Elementary Welding Operations for Automatic Robot Programming

Sarivan, Ioan-Matei; Larsen, Jørgen Schiønning; Madsen, Ole; Wæhrens, Brian Vejrum

Published in:
Advances in Automotive Production Technology

DOI (link to publication from Publisher):
10.1007/978-3-031-27933-1_9

Creative Commons License
CC BY 4.0

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Sarivan, I-M., Larsen, J. S., Madsen, O., & Wæhrens, B. V. (2023). Elementary Welding Operations for Automatic Robot Programming. In N. Kiefl, F. Wulle, C. Ackermann, & D. Holder (Eds.), *Advances in Automotive Production Technology: Towards Software-Defined Manufacturing and Resilient Supply Chains* (Vol. 2, pp. 85-95). [9] Springer. https://doi.org/10.1007/978-3-031-27933-1_9

Elementary Welding Operations for Automatic Robot Programming

Ioan-Matei Sarivan[1](✉), Jørgen S. Larsen[2], Ole Madsen[1], and Brian V. Wæhrens[1]

[1] Department of Materials and Production at Aalborg University, Fibigerstræde 16, Aalborg 9220, Denmark
ioanms@mp.aau.dk

[2] PDMTechnology A/S, Niels Jernes Vej 10, Aalborg 9220, Denmark

Abstract. Highly-skilled engineers are needed to reprogram a welding-robot upon a new variant or product introduction. This paper addresses the complexity of the welding-robot programming process by proposing and demonstrating the theoretical concept of "elementary welding operation" (EWO). The EWO is a fundamental building block for robot-welding programs containing most of the data necessary to generate a program automatically. The EWO is a descriptive data structure for the welding process, comprehensible by humans and machines. A technical proof of concept is put together and tested. The results indicate that the programming of the welding robot can be automated as part of a one-piece information flow based on a product model. Further development is required to ensure the system can handle complex weldment geometries, multi-pass welding, and proper validation of the robot program upon deployment.

Keywords: robot programming · welding robot · elementary welding operation · digitalisation · data structure

1 Introduction

Welding is the most significant application for industrial robotics, with over 50% of the existing industrial manipulators being commissioned in welding tasks [1]. The usage of welding-robots comes with many advantages in terms of workers' health, increased quality assurance, shorter lead-time, and reduced production cost. The inflexible nature of welding-robots is seen, however, as a problem in engineering-to-order (ETO) and manufacturing-to-order (MTO) production environments, where high variation across produced items occurs. Each product is unique and requires the welding robot to be reprogrammed for every new incoming order. High-skilled robotics engineers are required to reprogram the welding-robots in order for a new product to be produced. The reprogramming process requires human, financial, and time resources which drive ETO and MTO companies away from the prospect of using welding-robots. [2]

N. Kiefl et al. (Eds.): SCAP 2022, ARENA2036, pp. 85–95, 2023.
https://doi.org/10.1007/978-3-031-27933-1_9

This paper proposes the theoretical concept of *elementary welding operation* (EWO). The EWO is a data structure meant to be used as a fundamental building block for the automatic programming of welding robots. The data contained inside an EWO is extracted automatically from a product instance. The data can be further processed to automatically program a welding-robot [3]. The digital integration of engineering and pre-production operations is obtained, and human input is reduced. The novelty of the implementation stands in having the geometry of the weldments programmed automatically based on unique product models. Experiments presented in this paper are limited to the validation of the weldment's geometry in an off-line programming (OLP) virtual environment using 6DOF welding-robots. The results show that EWO can potentially reduce the necessary time and cost involved in programming welding robots by automating this process.

The following meaning is attributed to the terms:

- Product Model: a parametric CAD entity that is configurable using a product configurator tool containing associated KBE (knowledge-based-engineering) and welding process data [3].
- Product Instance: a CAD entity obtained following the configuration of the Product Model using a product configurator by providing values for design parameters.

