Available online at www.sciencedirect.com
ScienceDirect
Procedia CIRP 60 (2017) 122 – 127
www.elsevier.com/locate/procedia

27th CIRP Design 2017

Manufacturing task description for robotic welding and automatic feature recognition on product CAD models

Alexander Kuss[a,*], Thomas Dietz[a], Konstantin Ksensow[b], Alexander Verl[c]

[a] *Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Nobelstrasse 12, 70569 Stuttgart, Germany*
[b] *University of Applied Sciences, Schellingstr. 24, 70174 Stuttgart, Germany*
[c] *Institute for Control Engineering of Machine Tools and Manufacturing Units ISW, University of Stuttgart, Seidenstrasse 36, 70174 Stuttgart, Germany*

* Corresponding author. Tel.: +49-711-979-1297 ; fax: +49-711-979-1008. *E-mail address:* alexander.kuss@ipa.fraunhofer.de

Abstract

In this paper, a novel approach for an ontological representation of continuous-path manufacturing tasks is presented on the example of robotic fillet welding. The representation specifies manufacturing information and product related geometric constraints relevant for automatic program planning in robotic arc welding. The geometric constraints are transformed into simple continuous curve primitives such as lines, circular arcs and B-spline curves supporting the easy integration in robot program planning systems decoupled from a complex CAD environment. It also represents information on welding gaps along the joint geometry. Moreover, a method for automatic feature recognition based on rule-based pattern recognition is presented to automatically retrieve manufacturing information from the CAD model of the product and generate the respective manufacturing task specification. The method is evaluated in several test scenarios for welding product assemblies indicating the usability and effectiveness for automatic program planning in robotic welding.

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference
Keywords: Robot; CAD model; Welding

1. Introduction

Fast and intuitive programming of robotic manufacturing systems is a main challenge to enable cost-effective automation of complex production processes in small and medium-sized enterprises (SMEs) with changing product variants and small lot sizes [1]. Research in this field falls into the two categories of online and offline programming [2]. In online programming, the user typically moves the robot to desired positions using a teach pendant. Even though online programming is simple and widely used in industry, it can be time-consuming and leads to downtimes of the robot cell. Moreover, the quality of the robot process depends on the skills of the human operator. This knowledge is not modeled and cannot be reused for new product variants.

To overcome these drawbacks, offline programming (OLP) can be applied based on a 3D simulation of the robot cell. Here, the 3D CAD model of the workpiece is used to plan the manufacturing task and the required robot process. The planning of the manufacturing task is typically based on the determination of manufacturing features describing product related characteristics of the manufacturing process. Research in the field of manufacturing features focusses mainly on machining [3–5] and there is only few research on welding. A general description of manufacturing features for welding without implementation details is given in the guideline VDI 2218 [6]. A task description for remote-laser-welding based on simple seam elements consisting of two or more points is proposed by [7]. Another approach also describes weld features with discrete contour points and additional vectors for seam direction and seam normal as well as information about the welding process [8]. A weld seam description based on contour segments consisting of discrete points including normal vectors for joint faces description is proposed by [9]. However, in all these approaches, there is a loss of accuracy due to discretization of contour elements. Moreover, gaps at the joint geometry are not considered. This is important in robotic welding of manually prepared parts, where gaps at the joint geometry have to be considered in the process planning [10,11]. A weld feature description based on continuous linear, circular or curved contour segments and additional geometric joint and weld parameters is proposed by [12]. However, there is also no possibility of representing gap information.

Besides the description of the manufacturing task through welding features, it is also important how to determine these features in the OLP system. There exist three main approaches: (1) Design-by-feature, (2) feature mapping and (3) feature recognition. The Design-by-feature approach is based on in-

Peer-review under responsibility of the scientific committee of the 27th CIRP Design Conference
doi:10.1016/j.procir.2017.01.045

cluding manufacturing information in the design phase of the product. The application of this approach on programming of robotic welding processes has been shown in [13]. However, it only shifts programming efforts to the design process and lacks standardized interfaces to the OLP system.

In the feature mapping approach, design features are automatically converted to manufacturing features by mapping techniques [8]. This however still requires a specific feature modeling in the design process.

