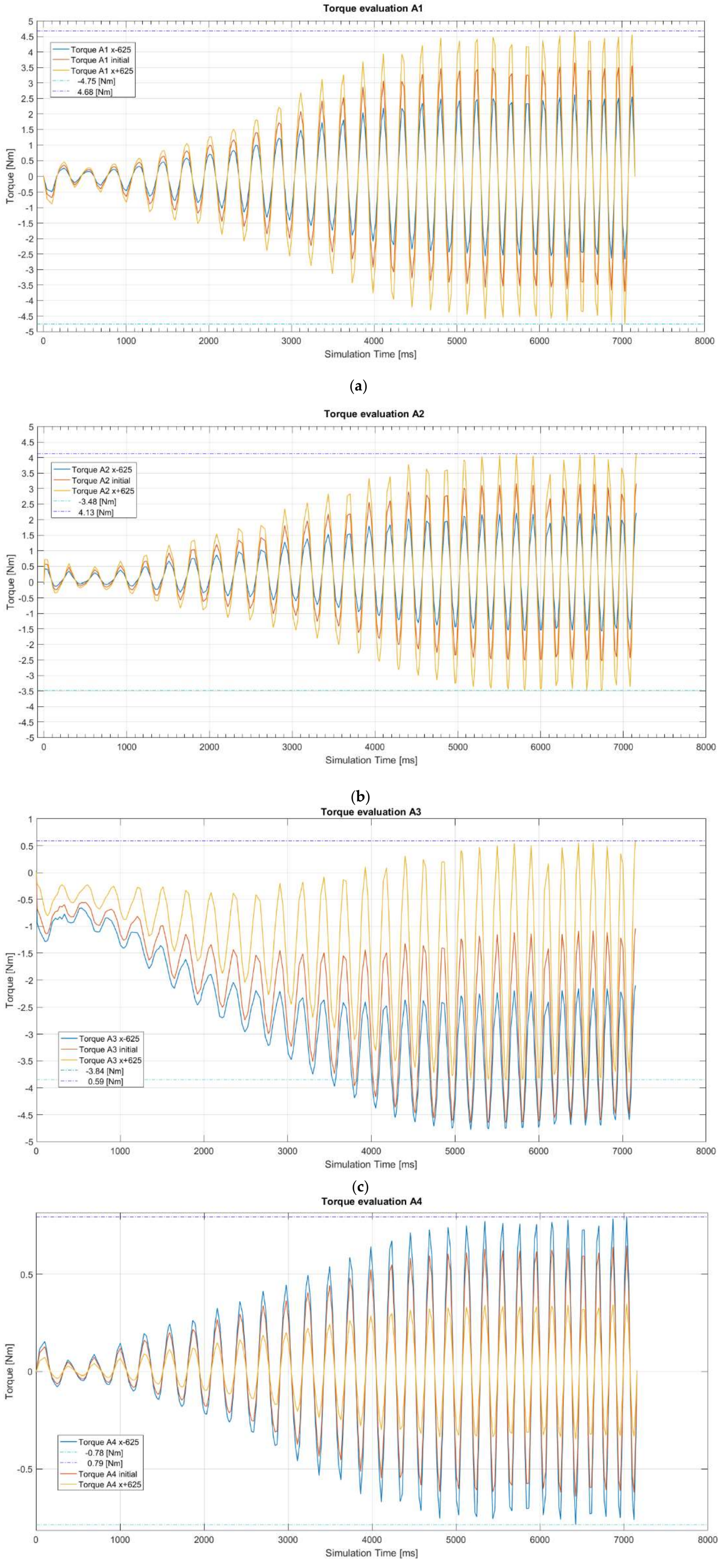


Figure 18. Measured technological forces during incremental forming process.

The simulated differential joint torques (i.e., the difference between the processing torques and the dry run torques) for joints A1 to A6 are presented in Figure 19.



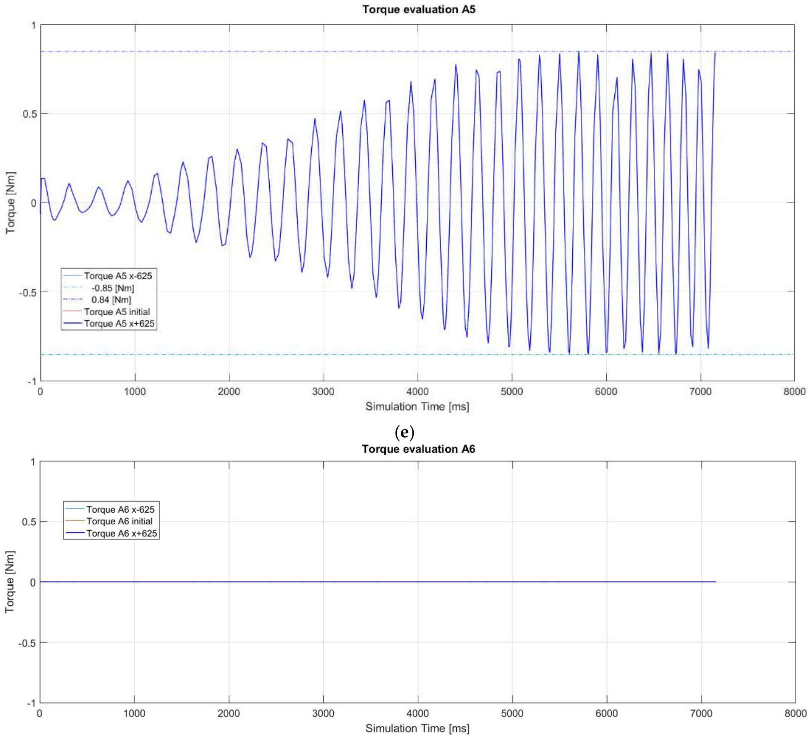
(**a**)

(**b**)

(**c**)

(**d**)

Figure 19. *Cont*.



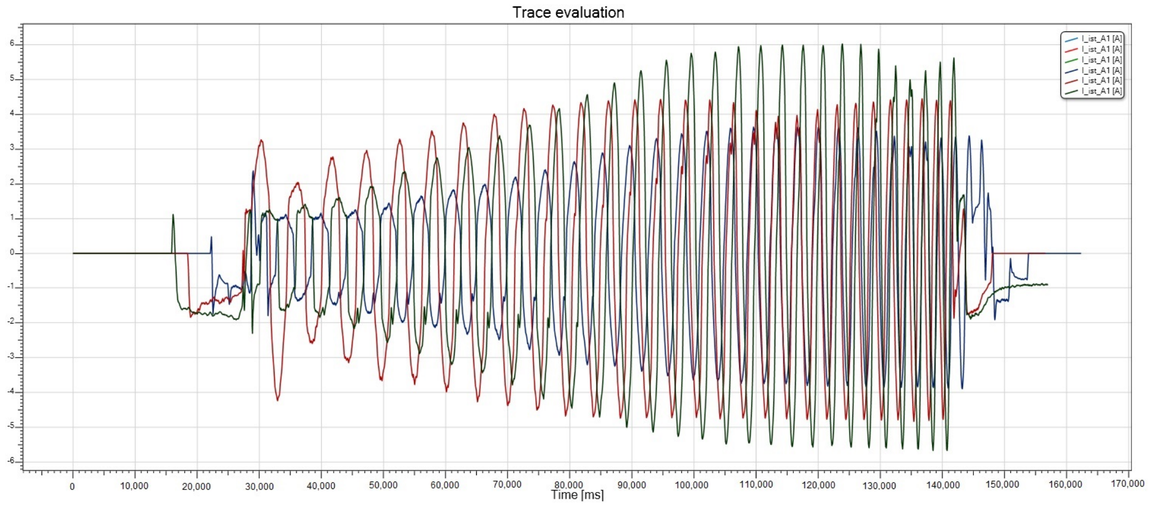
(**e**)

(**f**)

Figure 19. Differential joint torques for three workpiece positions: (a) A1 axis; (**b**) A2 axis; (**c**) A3 axis; (**d**) A4 axis; (**e**) A5 axis; (**f**) A6 axis.