The paper is structured as follows: related work in the field of automatic robot programming and state-of-practice are presented in Sect. 2. Section 3 presents the overall structure of the elementary welding operation. Section 4 contains the technical description of a proof-of-concept implementation using the EWO concept. Results of experiments with the technical system are presented in Sect. 5. The system is bench-marked against the state-of-practice method. Conclusions and final considerations towards future development are mentioned in Sect. 6.

2 Related Work

Related work was found to be documented in the literature regarding systems that automatically extract welding geometry from CAD models and generate robot programs. The fundamental information used to program a welding robot is the geometry of the item to be welded, which is generally stored in a CAD model. The geometry of the item can be extracted from *STEP* (Standard for Exchange of Product Data) format CAD files by accessing the relevant data structures which define the item: vertices coordinates and vectors [4]. The STEP file format is generally used for CAD applications and is documented under the ISO10303 ISO standard.

In subtractive and additive manufacturing, parametric robot programming is a method widely used to handle large variations in the geometry of the item that is produced. Such methods are often found in systems like CNC machines [5] and 3D printers [6]. However, in the case of CNC and 3D printing machines, the orientation of the tool is often not an issue. Still, parametric programming allows for facile adjustment of existing robot programs, where the orientation of the tool is relevant [7].

In order to facilitate the geometry extraction of the product from CAD models, Lobov and Tran (2020) propose an object oriented automated method for product design. The method makes available the necessary functions to create and extract the geometry of the intersection edges of the solid bodies where it is possible for a weldment to be located [8,9]. Prescott et al. (2020) further developed the method by making it possible for geometry data for the weldment to be added on top of the CAD model at the identified intersection edges. This data is also known as KBE data (knowledge-based-engineering) [10] and can also contain process information about the weldment (e.g., filler-wire material). The data extracted from the CAD-KBE models is used to generate welding-robot trajectories based on the identified weld paths.

Alongside the geometry data of the item, the process data (e.g., torch travel angle, welding angle, torch travel speed) is needed to perform a weldment. Schmidt et al. propose a hierarchical data structure where process data like the welding current and gas flow are stored [11]. Mohammed et al. use the structure of the STEP CAD files to store process data like the orientation of the torch, welding speed, gas flow, and others [12].

Several methods and system architectures for programming of welding-robots based on CAD models were identified in the literature as related work, focusing on creating an end-to-end seamless pipeline from CAD to robot program. Some of these initiatives require human intervention to validate the final robot program [13], and others are completely automatic [14,15]. The EWO method presented in this paper brings as a contribution a data structure meant to facilitate the automatic transfer of geometry data and process data from a CAD system to an OLP (offline-programming) system.