The feature recognition approach can be divided into interactive and automated feature recognition. Interactive feature recognition is based on user input to interactively select geometric entities on the CAD model to be grouped into a manufacturing feature [14,15]. This approach is the basis of numerous robot OLP environments. However, the procedure of manually defining manufacturing features is time-consuming and error-prone. Therefore, automated feature recognition aims at automatic detection of manufacturing features on the CAD model of the product. Here, one approach is to use rule-based pattern recognition with logic if-then rules, first introduced by [16] for machining features. This approach has also been applied to robotic welding to automatically detect weld joints [17] on the CAD model of a workpiece. Here, candidate weld joints are identified as edge features on the workpiece geometry. However, no detailed weld feature specification and no general approach for describing the derived welding task are provided.

In this paper, we introduce a general ontological manufacturing task description based on continuous welding features. It includes geometric constraints from the product joint geometry as well as information on the seam geometry and especially accounts for representing information on welding gaps. Moreover, a method for automatic feature recognition based on rule-based pattern recognition is presented to automatically detect weld features on CAD models of welding products and generate the respective manufacturing task description.

This paper is structured as follows: In Section 2, a manufacturing task description for automated welding is proposed. Section 3 describes the approach for automatic detection of weld tasks on product CAD models based on rule-based pattern recognition. In section 4, the proposed approach is validated for different welding product geometries. Finally, conclusions are presented in Section 5.

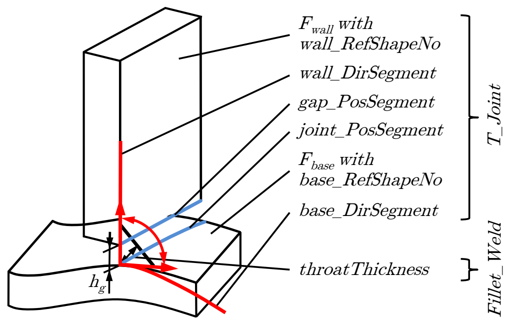

Fig. 1: Weld feature description for T-joint and fillet weld seam.

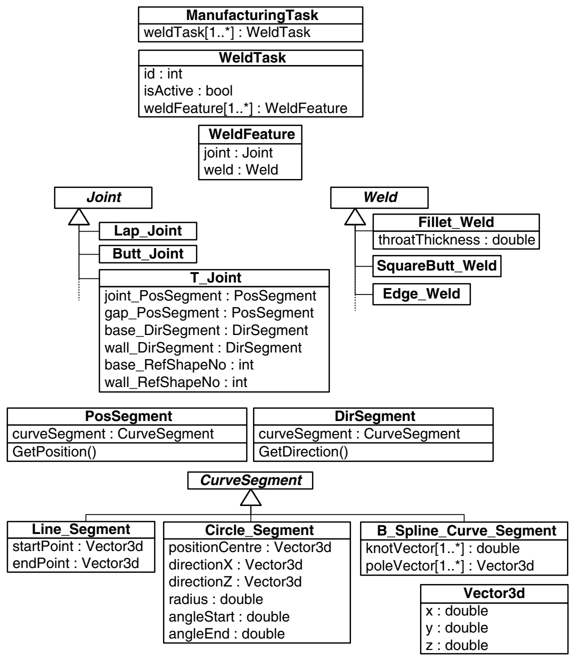

Fig. 2: Class diagram of manufacturing task description.

2. Manufacturing task description for robotic welding

The manufacturing task description is based on welding features. Unlike in machining where a feature is composed of different elements of the CAD model, the geometric elements of a welding feature are typically not modeled in the 3D CAD model during the design phase. Instead, information about welds is given in the 2D drawing of the product with references to the product geometry and is interpreted by the human process experts in the production process. Hence, a machine readable description of a weld feature represents additional information to an incomplete CAD model. It has to define constraints for the robot motion along the workpiece. The transformation of the motion constraints to a tool frame position of the robot welding gun can be performed by a subsequent process planning with a robot offline planning system but is not focus of this work.