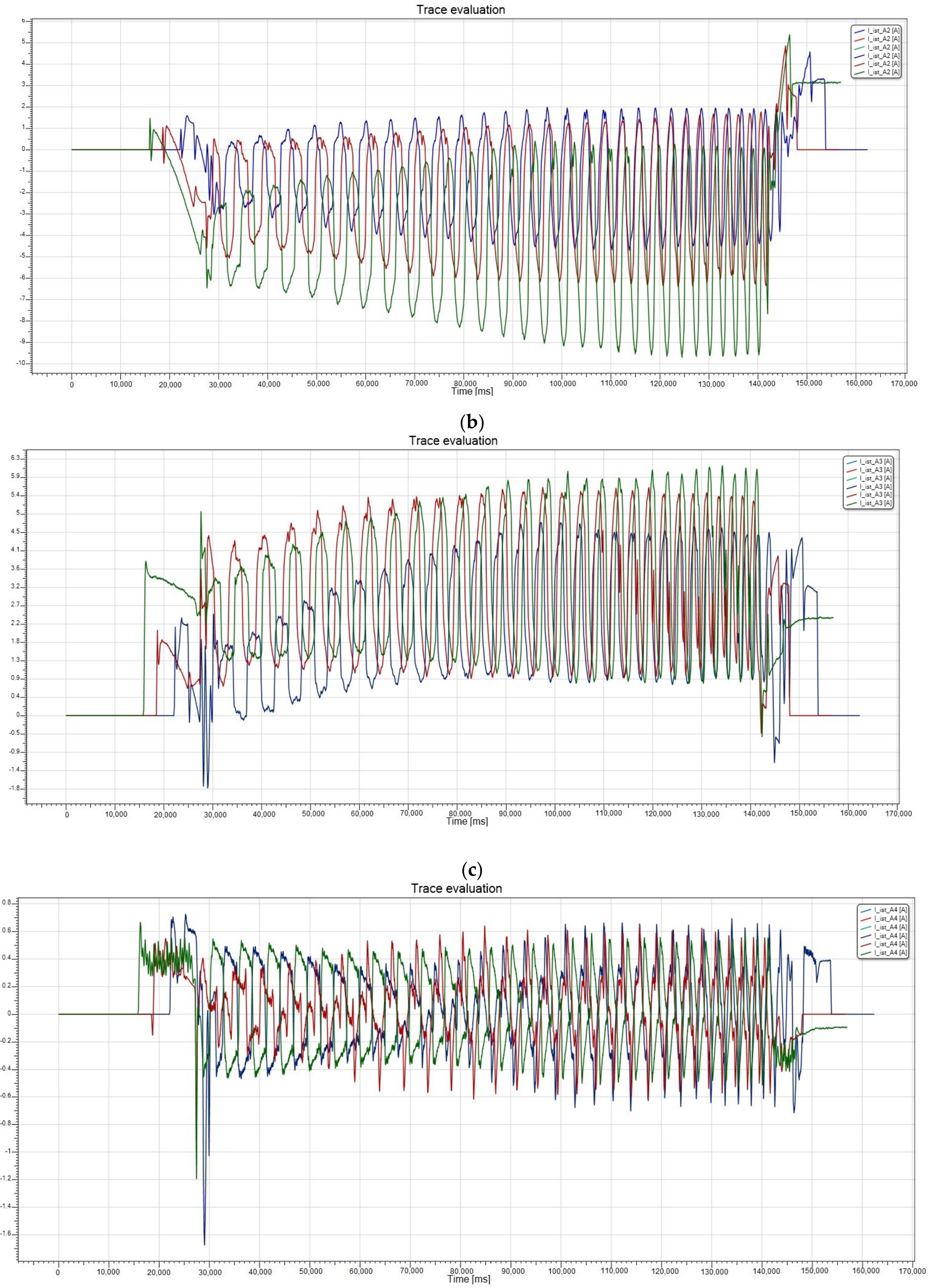
From Figure 19 it can be seen that for the A1 to A4 joint axes, the joint torques decrease together with a decrease in the distance on the *X*-axis between the robot and the workpiece. On the other hand, there are no alterations related to the A5 axis (with similar torques for all three positions) or the A6 axis (with no torques for all three positions).

To qualitatively validate the simulated differential torques, the currents through the joint motors were measured (a feature allowed by the robot controller) during the incremental forming process, and represented graphically using KUKA Visual Work 4.0 (KUKA Deutschland Gmbh, Augsburg, Germany), Figure 20.



(**a**)

Figure 20. *Cont.*

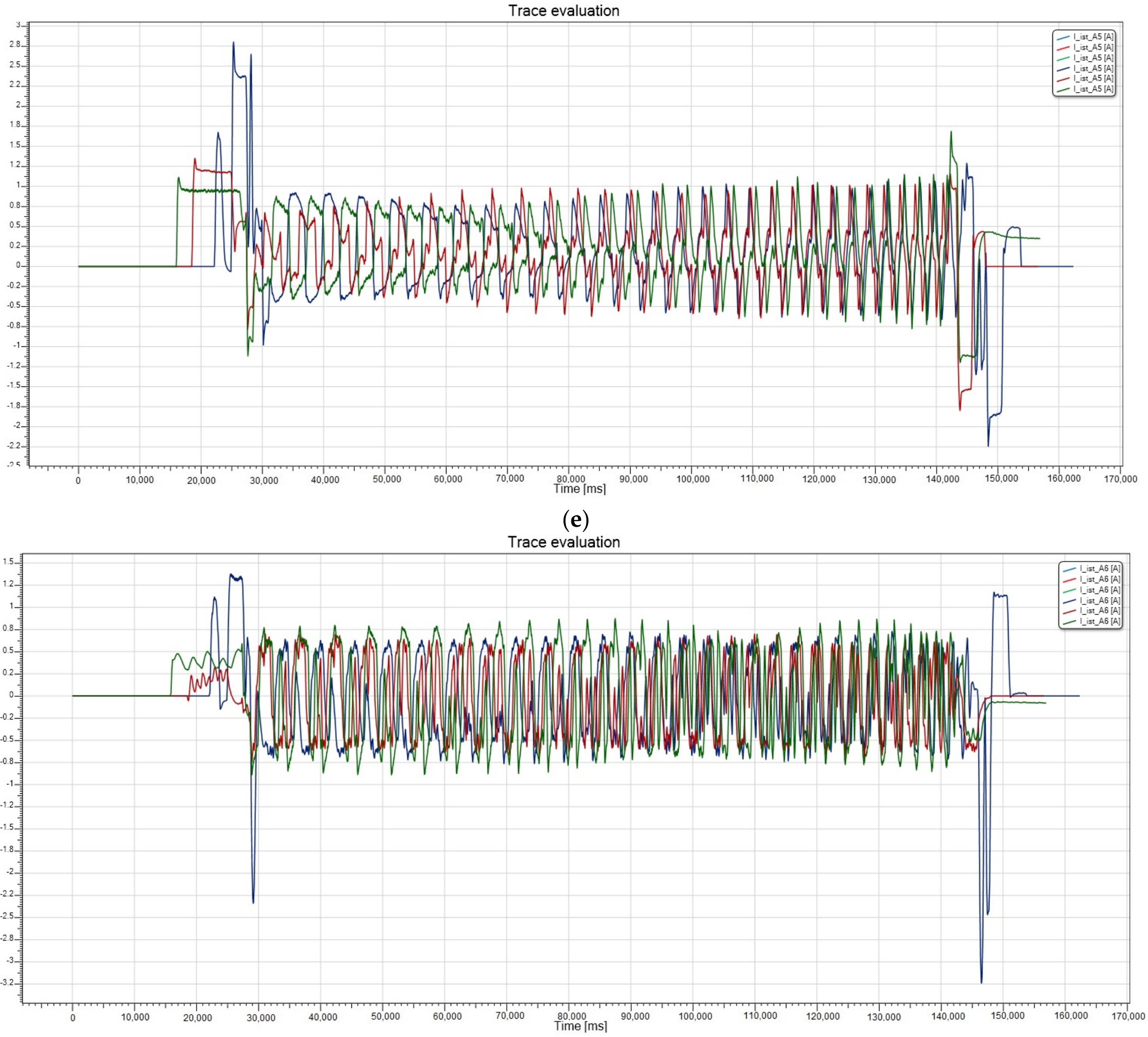


(**b**)

(**c**)

(**d**)

Figure 20. *Cont*.



(**e**)

(**f**)

Figure 20. Currents through joint motors for three workpiece positions: (**a**) A1 axis; (**b**) A2 axis; (**c**) A3 axis; (**d**) A4 axis; (**e**) A5 axis; (**f**) A6 axis.

The colors in Figure 20 represent the following:

- Red—currents for initial position X;
- Blue—currents for X − 625 mm relative to the initial position
- Green—currents for X + 625 mm relative to the initial position

A synthesis of the data related to the simulated differential joint torques and measured currents through the joint motors is presented in Tables 1–4.

Table 1. Simulated differential joint torque values—minimum and maximum values.