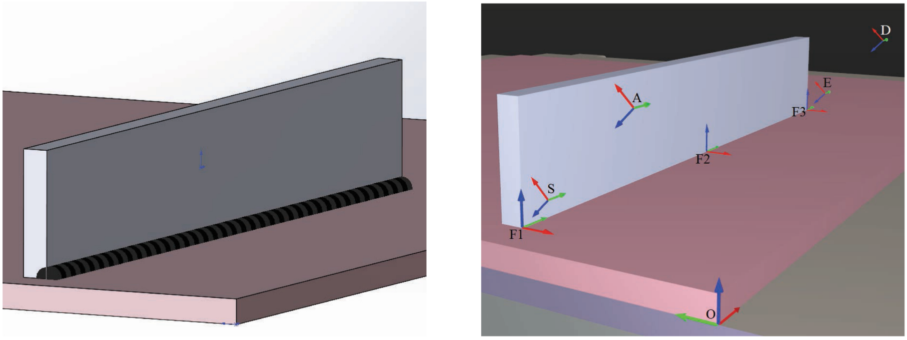
State-of-Practice

A state of practice study was done in collaboration with a partner ETO/MTO company where products vary order by order. The company is involved in the metal-fabrication industry, and they use welding-robots with the GMAW welding process. The robot-programming is handled by robotics engineers who use OLP software. The product instance is imported into the OLP software. Any overlaying process data is lost during the import procedures, including the locations of the weldments, and only the geometric CAD model is available in the OLP system. The robotics engineers must manually program the locations of the weldments and the process related data, using information available on the item's technical drafts, or using the product instance available in the source CAD system. For products where the geometry of the weldments is similar, and only the scale of the product differs, parametric robot programming is used, and older robot programs are edited to shorten the time needed to reprogram the robot. The programming process time varies between one to five days of work (considering a day of work has eight hours), depending on the complexity of the product.

3 Elementary Welding Operation

The proposed "elementary welding operation" EWO concept is designed to establish the data conduit between a product instance and the welding-robot program necessary for manufacturing that product. The EWO is part of a hierarchical data structure organizing the KBE data and the process data needed to program the welding-robot. The following assumptions are made regarding the welding-robot program:

1. The welding process goes through a number of quasi-stationary states
2. Each state is prismatic and can be represented using an EWO (Fig. 1)
3. A welding with complex geometry can be realised by combining a number of EWO (Fig. 2)

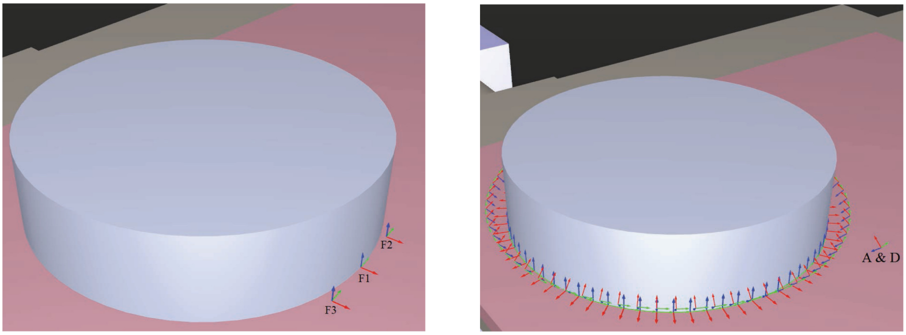


(a) Product instance with overlaid weldment KBE data

(b) Product instance imported in OLP tool

Fig. 1. Visual representations of a product instance (inside the SolidWorks CAD software) and a product instance imported in the RoboDK OLP software. The product is a fillet joint with a prismatic weldment.

The data structure containing EWO is represented in Fig. 3. The hierarchical root is named "Weldments" and is composed of at least one "Weldment." In the context of the data structure, a "Weldment" results from a continuous motion of the torch along the welding seam, together with the approach and departure motions before and after starting to weld. The approach and the departure motions are defined using the *approach pose* and a *departure pose* as seen in Fig. 3. A "Weldment" contains at least one Elementary Welding Operation. The EWO is always a straight line and represents the indivisible element that composes the process of performing a robotic weldment.



(a) Auxiliary frames for EWO contruction on a curved weldment seam

(b) Frames are added for each tessellation point of the curve where an EWO is located

Fig. 2. Visual representation of round shaped object. For each tessellation point of the curve, a coordinate frame is used to represent an EWO.