A weld feature includes information on the joint geometry and the seam geometry . In the following, the welding feature is exemplarily described for a T-joint geometry with a fillet weld seam, like shown in Fig. 1. However, it can also be adapted to other joint and seam types. In the current framework, the STEP Application Protocol AP203 is used. However, future implementations will also allow other STEP protocols to be used with the proposed framework. The T-joint can be derived by the intersection of a shape of a base element *base_RefShape* and a shape of the wall element *wall_RefShape* of the B-Rep CAD model. The intersection contour of these shapes is called *joint_PosSegment*. For a T-joint, the angle of the intersecting faces should be $\alpha \approx 90\,\text{deg}$. Also, there might be a gap at the joint geometry. Therefore, the contour *gap_PosSegment* is derived from the *wall_RefShape*. The gap height h_g is consequently defined as the distance between *joint_PosSegment* and *gap_PosSegment*. Moreover, the seam geometry for a fillet seam can be defined by a *throatThickness* parameter.

The ontology of the manufacturing task is presented as an UML class diagram, shown in Fig. 2. The manufacturing class consists of a list of *WeldTask* objects. The *WeldTask* class consists of an identifier object *id*, a task state object *isActive* and a list of one or more *weldFeature* elements. The *WeldFeature* class is composed of a *joint* and a *weld* object. There exist several joint types, like T-joint, lap joint or butt joint, according to [12]. Therefore, inheritance of an abstract *Joint* class is used.

The focus of this work is on T-joints as shown in Fig. 1. The *T_Joint* class is composed of the objects *joint_PosSegment* and *gap_PosSegment* of *PosSegment* type, *base_DirSegment* and *wall_DirSegment* of *DirSegment* type as well as *base_RefShapeNo* and *wall_RefShapeNo* of *Integer* type. The classes *PosSegment* and *DirSegment* both consist of a *curveSegment* object. However, the *PosSegment* includes the function *GetPosition(t)*, $t \in [t_0, t_K]$ which returns a vector to a point on the respective curve segment at parameter t. The *DirSegment* has the function *GetDirection(t)* returning a vector of the derivative at t of its respective curve segment. The abstract class *CurveSegment* represents the different types of continuous curve segments and is the base class of *Line_Segment*, *Circle_Segment* or *B_Spline_Curve_Segment*. These segments are similar to the continuous curve segments used in the STEP format. This means, that the joint geometry can be modeled with its mathematically exact contour and does not have to be represented by discrete coordinate systems, as it is the case in most robot OLP systems.

Moreover, there exist several weld seam types according to ISO 2553 standard [18]. The abstract *Weld* class is the base class of the several weld type classes, like *FilletWeld*, *SquareButt_Weld*, *Edge_Weld* etc. In this paper, we focus on fillet weld seams. The *Fillet_Weld* class consists of a *throatThickness* object of type *double* describing the design throat thickness of the fillet weld seam.

The manufacturing task description can be represented in Extensible Markup Language (XML) as shown in Listing 1. The XML document can then be used as an addition to the original CAD file of the welding product.

```xml
<?xml version="1.0" ?>
<manufacturingTask>
  <weldTask id = "0" isActive = "true">
    <weldFeature>
      <t_Joint>
        <joint_PosSegment> ... </joint_PosSegment>
        <gap_PosSegment> ... </gap_PosSegment>
        <base_DirSegment> ... </base_DirSegment>
        <wall_DirSegment> ... </wall_DirSegment>
        <base_RefShapeNo> ... </base_RefShapeNo>
        <wall_RefShapeNo> ... </wall_RefShapeNo>
      </t_joint>
      <fillet_Weld>
        <throatThickness> ... </throatThickness>
      </fillet_Weld>
    </weldFeature>
    <weldFeature> ... </weldFeature>
  </weldTask>
  <weldTask id = "1" isActive = "false"> ... </weldTask>
  ...
</manufacturingTask>
```

Listing 1: Manufacturing task description in XML format

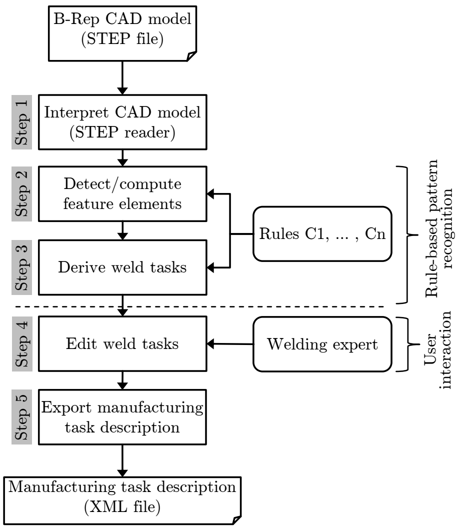

Fig. 3: Detection of manufacturing tasks on CAD model

In addition, also tolerance information could be included in the proposed framework. A possible example is information on dimensional tolerances of the throat thickness of the fillet weld seam that could be represented in the *throatThickness* object. This tolerance information could then be used i.e. as an input for subsequent quality inspection processes of weld seams on a real workpiece geometry. However, the above mentioned representation of tolerance information in the proposed framework is beyond the scope of this paper and will be subject to future work.