Joint Axes	A1	A2	A3	A4	A5	A6
Simulated differential joint torque values MATLAB—minimum and maximum values [Nm]						
X − 625	−2.67, 2.62	−1.57, 2.21	−4.77, 0	−0.78, 0.79	−0.85, 0.84	−0, 0
X	−3.71, 3.65	−2.53, 3.17	−4.64, 0.01	−0.64, 0.64	−0.85, 0.84	−0, 0
X + 625	−4.75, 4.68	−3.48, 4.13	−3.84, 0.59	−0.34, 0.34	−0.85, 0.84	−0, 0

Table 2. Simulated differential joint torque values—maximum ranges.

Joint Axes	A1	A2	A3	A4	A5	A6
Simulated differential torque values MATLAB—maximum ranges [Nm]						
X − 625	5.29	3.78	4.77	1.57	1.69	0
X	7.36	5.7	4.65	1.28	1.69	0
X + 625	9.43	7.61	4.43	0.68	1.69	0

Table 3. Currents through joint motors—minimum and maximum values.

Joint Axes	A1	A2	A3	A4	A5	A6
Currents through joint motors KUKA—minimum and maximum values [A]						
X − 625	−3.89, 3.63	−4.67, 2.02	0.71, 4.76	−0.72, 0.72	−0.66, 1.12	−0.75, 0.76
X	−4.83, 4.46	−6.4, 1.8	0.9, 5.63	−0.61, 0.66	−0.64, 1.03	−0.88, 0.88
X + 625	−6.35, 6.5	−9.67, 0.43	0.75, 6.18	−0.54, 0.59	−0.75, 1.16	−0.89, 0.88

Table 4. Currents through joint motors—maximum ranges.

Joint Axes	A1	A2	A3	A4	A5	A6
Currents through joint motors KUKA—maximum ranges [A]						
X − 625	7.52	6.69	5.47	1.44	1.78	1.51
X	9.29	8.02	6.53	1.27	1.67	1.76
X + 625	12.85	10.01	6.93	1.13	1.91	1.77

A graphical summary of the significant values extracted from Figures 19 and 20 is presented in Figures 21–26.

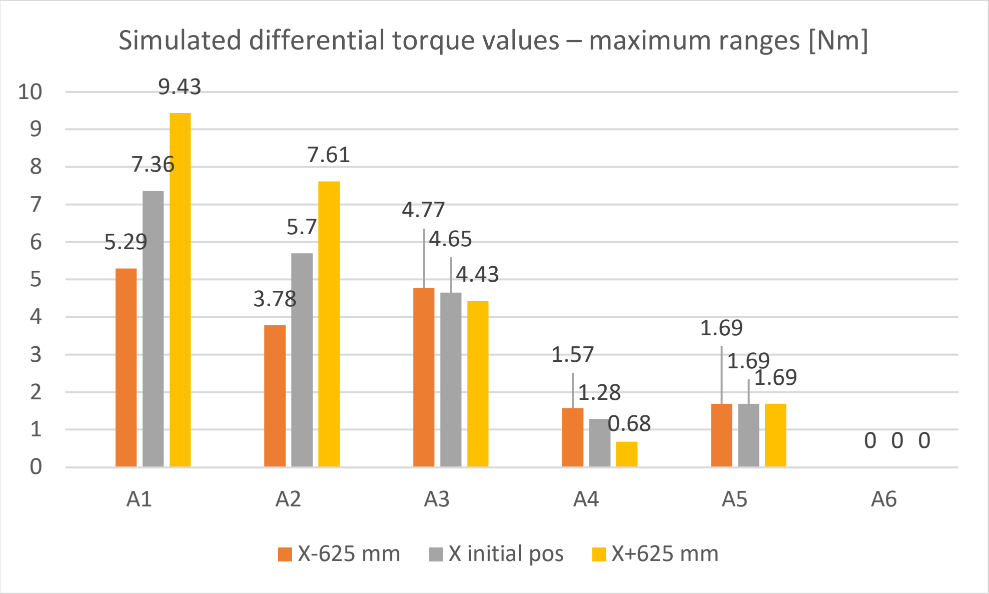


Figure 21. Simulated differential torque values MATLAB—maximum ranges [Nm].

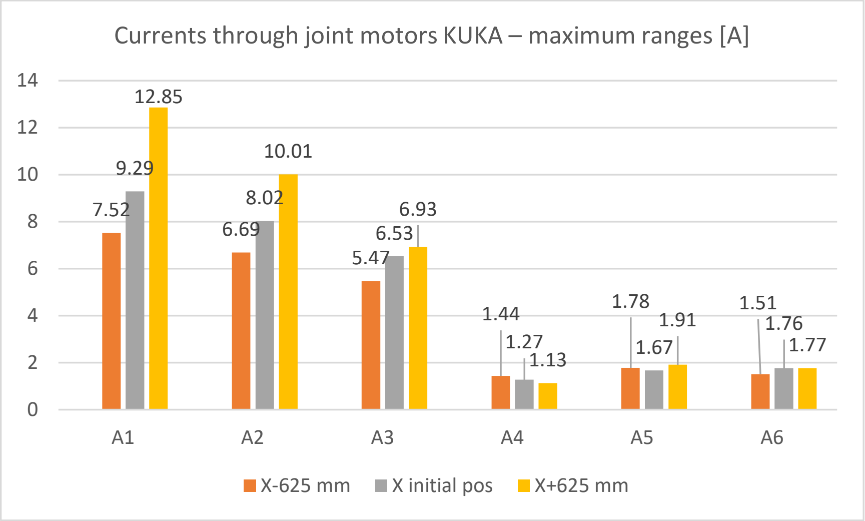


Figure 22. Currents through joint motors KUKA—maximum ranges [A].

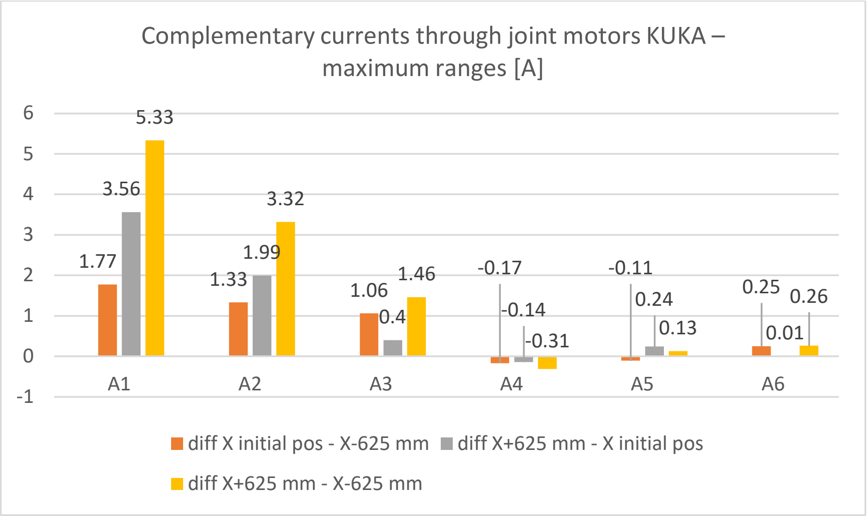


Figure 23. Complementary currents through joint motors KUKA—maximum ranges [A].

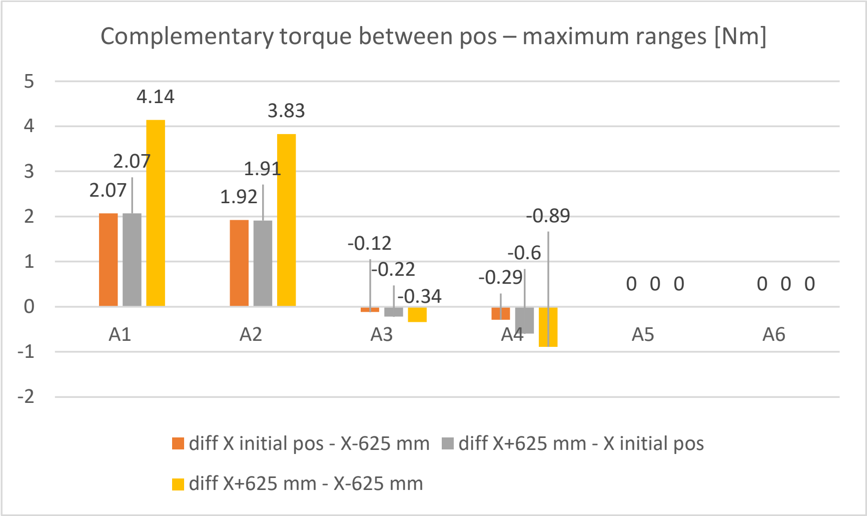


Figure 24. Complementary torque between positions—maximum ranges [Nm].