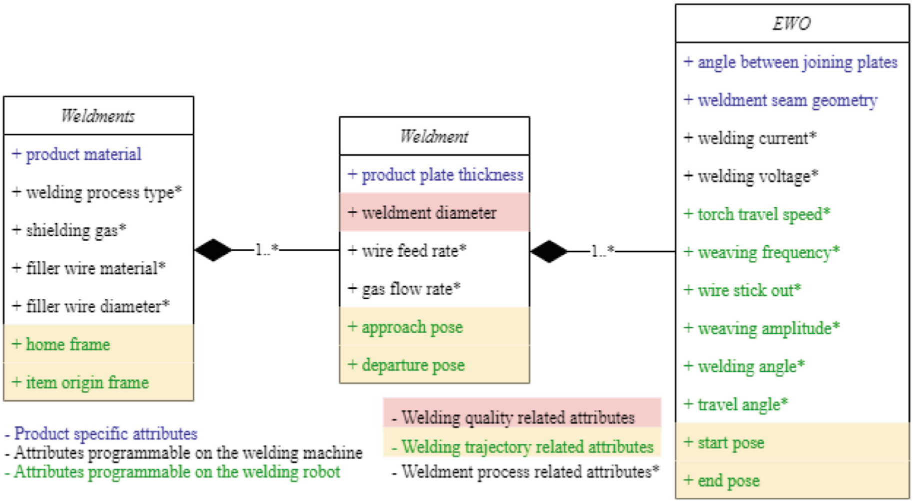


Fig. 3. UML diagram containing the data structure of welding program composed of EWOs. The attributes at each hierarchical level are classified using several criteria.

The attributes contained by each data class element of an EWO program can be observed in Fig. 3. The welding system considered in this paper is composed of a welding robot and a welding machine, each having their own controller. The attributes are organised at the level of the data structure as follows:

- "Weldments": attributes based on the material of the product which stay unchanged during the whole welding process.
- "Weldment": attributes that can vary from weldment to weldment based on the thickness of the plate, but kept constant while welding.

- "EWO": attributes that are variable while the welding torch is active and depend on the angle between the plates and the overall weldment seam geometry of the product.

The attributes are further classified into three categories. "Product," "Welding Machine," and "Welding Robot." The "Product" attributes are the properties of the item to be welded and are marked with blue text. These can be extracted from the CAD model and the overlaid KBE data. The attributes of the "Welding Machine" (black) and the "Welding Robot" (blue) are programmed on each machine based on the product's properties.

As observed in Fig. 3, the attributes related to the geometry of the robot trajectory are highlighter with yellow. The geometry attributes do not influence the quality of the weldment, only the path that is being followed by the welding torch. The geometry attributes are determined based on the geometry of the item that is extracted from the product instance. Process-related attributes do influence the quality of the weldment. A weldment quality parameter considered in this paper is the *weldment diameter*.

The main geometric characteristic of the EWO is a straight line. Therefore, when a weldment is composed of only one straight line, it contains one single EWO with a *start* and an *end pose*. When the weldment has a curved geometry, the "weldment" data structure is composed of n EWOs, where n is the number of tessellation points of the curve as observed in Subfigure 2b. For the intermediary EWOs contained in a "Weldment" data structure, only the start poses are considered for the welding-robot program. The values for the rest of the variables are contained in look-up tables or parametric rules, which are based on the product's material and plate thickness.

To configure the EWO *start* and *end pose*, the example in Fig. 1 is considered. The *item origin frame* represents the frame in which the geometry of the item is defined, marked with O in Subfigure 1a. The origin frame is imported from the product instance. The *approach pose* (marked with A in Subfigure 1a) and the *departure pose* (marked with D in Subfigure 1a) represent the approach/departure positions and orientations (poses) for the torch before and after engaging in a weldment. The *start pose* (marked with S) and *end pose* (marked with E) represent the start and end poses of the torch at the start and the end of the EWO.

To determine the approach/departure and the start/end poses for the torch, Horn's absolute orientation method is used [16]. The method makes it possible to determine the transformation matrix from one reference frame to another by having the coordinates of three points as input defined in both frames. First, reference frames are created along the weldment (marked with $F1$, $F2$ and $F3$ in Subfigure 1a). The positions of the frames are set using the coordinates of the tessellation points that compose the weldment. The coordinates of the tessellation points are expressed in relation to frame O. For a straight weldment, three frames are considered, at the start $F1$, at the middle $F2$, and at the end $F3$.