3. Detection of manufacturing tasks on product models

The welding tasks have to be detected on the CAD model of the workpiece. Therefore, an approach based on rule-based pattern recognition is proposed, as shown in Fig. 3. Here, a product geometry model is used based on a B-Rep CAD model in the STEP format. In step 1, a STEP reader is used to load the CAD model and represent its information in an appropriate model structure for further geometric processing. In step 2, welding features elements are detected or computed based on the workpiece geometry. Therefore, automatic feature recognition based on logic rules is used. The detected features are then used as a basis to derive different welding tasks in step 3. Here, again specific rules are used. In step 4, the welding tasks can be edited by user interaction. A welding expert can e.g. select which of the detected candidate weld tasks should be active or adapt the value of the throat thickness of the fillet weld. Finally, in step 5, the weld tasks can be exported to a manufacturing task description in XML format. In the following, the rule-based pattern recognition (step 2 and step 3) is described in more detail.

3.1. Recognition of weld features

The recognition of weld features is based on logic rules, as proposed by [16]. If conditions C1, ..., Cn are met, then a weld feature is detected. Unlike in recognition of machining features, where all geometric feature elements are already modeled in the product CAD model, in welding typically not all feature elements are modeled in the design phase and have to be computed. Hence, the rules C1, ..., Cn describe a process how weld feature elements are computed as well as conditions that have to be met for geometric elements to be recognized as valid features elements. The main rules for weld feature detection, also illustrated in Fig. 4 (a), are:

- C1: A face F_i is extended tangentially on its outer curve elements with an extension length of h_e; if the extended face $F_{i,e}$ has an intersection with any other face F_j, the intersection curve is a *joint_PosSegment*.
- C2: $F_{i,e}$ is the F_{Wall} face and the intersected face the F_{Base} face; the respective hash tag numbers of F_i and F_{Base} in the STEP file are stored in *wall_RefShapeNo* and *base_RefShapeNo*.
- C3: The curve segment on the extended curve of F_{Wall} is represented as a parametric curve in the *gap_PosSegment*(t), $t \in [t_0, t_K]$.
- C4: The vector perpendicular to the the parametric curve *joint_PosSegment*(t) and tangential to F_{Base} is a base vector $\vec{b}(t)$; the integral curve of $\vec{b}(t)$ along the *joint_PosSegment*(t) is stored in *base_DirSegment*(t).
- C5: The vector perpendicular to the *joint_PosSegment*(t) and tangential to F_{Wall} is a wall vector $\vec{w}(t)$; the integral curve of $\vec{w}(t)$ along the *joint_PosSegment*(t) is stored in *wall_DirSegment*(t).
- C6: The length $h_g(t)$ of any vector perpendicular to *joint_PosSegment* and between a point on the *joint_PosSegment* and the corresponding *gap_PosSegment* should be below a maximum gap height $h_{g,max} \in \mathbb{R}^+$.
- C7: The angle $\alpha(t)$ between any corresponding vectors $\vec{b}(t)$ and $\vec{w}(t)$ along the *joint_PosSegment*(t) has to conform $90° - \alpha_{min} < \alpha(t) < 90° + \alpha_{max}$, $(\alpha_{min}, \alpha_{max}) \in \mathbb{R}^+$.
- C8: Each *t_Joint* object is assigned to a *weldFeature* object, including a *fillet_Weld* object with a default *throatThickness* value $a_{def} \in \mathbb{R}^+$.