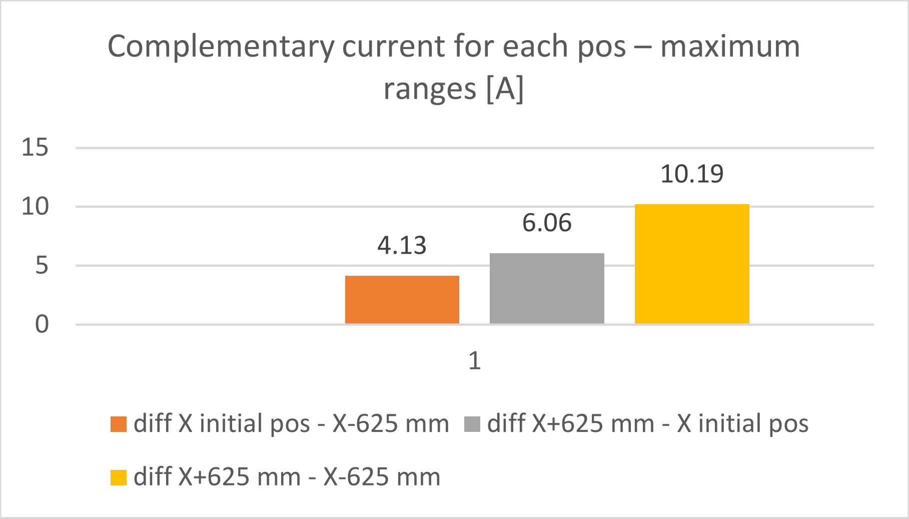


Figure 25. Complementary current for each position—maximum ranges [A].

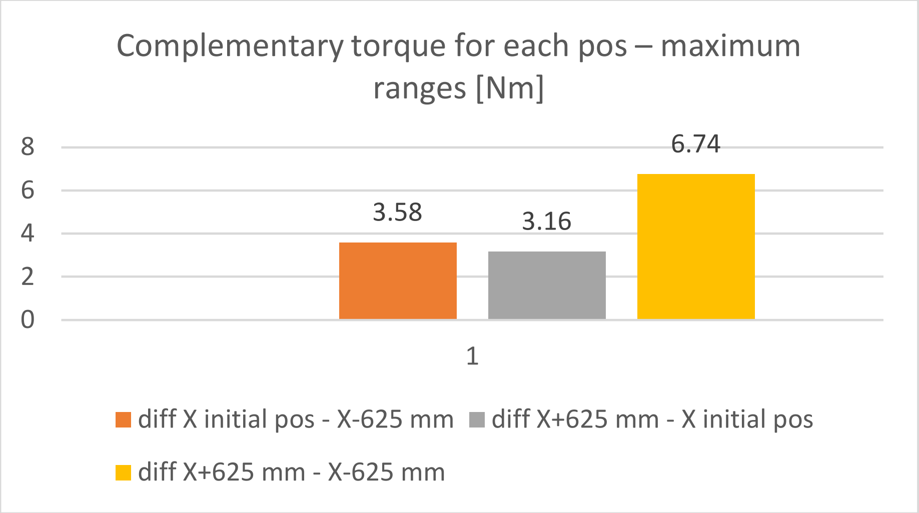


Figure 26. Complementary current for each position—maximum ranges [Nm].

It can be seen that the positioning of the workpiece at a smaller distance from the base of the robot leads to a decrease in the current consumed in the joints A1, A2, and A3, and also to a decrease in the torques determined by simulation. Given the dynamics of the open kinematic chains, the joints A1, A2, and A3 have the highest contribution to the energy consumption of the KUKA KR210-2 industrial robot.

The fact that the energy consumption and the torques at the joints level both decreased, led to a better rigidity of the kinematic chain, and at the same time this should result in an increased dimensional accuracy of the manufactured parts; this will be studied in future research. The results also show that the rigidity of the system increased with the positioning of the workpiece nearer the base of the robot.

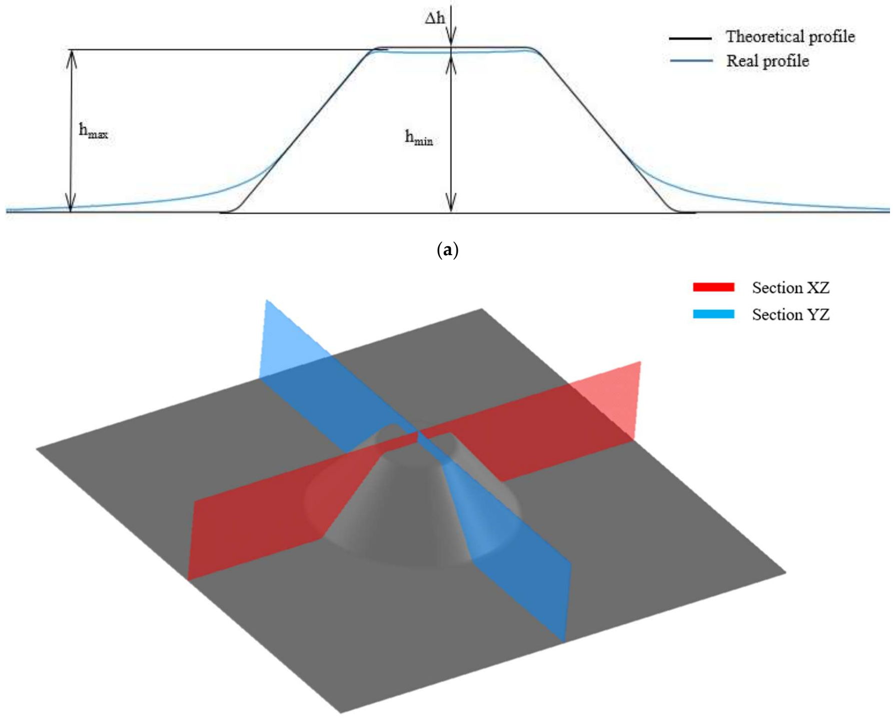
Future research directions will analyze the outcomes when the workpiece is repositioned from the base of the robot on both the longitudinal (OX) and the vertical (OZ) axes.

Statistically analyzing the data obtained from both the simulations and the experimental measurements, it can be concluded that the energy consumption during the incremental deformation of thin sheets decreased when the workpiece was placed closer to the base of the robot on the *OX* axis.

Processed Parts

After evaluating the joint torques and the currents evolution, three similar parts, positioned at the initial position (2432.5 mm from the robot origin) and at positions ±625 mm relative to that, were processed in order to assess the influence of the workpiece position upon the shape and dimensional accuracy of the parts.

The determination of the accuracy of the dimensions and shape of the parts was achieved using a 3D optical measurement system—GOM ATOS Core (GOM Gmbh, Braunschweig, Germany). Two curves, perpendicular to each other, were obtained for a cross section of the part. These curves were extracted and analyzed for each part obtained at X − 625 mm, initial X position, and X + 625 mm. The first measurement of the part profile was set towards the XOZ plane, at 0° in relation to the rolling direction of the sheet metal, and the second measured profile was set towards the YOZ plane. These two planes pass through the middle of the part, so that the information obtained could be considered as having a high degree of trustworthiness (Figure 27).



(**a**)