The frames have the Y axis along the weldment seam and the X axis along the horizontal surface. The linear transformation from frame O to frame $F1$

is determined by considering the matrices A and B. Matrix A contains the coordinates of frames $F1$, $F2$ and $F3$ represented in relation to frame O. The matrix B contains the coordinates of the frames in relation to frame $F1$. The matrices can be observed in Eq. 1. The x and z coordinates of $F2$ in relation to $F1$ are set to be 0, while the y coordinate is the distance between the two frames. This specifies that the y axis of the frame is along the weldment seam. The same procedure applies for $F3$.

$$A_O = \begin{bmatrix} x_{F1} & x_{F2} & x_{F3} \\ y_{F1} & y_{F2} & y_{F3} \\ z_{F1} & z_{F2} & z_{F3} \end{bmatrix} \quad B_{F1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & dist(F1, F2) & dist(F1, F3) \\ 0 & 0 & 0 \end{bmatrix} \tag{1}$$

In the case of a weldment along a curved edge, the Y axis is tangent with the curved edge and makes contact with at least one tessellation point of the weldment. As observed in Subfigure 2a, frames $F2$ and $F3$ are placed along the curve's tangent, $F2$ on the positive side and $F3$ on the negative side, therefore the $dist(F1, F3)$ term of Matrix B in Eq. 1 will have the sign changed. The procedure is repeated for each tessellation point of the weldment's curve, as can be observed in Subfigure 2b.

With the input A and B matrices, the Horn's method yields the transformation matrix "T" from the "O" frame to the "F1" frame of each tessellation point. Therefore, the whole geometry of the weldment can be described mathematically in relation to "O". In order to express the start/end poses and the approach/departure poses the welding parameters can be used. The wire stick out s, the welding angle α, the travel angle β, and the approach distance d are considered. The matrix transformation from the frame "F" of the tessellation point to the start pose is displayed in Eq. 2. The same transformation can be used to determine the end pose and the approach/departure pose by replacing the s term with d. The results of the transformation applied for start/end poses, and approach/departure poses can be observed in Fig. 1 for a straight fillet weldment and Fig. 2 for a curve-shaped weldment.

$${}^{F1}_{S}T = \begin{bmatrix} -cos(\alpha) & -sin(\alpha)sin(\beta) & -sin(\alpha)cos(\beta) & s * sin(\alpha)cos(\beta) \\ 0 & cos(\beta) & -sin(\beta) & s * sin(\beta) \\ sin(\alpha) & -cos(\alpha) * sin(\beta) & -cos(\alpha) * cos(\beta) & s * cos(\alpha)cos(\beta) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

4 Technical Implementation

To implement the EWO theoretical concept, the XML file format was chosen to store the data represented in Fig. 3. The SolidWorks CAD software was selected to create product instances based on product models that store KBE data and process data [3]. For robot programming, the RoboDK OLP tool was selected as welding programs can be created for a variety of welding-robot vendors.

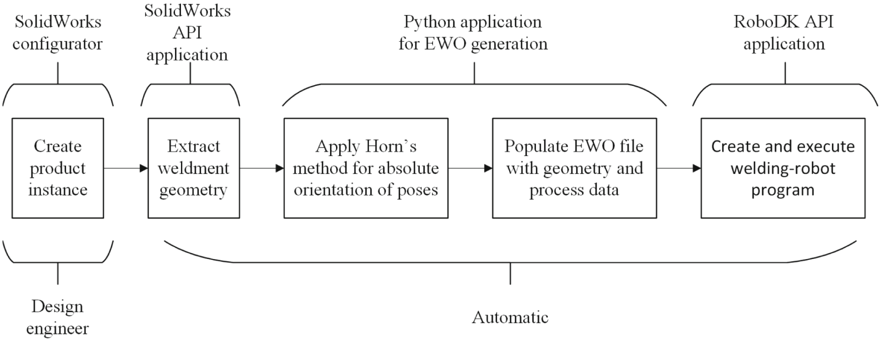

SolidWorks configurator
SolidWorks API application
Python application for EWO generation
RoboDK API application
Create product instance
Extract weldment geometry
Apply Horn's method for absolute orientation of poses
Populate EWO file with geometry and process data
Create and execute welding-robot program
Design engineer
Automatic

Fig. 4. Implementation pipeline using elementary welding operations.