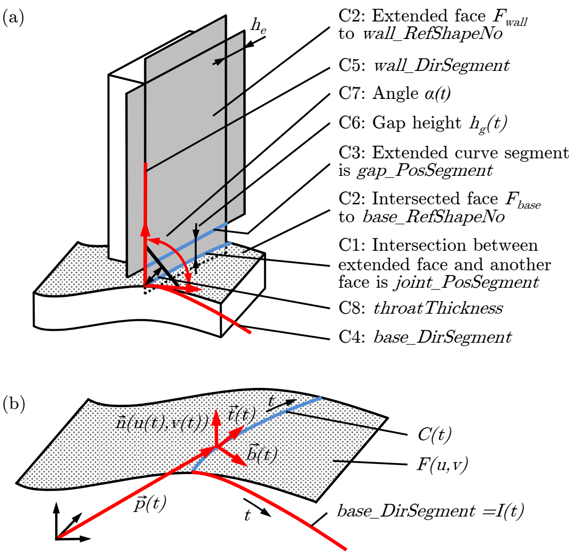

Fig. 4: (a) Application of rules for weld feature recognition and (b) computation of *base_DirSegment*

Especially rule C4 (and C5 accordingly) is discussed in more detail as it describes the process to generate a new geometric element, as shown in Fig. 4 (b), that is not modeled in the product CAD model. For a point $\vec{p}(t)$ that is part of the surface $F(u, v) = F_{base}(u, v)$, $(u, v) \in U \subset \mathbb{R}^2$ and the curve segment $C(t)$ = *joint_PosSegment(t)*, $t \in I \subset \mathbb{R}$, a vector $\vec{b}(t)$ has to be found that is perpendicular to $C(t)$ and tangential to $F(u, v)$. To define a curve on the surface, $C(t)$ can be written as:

$$C(t) = F(u(t), v(t)) \tag{1}$$

The base vector $\vec{b}(t)$ can then be calculated as the cross product of the curve tangent $\vec{t}(t) = \dot{C}(t)$ and the surface normal $\vec{n}(u(t), v(t))$ as:

$$\begin{aligned}\vec{b}(t) &= \vec{t}(t) \times \vec{n}(u(t), v(t)) \\ &= \dot{C}(t) \times F_u(u(t), v(t)) \times F_v(u(t), v(t)) \\ &= \frac{d}{dt}F(u(t), v(t)) \times \frac{\partial}{\partial u}F(u(t), v(t)) \times \frac{\partial}{\partial v}F(u(t), v(t))\end{aligned} \tag{2}$$

The different base vectors $\vec{b}(t)$ along the *joint_PosSegment(t)* form a vector field and can be represented by an integral curve $I(t)$ with the same parameterization of t which is stored based on a continuous curve element in the *base_DirSegment* (see section 2). A specific base vector $\vec{b}(t_s)$ can then be retrieved by the derivative of the *base_DirSegment* at $t_s \in \mathbb{R}$. The same procedure is applied accordingly for computing the *wall_DirSegment*.

The application of the rule-based feature detection is exemplified on the welding assembly in Fig. 4 (a). Rule C1 results in the extension of all faces of the welding assembly tangentially on their outer curve elements. In Fig. 4 (a) one extended planar face is visualized with the extensions on the linear outer curves. In this case, the intersection of the extended face with another B-spline face of the welding assembly results in a *B_Spline_Curve_Segment* that is stored in the indicated *joint_PosSegment*. Using rule C2, the extended face is identified as F_{Wall} and the intersected face as F_{Base} and their respective hash tag numbers of the STEP file are stored in the *wall_RefShapeNo* and *base_RefShapeNo* objects. After this, rule C3 is applied resulting in the indicated linear curve of F_{Wall} that is stored as *Line_Segment* in a *gap_PosSegment* object. In accordance with rule C4, a *B_Spline_Curve_Segment* is identified as *base_DirSegment* as shown in Fig. 4 (a). Rule C5 is applied accordingly resulting in a *Line_Segment* that is stored in the *wall_DirSegment* object. Thereafter, in rule C6, gap heights along the *joint_PosSegment* can be calculated to ensure that all gap heights are below a maximum value $h_{g,max}$. Accordingly, in rule C7, angles are tested to not exceed specific limits α_{min} and α_{max}. If the detected feature elements satisfy rules C6 and C7, they are stored in a *t_Joint* object. Finally, by application