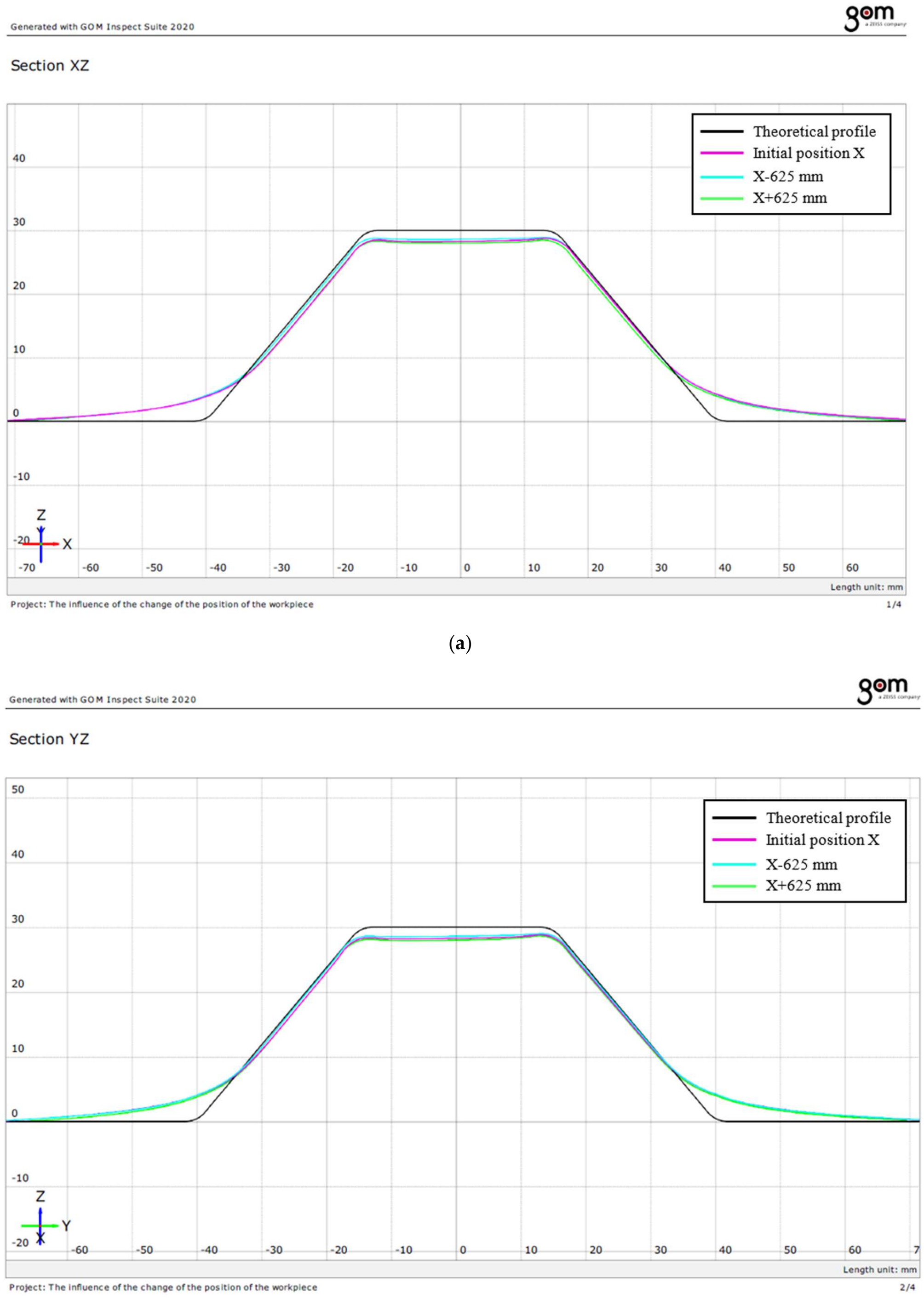
(**b**)

Figure 27. Assessing the shape and dimensional accuracy of the parts: (**a**) theoretical and real profile; (**b**) sections XZ and YZ.

In all the analyzed cases, the level of deviation at the bottom of the part, Δh, was determined in relation to the theoretical profile. These variations in size were analyzed at the point of intersection of each curve with the axis of symmetry of the cone; a synthesis of the results is presented in Table 5, while the measurements are presented in Figure 28.

Table 5. Synthesis of measurements results.

Case	Angle of the Desired Part [°]	Blank Position [mm]	Type of the Trajectory, Spiral Movement	Deviation of the Obtained Part Δh [mm]
C1	50	X	Frustum of a cone	1.72
C2		X − 625		1.38
C3		X + 625		2


Generated with GOM Inspect Suite 2020

Section XZ

Project: The influence of the change of the position of the workpiece 1/4

(**a**)

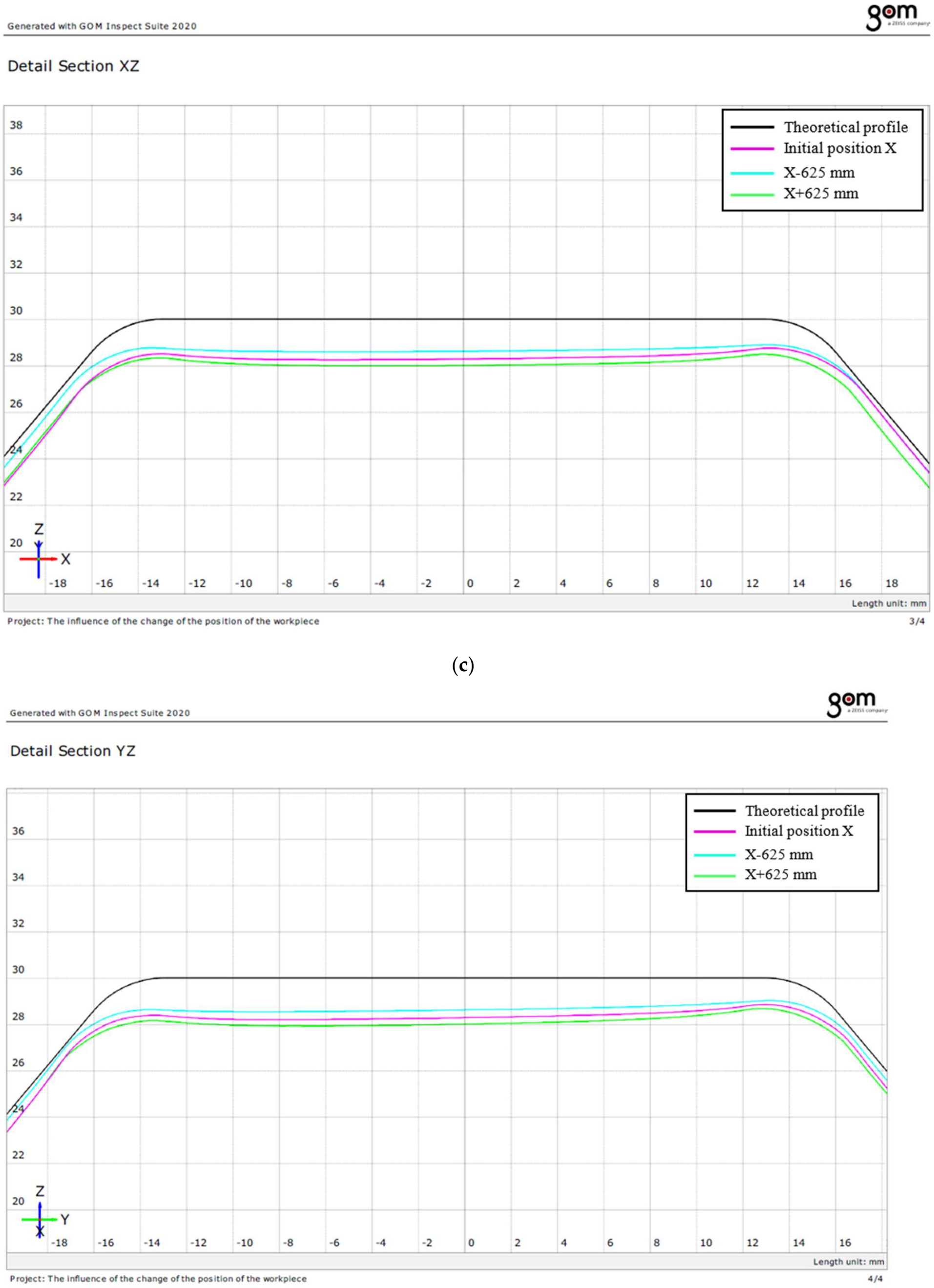
Generated with GOM Inspect Suite 2020

Section YZ

Project: The influence of the change of the position of the workpiece 2/4

(**b**)

Figure 28. *Cont*.



(**c**)

(**d**)

Figure 28. Measured profiles of the parts: (**a**) section XZ; (**b**) section YZ; (**c**) detail section XZ; (**d**) detail section YZ.

Figure 29 presents the upper and lower faces of a processed part.

From Tables 2–5 it can be seen that the joint torque and current values are the lowest for the X − 625 mm position, are higher for the initial position, and are the largest for the X − 625 mm position.

Similarly, the data in Figure 23 and Table 5 prove that the most accurate part was obtained when the workpiece was placed at X − 625 mm; the accuracy decreased for the initial position, and the lowest accuracy was obtained for the workpiece placed at X + 625 mm. Even though more experiments should be performed to confirm this assumption, it can be stated that a direct link could be established between the values of the joint torques and currents estimated by means of the proposed method and the actual accuracy of the parts based on the proposed workpiece placement.