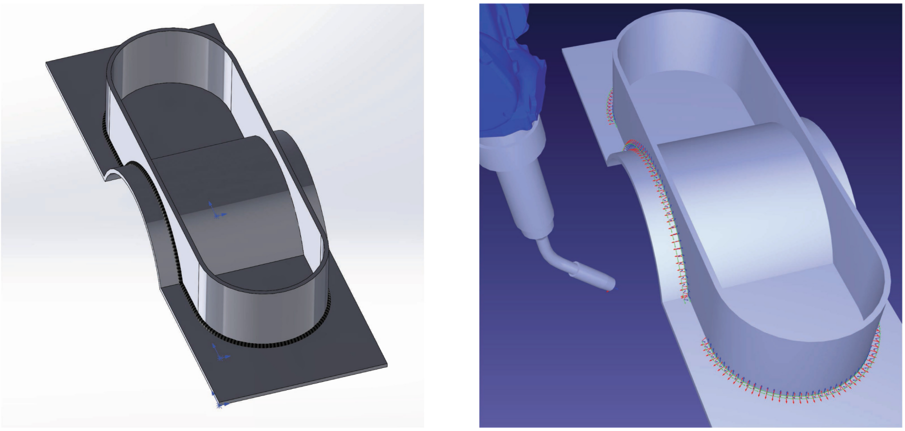
The EWO concept enables seamless data transfer between the two software environments by using C#-Python application specially developed for this purpose. The implementation is summarised in Fig. 4.

With the EWO data structure in place, the RoboDK API is used to create TCP targets. A collision check is executed for each target in order to create a collision-free weldment trajectory for the welding torch. The process is automatic and yields the robot program that can be either executed in real-time or transferred to the robot's controller for execution.

5 Experiments and Results

Experiments were conducted using the presented technical implementation to test the viability of the EWO theoretical concept. The testing criteria used to evaluate the results are the number of engineers needed to operate the required software and the time required to generate a welding program for a welding robot. The implementation was compared against the state-of-practice method at the partner company. By using the implementation pipeline from Fig. 4, weldment geometry data is automatically extracted from the product instance (Fig. 5, Subfigure 5a) which is turned into TCP goal poses for the welding torch (Subfigure 5b).

The EWO method allows for the automatic import of most of the geometry and process data (welding angle and travel angle) needed to generate the welding-robot program without further intervention from engineers to fine-tune the positioning of the welding torch. Once the product instance is generated, the design engineer only needs to initiate the Python software described in the previous section, which will automatically generate the welding-robot program based on the EWO data. The time required to generate the program can take up to a few minutes, depending on the complexity of the product and the performance of the computer running the software.



(a) Product instance generated using a product configurator [3]. The overlaid KBE data can be observed.

(b) Visual representation of the product inside the OLP system with EWO data.

Fig. 5. EWO welding program is obtained inside the RoboDK OLP system using the welding geometry and process data imported from the product instance.

6 Discussion and Conclusion

In this paper, a novel method for structuring geometry and process data for automatic programming of welding robots was presented. The initiative is aligned with previous efforts for creating digital tools that reduce the trade-off between automation and flexibility in MTO, and ETO environments [19]. Experiments show that the EWO theoretical concept can support technical implementations that enable the reduction of time needed to program welding-robots and the amount of skilled human resources required.

Further tests are required to determine how the complexity of the product affects the performance of the implementation, and how advanced geometries can be handled (e.g. multi-pass welding). The EWO shows promising potential as a platform for welding process-data. It is, therefore relevant to investigate further if the data structure presented is sufficient to support the parameters required to perform other welding procedures besides the GMAW procedure considered in the experiments conducted in this paper. Transferability to other applications where geometric discretization is possible (e.g., 3D printing) is also relevant to be investigated, especially where the orientation of the tool is relevant.