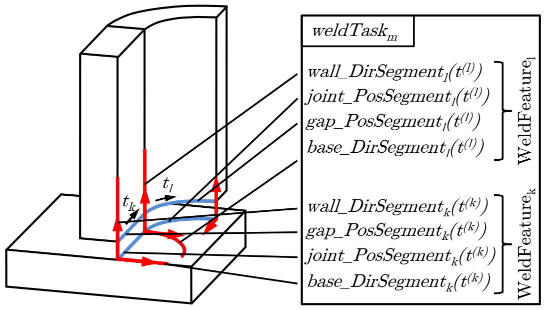

Fig. 5: Recognition of weld tasks

of rule C8, a *weldFeature* object is created that includes the detected *t_Joint* object and a *FilletWeld* object. The latter includes a *throatThickness* object representing a value for the design throat thickness of the weld seam to be produced.

3.2. Recognition of weld tasks

As described in section 2, a weld task is composed of an identifier *id*, a boolean state variable *isActive* and a list of one or more *weldFeature* elements. To decide how many weld features are combined to one weld task further rules are applied. The basis of these rules is the C^0 and C^1 continuity between curve elements of the weld feature. Fig. 5 exemplarily shows the weld task recognition for $n = 2$ $weldFeature$ elements on a CAD model. The main rules for weld task recognition are:

- C9: For every $weldFeature_k$, $k \in I$, $I = 1, ..., n$, $n \in \mathbb{N}$, check if the $joint\_PosSegment_k(t^{(k)})$ has C^0 and C^1 continuity with any other $joint\_PosSegment_i(t^{(i)})$ of a $weldFeature_i$, with $i \in I, (i \neq k)$.
- C10: If C9 applies for a $weldFeature_k$ and a $weldFeature_l$, check if $gap\_PosSegment_k(t^{(k)})$ and $gap\_PosSegment_l(t^{(l)})$ have C^0 and C^1 continuity.
- C11: If C10 applies, check if $\dot{C}^{(k)}_{base}(t^{(k)}_K)$, with $C^{(k)}_{base}(t^{(k)}) = base\_DirSegment_k(t^{(k)})$, $t^{(k)} \in [t^{(k)}_0, t^{(k)}_K]$ is parallel to $\dot{C}^{(l)}_{base}(t^{(l)}_0)$, with $C^{(l)}_{base}(t^{(l)}) = base\_DirSegment_l(t^{(l)})$, $t^{(l)} \in [t^{(l)}_0, t^{(l)}_L]$.
- C12: If C11 applies, check if $\dot{C}^{(k)}_{wall}(t^{(k)}_K)$, with $C^{(k)}_{wall}(t^{(k)}) = wall\_DirSegment_k(t^{(k)})$ is parallel to $\dot{C}^{(l)}_{wall}(t^{(l)}_0)$, with $C^{(l)}_{wall}(t^{(l)}) = wall\_DirSegment_l(t^{(l)})$.
- C13: All $weldFeature$ sequences that satisfy C9 until C12 are stored in a separate $weldFeature$ list and form a separate $weldTask$ element.

Note, that for C11 and C12 only the first derivatives have to have the same direction, which does not represent C^1 continuity as the respective curve segments for base and wall directions between the two weld features do not have to be C^0 continuous.

4. Experimental validation

The usage of the manufacturing task description including the automatic detection of weld tasks on product CAD models is tested for different welding product geometries. Therefore, the class diagram of the manufacturing task description (see Fig. 2) is implemented in a C++ test environment. Moreover, a set of custom build functions based on the OCCT class libray are used to realize the automatic recognition of weld features and weld tasks on product CAD models as already presented in Fig. 3.