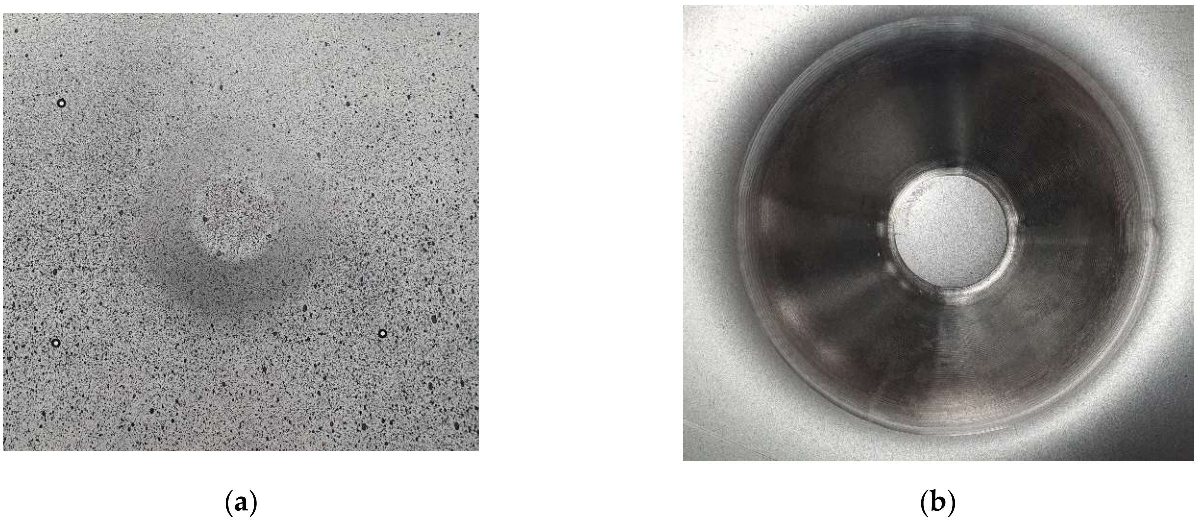


Figure 29. Processed part: (**a**) upper face; (**b**) lower face.

4. Discussion

A comparative analysis of Figures 19 and 20 shows the following:

- The variation in the joint torques (differential) as a function of time had a similar evolution to that of the currents through the joint motors, from a qualitative point of view;
- Both the torques and the currents became smaller as the distance between the workpiece and the robot decreased (also confirmed by the values shown in Tables 2–5). Of course, there are functional limits to the minimum value of this distance;

At present, a quantitative analysis to validate the calculated values of the joint torques based on the measured currents cannot be made, due to the unknown values related to the moments of inertia of the dimensional components and the frictions in the robotic structure, meaning the joint torques were calculated in a differential variant.

A solution to this problem could be to measure the currents through the joint motors differentially (i.e., in the dry run regime and in the working regime) and to apply the formula:

$$\mathrm{M} = k_t \cdot \mathrm{I} \tag{2}$$

where M is the torque developed by the motor, k_t is the motor torque constant and I is the motor current.

However, unfortunately, the torque motors are special variants, developed by Siemens, Munich, Germany for KUKA robots, and at the moment their torque constants are not known. Nevertheless, it can be stated that the proposed approach allows the user to estimate the influence of the workpiece position upon the values of the joint torques.

Consequently, by combining the Matlab-Simulink Simscape model (also called the CAE model) of the robotic structure with the CAM model, the user was able to:

- Calculate the motion (positions, velocities, and accelerations) of the robotic structure in direct and inverse kinematics using the CAE model, while checking the positions using the CAM model;
- Issue the code for driving the robot on any given toolpath, avoiding collisions and singularities, using the CAM model, while checking and reducing the joint torques (by proper workpiece placement) using the CAE model.

Thus, it can be considered that the proposed approach of integrating and interconnecting the CAE and the CAM models enabled the creation of a digital twin of the complex mechatronic system consisting of the industrial robot and the workpiece fixing system (Figure 30).

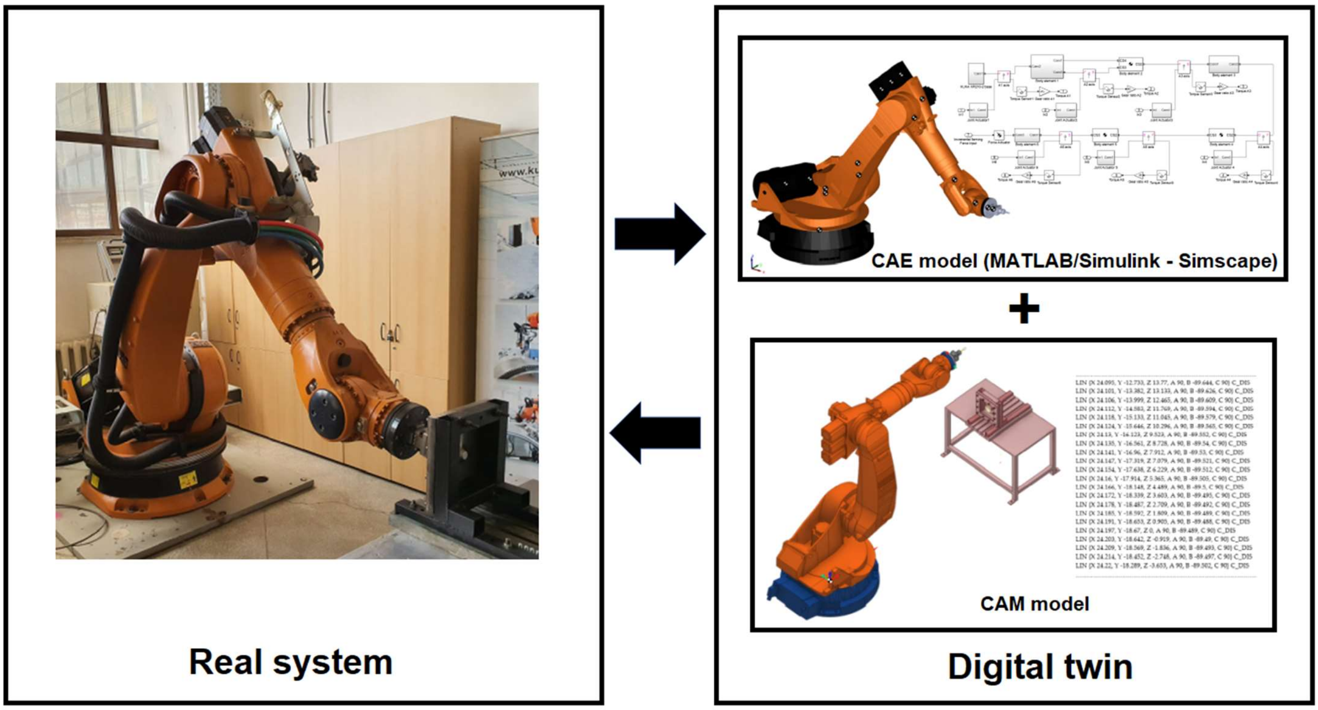


Figure 30. Digital twin of the real system created by interconnecting the CAE and CAM models.

5. Conclusions

Industrial robots are complex mechatronic systems, and their use in machining processes which require continuous path control raises specific challenges. The main problems are related to the low rigidity of the robotic structure and the limited accuracy of the machined parts.

The incremental forming manufacturing process is especially suited to the use of robots because the demands related to the precision of the parts are lower than in the case of cutting manufacturing processes. On the other hand, however, the technological forces involved can lead to the appearance of joint torques that exceed the admissible values.

The present study aimed to build and validate, at least from a qualitative point of view, a digital twin of a real incremental forming processing system by means of a KUKA KR 210-2 industrial robot. We showed that by integrating CAE and CAM models, the user can program the tool trajectory necessary for processing the part, analyze the direct and inverse kinematics of the robot, and, moreover, estimate its dynamic behavior, especially its joint torque values. This digital twin, among other things, allows the user to estimate the influence of the change of the position of the workpiece on the dynamic demands of the robot, and more implicitly, on the energy consumption during processing.

The main outcomes of this work can be summarized as:

- The development of an initial Matlab-Simulink Simscape Multibody model starting from the 3D CAD model of the KUKA KR 210-2 robot. It is notable that the process of building the model by simply importing the xml file generated by the CAD software was not automatic; manual intervention to correct the importing inaccuracies was required;
- The integration of direct and inverse kinematics into the Matlab-Simulink Simscape Multibody model, and its validation through comparing the end-effector position and orientation issued by the model with the ones issued by dedicated CAM software;

- The integration into the model of the readings from an external force sensor fitted onto the robot, using the motion information from the CAM software as inputs and calculated joint torques as outputs;
- The development of a differential joint torques calculation method to remove the influence of inaccurate data (i.e., the center of mass and moment of inertia of the robotic structure);
- The development of a joint simulation and experimental approach to assess the influence of the workpiece position upon the values of the joint torques, and indirectly upon the energy consumption of the process. Moreover, the experimental work qualitatively showed that the workpiece position associated with a lower energy consumption can also be related to a higher accuracy of the processed part.