Acknowledgements. This work is financed and supported by Manufacturing Academy of Denmark (MADE).

References

1. Sprovieri, J.: New technology for robotic welding. Assembly magazine (2016). https://www.assemblymag.com/articles/93555-new-technology-for-robotic-welding. Accessed 9 May 2022
2. The Benefits of Robotic Welding in Small Business. Robotics and Automation News (2020). https://roboticsandautomationnews.com/2020/02/20/the-benefits-of-robotic-welding-in-small-businesses/30181/. Accessed 9 May 2022
3. Sarivan, IM., Madsen, O., Waehrens, B.V.: Towards automatic welding-robot programming based on product model. In: Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems. CARV MCPC 2021 2021. Lecture Notes in Mechanical Engineering, pp. 174–181. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-90700-6_19
4. Sivakumar, S., Dhanalakshmi, V.: An approach towards the integration of CAD/CAM/CAI through STEP file using feature extraction for cylindrical parts. Int. J. Comput. Integr. Manuf. **26**(6), 561–570l (2013)
5. Sigrid, B.C., Johannes, B.: Parametric robot control: integrated CAD/CAM for architectural design. In: ACADIA Proceedings (2011)
6. Kuipers, T., et al.: A framework for adaptive width control of dense contour-parallel toolpaths in fused deposition modeling. Comput. Aided Des. **128**, 102907 (2020)
7. Feucht, T., Lange, J., Erven, M., et al.: Additive manufacturing by means of parametric robot programming. Constr. Robot. **4**, 31–48 (2020)
8. Lobov, A., Tran, T.A.: Object-oriented approach to product design using extended NX open API. Procedia Manuf. **51**, 1014–1020 (2020)
9. Tran, T.A., Lobov, A., Kaasa, T.H., et al.: CAD integrated automatic recognition of weld paths. Int. J. Adv. Manuf. Technol. **115**, 2145–2159 (2021)
10. Prescott, S.A.O., Tran, T.A., Lobov, A.: Automatic weld path definition in CAD. Procedia Manuf. **51**, 478–484 (2020)
11. Schmidt, A., et al.: A generic data structure for the specific domain of robotic arc welding. Procedia CIRP **72**, 322–327 (2018)
12. Mohammed, S.K., Arbo, M.H., Tingelstad, L.: Constraint Identification from STEP AP242 files for automated robotic welding. In: 2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), pp. 277–282 (2021)
13. Bedaka, A.K., Lin, C.-Y.: CAD-based offline programming platform for welding applications using 6-DOF and 2-DOF robots. In: 2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS), pp. 1–4 (2020)
14. Jiang, Q.X., Wang, S.H., Wei, X.Q., Hou, R.S., Chang, B.H., Du, D.: Model-driven online correction welding robot automatic programming system. In: IOP Conference Series. Materials Science and Engineering, vol. 504, no. 1 (2019)
15. Shen, W., et al.: A welding task data model for intelligent process planning of robotic welding. Rob. Comput. Integr. Manuf. **64**, 101934 (2020). https://doi.org/10.1016/j.rcim.2020.101934
16. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Am. A **4**, 629–642 (1987). https://doi.org/10.1364/JOSAA.4.000629
17. Matt, J.: Absolute Orientation - Horn's method (2022). https://www.mathworks.com/matlabcentral/fileexchange/26186-absolute-orientation-horn-s-method. MATLAB Central File Exchange. Accessed 16 May 2022
18. https://github.com/MateiSarivan/EWO

19. Bejlegaard, M., Sarivan, I.-M., Waehrens, B.V.: The influence of digital technologies on supply chain coordination strategies. J. Global Oper. Strateg. Sourcing **14**(4), 636–658 (2021)