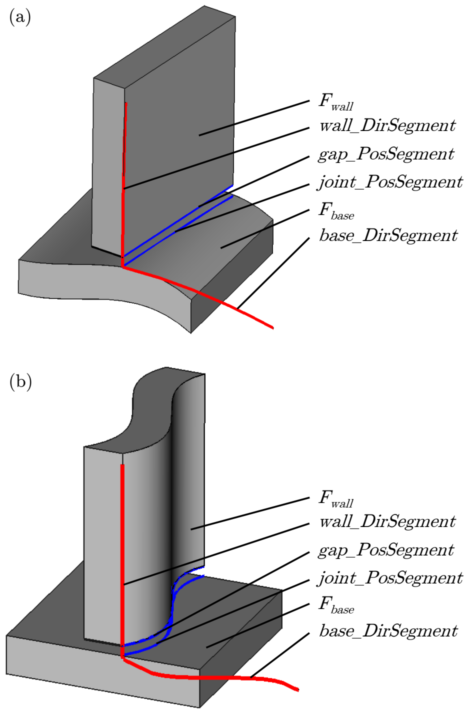

Fig. 6: Weld feature recognition for a joint geometry based on the intersection between (a) a plane F_{wall} face and a B-Spline F_{base} face and (b) a B-Spline F_{wall} face and a plane F_{base} face

Fig. 6 shows the results of the weld feature recognition process based on the rules presented in Section 3.1 for two different product geometries. In Fig. 6 (a), a plane surface F_{wall} is intersected with a B-spline surface F_{base}. Here, the *joint_PosSegment*, *base_DirSegment* and *wall_DirSegment* are of the type *B_Spline_Curve_Segment*. The *gap_PosSegment* is represented as *Line_Segment*. In Fig. 6 (b), a B-spline surface F_{wall} is intersected with a plane surface F_{base}. In this case, the *joint_PosSegment*, *gap_PosSegment* and *base_DirSegment* are of the *B_Spline_Curve_Segment* type. The *wall_DirSegment* is represented as *Line_Segment*. These test cases show that the curve types of the detected weld features depend on the geometry of the intersected faces.

Fig. 7 shows the weld task recognition for a workpiece with associated weld features with different curve segment types. Here, the $joint\_PosSegment_j(t^{(j)})$ is a *B_Spline_Curve_Segment*, the $joint\_PosSegment_k(t^{(k)})$ is a *Circle_Segment* and the $joint\_PosSegment_l(t^{(l)})$ is represented as *Line_Segment*. In this case, $weldFeature_j$ and $weldFeature_k$ are combined to a $weldTask_m$ as $joint\_PosSegment_j(t^{(j)})$ and $joint\_PosSegment_k(t^{(k)})$ have C0 and C1 continuity and the weld features also satisfy all other rules for recognition of weld tasks (see Section 3.2). The $joint\_PosSegment_l(t^{(l)})$ of

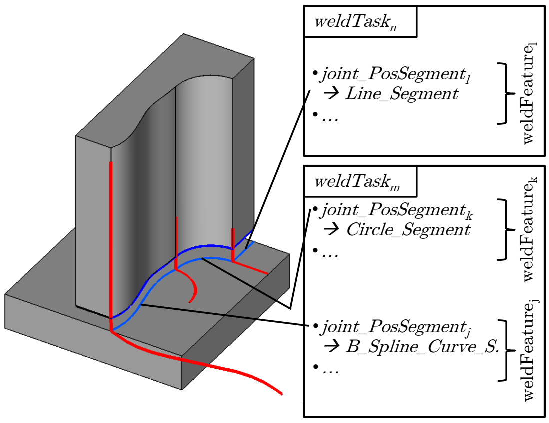

Fig. 7: Task recognition based on associated weld features with different curve segment types

$weldFeature_l$ also satisfies the C0 continuity condition with the $joint\_PosSegment_k(t^{(k)})$ of $weldFeature_k$. However, both joint segments are not C1 continuous, so that $weldFeature_l$ is assigned to a separate $weldTask_n$.

5. Conclusion and future work

In this paper, a novel ontological representation of continuous-path manufacturing tasks has been presented for the example of robotic welding tasks. The description represents information on the joint geometry of the welding product as well as of the geometry of the weld seams to be produced. This geometric information is based on simple continuous curve primitives, like lines, circular arcs or B-spline curves allowing the usage in robot program planning systems without the need for complex geometry engines. Moreover, the manufacturing task description enables the representation of information on welding gaps along the joint geometry which is important in robot program planning for manually prepared workpieces. An XML representation of the manufacturing task description has been presented that can be attached to the original CAD file of a welding product supporting an easy integration in existing robot offline-programming systems. Moreover, a method for automatic rule-based recognition of welding tasks on product CAD models has been presented. The method was shown for the recognition of weld features on product CAD models and the subsequent recognition of weld tasks. Specific rules for detection of weld features based on T-joints and fillet welds have been presented. Especially, the process of mathematically transforming shape-related information of the B-Rep CAD model into the continuous curve primitives of the manufacturing task description has been discussed. Moreover, rules for recognition of weld tasks were presented based on continuity conditions between associated weld feature elements. The proposed method has been evaluated on several test scenarios with different geometries of the welding products showcasing its efficiency for weld task recognition on real workpiece geometries.