Further research will aim to:

- Improve the robot CAE model by better determining the masses and moments of inertia of the structural components and the frictions in the robot joints;
- Improve the experimental validation of the torque values calculated by the CAE model;
- Consider workpiece placement optimization, by means of a set of experiments based upon experimental planning;
- Consider the influence of the robot dynamics upon the toolpaths and how these can be optimized.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/machines10070531/s1, Archive S1: archive with MATLAB/Simulink-Simscape files, forces data file and toolpath code file, Video S1: video showing an example of a part being processed.

Author Contributions: Conceptualization: S.-G.R., M.C., A.B. and R.-E.B.; methodology: R.-E.B., M.C., S.-G.R., C.-E.G., M.T. and C.-M.B.; investigation: R.-E.B., S.-G.R., A.B., M.C. and M.T.; resources: S.-G.R., C.-E.G. and R.-E.B.; writing—original draft preparation: R.-E.B.; writing—review and editing: R.-E.B.; project administration: S.-G.R., M.C., C.-M.B. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Romanian Ministry of Research, Innovation and Digitization through Program 1—Development of the National Research-Development System, Subprogram 1.2-Institutional Performance- Projects to Finance Excellence in RDI, Contract no. 28PFE/30 December 2021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A

To solve the inverse kinematics for a serial robot, one must obtain the relative movements in each kinematic joint, knowing the absolute position and orientation of the end-effector. When an analytical problem resolution is not possible because the robot structure has a more general configuration, numerical approximations must be adopted. In this case, the mechanical structure of the KUKA KR 210-2 robot allows an analytical resolution of the inverse kinematic problem, and therefore a geometrical approach resolution was adopted.

For this geometrical approach one must start from the imposed path/trajectory of the end-effector. Knowing the X, Y, Z, φ_x, φ_y, and φ_z positions and the orientation of the end effector, one can determine the θ_i, and the relative movements for each kinematic joint by reducing the geometric structure into smaller parts.

The first step of the inverse kinematics resolution is to build the total homogenous matrix for the position and orientation of the end-effector:

$$\mathrm{H = T_x \cdot Ty \cdot Tz \cdot Rz \cdot Ry \cdot Rx}, \tag{A1}$$

where: Tx, Ty, and Tz, represent the imposed OX, OY, and OZ absolute positions of the end-effector, respectively, and R_x, R_y, and R_z, represent the imposed OX, OY, and OZ absolute orientation of the end effector, respectively.

Knowing the absolute position of the end-effector to simplify the robotic structure, one must obtain the wrist position of the robot. This position is marked as C in Figure 3.

$$\mathrm{HC} = \mathrm{H} \cdot \begin{bmatrix} 1 & 0 & 0 & -\mathrm{l}6 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{A2}$$

Once the wrist position is known, the first base rotation can be computed using the following expression:

$$\theta_1 = \mathrm{atan2(YC, XC)} \tag{A3}$$

After this first parameter determination, the shoulder position can be obtained using the following mathematical expression:

$$\mathrm{HA = A1}. \tag{A4}$$

The second parameter, the θ_3 joint angle, is easier to determine. For this parameter one must apply the law of cosines for an ABC triangle, per Figure A1:

$$\mathrm{L}^2 = \mathrm{l}3^2 + \mathrm{L}1^2 - 2 \cdot \mathrm{l}3 \cdot \mathrm{L}1 \cdot \cos(\alpha). \tag{A5}$$

Consequently,

$$\alpha = \arccos\left(\frac{\mathrm{L}^2 - \mathrm{l}3^2 - \mathrm{L}1^2}{-2 \cdot \mathrm{l}3 \cdot \mathrm{L}1}\right), \tag{A6}$$

where:

$$\mathrm{L} = \sqrt{(\mathrm{XC} - \mathrm{XA})^2 + (\mathrm{YC} - \mathrm{YA})^2 + (\mathrm{ZC} - \mathrm{ZA})^2}, \tag{A7}$$

and

$$\mathrm{L}1 = \sqrt{\mathrm{l}5^2 + \mathrm{l}4^2}. \tag{A8}$$

The joint movement, θ_3, is then computed using the equation:

$$\theta_3 = \pm(\alpha - \beta) \tag{A9}$$

where β is the angle between l4 and l5:

$$\beta = \mathrm{atan}\frac{l_5}{l_4}\,. \tag{A10}$$

Following the calculation of the θ_1 and θ_3 joint angles, one can also determine the θ_2 joint position. To do so, one uses the Matlab *solve* function, which will solve a two-equations–two-variables system, where sin θ_2 and cos θ_2 are the considered variables. To build the system of equations, sin θ_2 and cos θ_2 must be extracted from the following equation:

$$\mathrm{HC} = \mathrm{A}_1 \cdot \mathrm{A}_2 \cdot \mathrm{A}_3. \tag{A11}$$

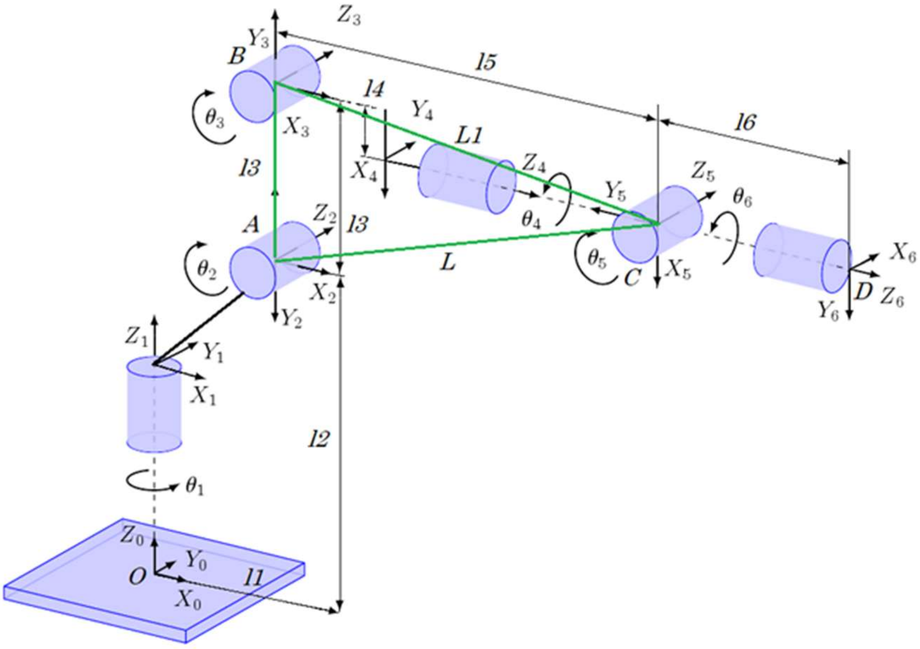


Figure A1. Schematic diagram used for inverse kinematic problem for KUKA KR 210 robot.

Through solving the system of equations, using the solve function, the following relationships can be written for $\sin(\theta_2)$ and $\cos(\theta_2)$:

$$\sin(\theta_2) = \frac{l2 - ZC}{a} + \frac{\left(b\cdot\left(l2\cdot l3 - l3\cdot ZC - c3\cdot l5\cdot\sqrt{c} + l4\cdot s3\cdot\sqrt{c} - e\right)\right)}{\left((c3\cdot l5 - l4\cdot s3)\cdot d\right)}, \tag{A12}$$

$$\cos(\theta_2) = \frac{-\left(l2\cdot l3 - l3\cdot ZC - c3\cdot l5\cdot\sqrt{c} + l4\cdot s3\cdot\sqrt{c} - e\right)}{d}. \tag{A13}$$

where:

$s3 = \sin(\theta_3)$, $c3 = \cos(\theta_3)$, $a = c3 \cdot l5 - l4 \cdot s3$, $b = c3 \cdot l4 - l3 + l5 \cdot s3$
$c = c3^2 \cdot l4^2 + c3^2 \cdot l5^2 - 2 \cdot c3 \cdot l3 \cdot l4 - l2^2 + 2 \cdot l2 \cdot ZC + l3^2 - 2 \cdot l3 \cdot l5 \cdot s3 + l4^2 \cdot s3^2 + l5^2 \cdot s3^2 - ZC^2$
$d = c3^2 \cdot l4^2 + c3^2 \cdot l5^2 - 2 \cdot c3 \cdot l3 \cdot l4 + l3^2 - 2 \cdot l3 \cdot l5 \cdot s3 + l4^2 \cdot s3^2 + l5^2 \cdot s3^2$
$e = c3 \cdot l2 \cdot l4 + c3 \cdot l4 \cdot ZC - l2 \cdot l5 \cdot s3 + l5 \cdot s3 \cdot ZC$

In the above relationship, l1, l2, l3, l4, l5, and l6 are the lengths of the robot kinematic elements, while XC, YC, and ZC are the coordinates of point C, per Figure 3.

Thus, the θ_2 angle can be calculated as:

$$\theta_2 = \operatorname{atan}\frac{\sin(\theta_2)}{\cos(\theta_2)}. \tag{A14}$$

In order to finish the development of the inverse kinematics model, the θ_4, θ_5, and θ_6 joint movements must be determined, based upon the combination of (11) and (20):

$$H = HC \cdot A_4 \cdot A_5 \cdot A_6. \tag{A15}$$

Using the following notation for the set of unknown variables A_4, A_5, and A_6:

$$Hint = A_4 \cdot A_5 \cdot A_6, \tag{A16}$$

one can write:

$$Hint = H \cdot HC^{-1}. \tag{A17}$$

It is notable here that the *Hint* matrix expresses only the orientation of the end-effector of the robot. Using the Euler angles to describe the orientation of a rigid body with respect to a fixed coordinate system results in the following relationships:

$$\begin{gathered}\theta_5 = \mathrm{acos}(\mathrm{H}(1,1)),\\ \theta_6 = \mathrm{atan}\frac{\mathrm{H}(1,2)}{\mathrm{H}(1,3)},\ \text{and}\\ \theta_4 = -\mathrm{atan}\ \frac{\mathrm{H}(2,1)}{\mathrm{H}(3,1)}\end{gathered} \tag{A18}$$

References

1. Vosniakos, G.C.; Matsas, E. Improving feasibility of robotic milling through robot placement optimization. *Robot. Comput. Integr. Manuf.* **2010**, *26*, 517–525. [CrossRef]
2. Cordes, M.; Hintze, W.; Altintas, Y. Chatter stability in robotic milling. *Robot. Comput. Integr. Manuf.* **2019**, *55*, 11–18. [CrossRef]
3. Nam Huynh, H.N.; Assadi, H.; Rivière-Lorphèvre, E.; Verlinden, O.; Ahmadi, K. Modelling the dynamics of industrial robots for milling operations. *Robot. Comput. Integr. Manuf.* **2020**, *61*, 101852. [CrossRef]
4. Sugita, S.; Itaya, T.; Takeuchi, Y. Development of robot teaching support devices to automate deburring and finishing works in casting. *Int. J. Adv. Manuf. Technol.* **2004**, *23*, 183–189. [CrossRef]
5. Villagrossi, E.; Pedrocchi, N.; Beschi, M.; Molinari Tosatti, L. A human mimicking control strategy for robotic deburring of hard materials. *Int. J. Comput. Integr. Manuf.* **2018**, *31*, 869–880. [CrossRef]
6. Chen, Y.; Dong, F. Robot machining: Recent development and future research issues. *Int. J. Adv. Manuf. Technol.* **2013**, *66*, 1489–1497. [CrossRef]
7. Ji, W.; Wang, L. Industrial robotic machining: A review. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 1239–1255. [CrossRef]
8. Verl, A.; Valente, A.; Melkote, S.; Brecher, C.; Ozturk, E.; Tunc, L.T. Robots in machining. *CIRP Ann. Manuf. Technol.* **2019**, *68*, 799–822. [CrossRef]
9. Borboni, A.; Lancini, M. Commanded motion optimization to reduce residual vibration. *J. Vib. Acoust.* **2015**, *137*, 031016. [CrossRef]
10. Tiboni, M.; Bussola, R.; Aggogeri, F.; Amici, C. Experimental and Model-Based Study of the Vibrations in the Load Cell Response of Automatic Weight Fillers. *Electronics* **2020**, *9*, 995. [CrossRef]
11. Lin, Y.; Zhao, H.; Ding, H. Posture optimization methodology of 6R industrial robots for machining using performance evaluation indexes. *Robot. Comput. Integr. Manuf.* **2017**, *48*, 59–72. [CrossRef]
12. Caro, S.; Garnier, S.; Furet, B.; Klimchik, A.; Pashkevich, A. Workpiece placement optimization for machining operations with industrial robots. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2014, Besançon, France, 8–11 July 2014; pp. 1716–1721. [CrossRef]
13. Höfener, M.; Schüppstuhl, T. A method for increasing the accuracy of "on-workpiece" machining with small industrial robots for composite repair. *Prod. Eng.* **2014**, *8*, 701–709. [CrossRef]
14. Kothe, S.M.; Stürmer SP, V.; Schmidt, H.C.; Boehlmann, C.; Wollnack, J.; Hintze, W. Accuracy Analysis and Error Source Identification for Optimization of Robot Based Machining Systems for Aerospace Production. *SAE Tech. Pap.* **2016**. [CrossRef]
15. Rosa, D.G.G.; Feiteira, J.F.S.; Lopes, A.M.; de Abreu, P.A.F. Analysis and implementation of a force control strategy for drilling operations with an industrial robot. *J. Braz. Soc. Mech. Sci. Eng.* **2017**, *39*, 4749–4756. [CrossRef]
16. Brunete, A.; Gambao, E.; Koskinen, J.; Heikkilä, T.; Kaldestad, K.B.; Tyapin, I.; Hovland, G.; Surdilovic, D.; Hernando, M.; Bottero, A.; et al. Hard material small-batch industrial machining robot. *Robot. Comput. Integr. Manuf.* **2018**, *54*, 185–199. [CrossRef]
17. Uhlmann, E.; Reinkober, S.; Hollerbach, T. Energy Efficient Usage of Industrial Robots for Machining Processes. *Procedia CIRP* **2016**, *48*, 206–211. [CrossRef]
18. Zivanovic, S.; Slavkovic, N.; Milutinovic, D. An approach for applying STEP-NC in robot machining. *Robot. Comput. Integr. Manuf.* **2018**, *49*, 361–373. [CrossRef]
19. Behera, A.K.; de Sousa, R.A.; Ingarao, G.; Oleksik, V. Single point incremental forming: An assessment of the progress and technology trends from 2005 to 2015. *J. Manuf. Process.* **2017**, *27*, 37–62. [CrossRef]
20. Fiorentino, A. Force-based failure criterion in incremental sheet forming. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 557–563. [CrossRef]
21. Meier, H.; Dewald, O.; Zhang, J. Development of a Robot-Based Sheet Metal Forming Process. *Steel Res. Int.* **2005**, *76*, 167–170. [CrossRef]
22. Meier, H.; Dewald, O.; Zhang, J. A New Robot-Based Sheet Metal Forming Process. *Adv. Mat. Res.* **2005**, *6–8*, 465–470. [CrossRef]
23. Meier, H.; Laurischkat, R.; Zhu, J. A Model Based Approach to Increase the Part Accuracy in Robot Based Incremental Sheet Metal Forming. *AIP Conf. Proc.* **2011**, *1315*, 1407. [CrossRef]
24. Störkle, D.D.; Möllensiep, D.; Thyssen, L.; Kuhlenkötter, B. Geometry-dependent parameterization of local support in robot-based incremental sheet forming. *Procedia Manuf.* **2018**, *15*, 1164–1169. [CrossRef]
25. Burghardt, A.; Szybicki, D.; Gierlak, P.; Kurc, K.; Pietruś, P.; Cygan, R. Programming of Industrial Robots Using Virtual Reality and Digital Twins. *Appl. Sci.* **2020**, *10*, 486. [CrossRef]
26. Malik, A.A.; Brem, A. Digital twins for collaborative robots: A case study in human-robot interaction. *Robot. Comput. Integr. Manuf.* **2021**, *68*, 102092. [CrossRef]

27. Li, M.; Wang, H. Enabling Improved Learning Capability of Industrial Robots with Knowledge Graph Towards Intelligent Digital Twins. In Proceedings of the IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hangzhou, China, 4–6 May 2022; pp. 599–604. [CrossRef]
28. Huynh, B.H.; Akhtar, H.; Sett, M.K. A Universal Methodology to Create Digital Twins for Serial and Parallel Manipulators. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 3104–3109. [CrossRef]
29. Magrini, E.; Flacco, F.; De Luca, A. Estimation of contact forces using a virtual force sensor. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2126–2133. [CrossRef]
30. Villagrossi, E.; Simoni, L.; Beschi, M.; Pedrocchi, N.; Marini, A.; Molinari Tosatti, L.; Visioli, A. A virtual force sensor for interaction tasks with conventional industrial robots. *Mechatronics* **2018**, *50*, 78–86. [CrossRef]