Future work will focus on the implementation of the presented manufacturing task description in robotic offline programming environments and the performance testing in real robotic welding processes. Furthermore, the task description can be extended for other joint geometries and weld types as well as for other continuous-path manufacturing processes, like e.g. deburring or gluing. Moreover, future work will include the representation of tolerance information in the proposed framework.

References

[1] Kuss, A., Diaz P., J.R., Hollmann, R., Dietz, T., Haegele, M.. Manufacturing knowledge for industrial robot systems: Review and synthesis of model architecture. 12th IEEE Conference on Automation Science and Engineering 2016;.

[2] Pan, Z., Polden, J., Larkin, N., Van Duin, S., Norrish, J.. Recent progress on programming methods for industrial robots. Robotics and Computer-Integrated Manufacturing 2012;28(2):87–94.

[3] Lee, J., Kim, K.. Generating alternative interpretations of machining features. The International Journal of Advanced Manufacturing Technology 1999;15(1):38–48.

[4] Han, J., Pratt, M., Regli, W.C.. Manufacturing feature recognition from solid models: a status report. IEEE Transactions on Robotics and Automation 2000;16(6):782–796.

[5] Gao, J., Zheng, D., Gindy, N.. Extraction of machining features for cad/cam integration. The International Journal of Advanced Manufacturing Technology 2004;24(7-8):573–581.

[6] VDI 2218. Information technology in product development - feature technology. Guideline; Verein Deutscher Ingenieure; 2003.

[7] Reinhart, G., Munzert, U., Vogl, W.. A programming system for robot-based remote-laser-welding with conventional optics. CIRP Annals-Manufacturing Technology 2008;57(1):37–40.

[8] Liu, Z., Bu, W., Tan, J.. Motion navigation for arc welding robots based on feature mapping in a simulation environment. Robotics and Computer-Integrated Manufacturing 2010;26(2):137–144.

[9] Bickendorf, J.. Robotic welding of ship-subassemblies with fully automatic offline-programming. In: ISR/Robotik 2014; 41st International Symposium on Robotics. VDE; 2014, p. 1–7.

[10] Kuss, A., Schneider, U., Dietz, T., Verl, A.. Detection of assembly variations for automatic program adaptation in robotic welding systems. In: Proceedings of ISR 2016: 47st International Symposium on Robotics. 2016, p. 1–6.

[11] Kuss, A., Dietz, T., Spenrath, F., Verl, A.. Automated planning of robotic mag welding based on adaptive gap model. 10th CIRP Conference on Intelligent Computation in Manufacturing Engineering 2016;.

[12] Maropoulos, P., Yao, Z., Bradley, H., Paramor, K.. An integrated design and planning environment for welding: Part 1: Product modelling. Journal of Materials Processing Technology 2000;107(1):3–8.

[13] Norberto Pires, J., Godinho, T., Ferreira, P.. Cad interface for automatic robot welding programming. Industrial Robot: An International Journal 2004;31(1):71–76.

[14] Vosniakos, G., Chronopoulos, A.. Industrial robot path planning in a constraint-based computer-aided design and kinematic analysis environment. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 2009;223(5):523–533.

[15] Polden, J., Pan, Z., Larkin, N., Van Duin, S., Norrish, J.. Offline programming for a complex welding system using delmia automation. In: Robotic Welding, Intelligence and Automation. Springer; 2011, p. 341–349.

[16] Henderson, M.R., Anderson, D.C.. Computer recognition and extraction of form features: a cad/cam link. Computers in industry 1984;5(4):329–339.

[17] Hillbrand, C., Frank, G.. Knowledge-based automated programming of welding robots for lot-size one products. In: ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis. American Society of Mechanical Engineers; 2012, p. 27–36.

[18] DIN EN ISO 2553:2014(E). Welding and allied processes - Symbolic representation on drawings - Welded joints. Standard; International Organization for Standardization; 2014.