Automatic Optimal Motion Generation for Robotic Manufacturing Processes: Optimal Collision Avoidance in Robotic Welding

Julian R. Diaz P., Thomas Dietz, Philip Ockert, Alexander Kuss, Martin Hägele and Alexander Verl[1]

Abstract— **Optimal, efficient and intuitive robotic programming is still a challenge in robotic manufacturing and one of the main reasons why robots are not widely implemented in small and medium-sized enterprises (SME). In order to effectively and efficiently respond to the current product variability requirements, SMEs require easy and optimal programmable robotic manufacturing systems in order to achieve profitable and rapid changeover. To make up for this deficiency, this paper proposes a solution approach for computing optimal motions for manufacturing processes based on the interpretation of the manufacturing process and an automatic configuration of a state of the art sample-based algorithm, the Rapidly-exploring Random Tree RRT* which is provably asymptotically optimal, using as inputs the semantic and mathematical descriptions of the product, process and resource components. The approach is simulated on the example of collision avoidance for different scenarios in robotic welding revealing its functionality and outlining future potentials for the optimal motion generation for robotic manufacturing processes.** [1]

I. Introduction

Robot systems for manufacturing purposes have become more common in industry due to its low cost and versatility. However the main challenge for this technology, in order to also be spread into SMEs and face the current re-configuration and high quality production demands, is the time-consuming, expert dependent and complex programming process. According to the latest IFR statistics, most robots are installed in automotive producing countries showing a tremendous potential of installing them on emerging and traditional markets in which the unmodernized retooling is one of the limitations [1]. Moreover, according to the most recent NIST report, up to 60 percent of the cost of deploying a robot system is due to the system's integration efforts in comparison with the up to 25 percent cost of the robot itself [2], depicting the re-configuration problems in SMEs.

Several efforts have been made in order to overcome these limitations in robotics. Knowledge representation and semantics have been introduced from the early development of robotics' theory. For instance, the task frame formalism was introduced for synthesizing of robot force control [3]. Kinematic and dynamic description of spatial transformations is nowadays adopted as open-source software [4]. ITASC has been introduced for reducing complexity when programming robots [5]. Additionally, manufacturing process descriptions have been proposed separately. As an example: the product, process and resource model (PPR) is widely used in industry. This model is used for instance in the process planning phase [6], on the exchange of plant engineering information [7,8] and in some of the state of the art Computer-Aided Manufacturing (CAM) software. Moreover, optimal models and standards exist for several manufacturing processes. However, a semantic representation and modeling for robotic manufacturing processes is not found in literature.

Today, industry counts on highly developed commercial and open-source robot simulator software [9,10] which allows modeling of robot cells, collision checking and mapping of robot dependent properties such as reachability, maximal joint limits and singularities. Moreover, motion planning algorithms such as the sample-based technique, which has emerged as an efficient and extended solution to the complex motion problems of robots [11,12] in several applications [13] and in which optimality has been researched [14] is now available in commercial and open-source software [15]. Even with these developments, there is potential for further development of these algorithms and software for improving robotic manufacturing.

In the field of welding, the need of optimization of controllable parameters and an approach for modeling the process, robot and sensors for preparing the task offline has been pointed out [16]. Stand-alone optimization of the robot motion with respect to the manufacturing process are found in the literature. For instance, Dolgui proposed that the weld may be processed in the out-of-position location charged by reduction of the welding speed in order to satisfy the quality specifications and considered a cluster-based welding operations plan for minimizing processing time [17]. Huo introduces a redundancy-resolution algorithm to optimize the joint space trajectory of an arc-welding task [18]. Baron studied a joint-limits avoidance strategy by a virtual rotation around the electrode axis [19]. Henriques proposed using the screws theory for optimal welding positioning [20]. Kim [21] uses information of a laser vision sensor for programming robot welding tasks.

Although, there is a representative development in the field of semantics and knowledge representation, CAM and CAD software for the simulation and generation of motions, motion planning algorithms in robotics and modeling for optimal processing; it is still missing an automatic, changeable and structured approach for overcoming the re-configuration

This work is supported by the European Commission, within the projects FP7 ICT-258769 SMErobotics and H2020 ICT-688217 ROBOTT-NET.

Julian R. Diaz P., Thomas Dietz, Philip Ockert, Alexander Kuss and Martin Hägele are with the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA). Nobelstrasse 12, D-70569 Stuttgart, Germany. E-mail: (julian.diaz.posada@ipa.fraunhofer.de)

[1] Alexander Verl is with the Institute for Control Engineering of Machine Tools and Manufacturing Units ISW, University of Stuttgart, Seidenstrasse 36, 70174 Stuttgart, Germany

[1]A video is provided with the paper in order to depict the proposed approach.

problems of motion generation for manufacturing robot systems which is the main contribution of this paper.

Therefore, with the first goal of simplifying programming of a robot manufacturing process (RMP), and second of decreasing the dependency of manufacturing process and robot experts in the re-configuration phase, this paper proposes an approach, architecture and mathematical notation based on the widely spread PPR model for configuring automatically state of the art sample-based motion planner algorithms in order to optimally solve a robotic manufacturing task.

The functionality of the approach is evaluated through simulations on the example of collision avoidance while specifying an optimal welding process for three scenarios. Simulations demonstrate an effective and efficient offline programming when uncertainties such as collision avoidance are present in a manufacturing task. The approach opens new perspectives on the interpretation and in the automatic optimal motion generation of robotic manufacturing processes.

This paper is organized as follows: Section II defines the problem of optimal motion generation for robotic manufacturing processes and introduces the solution approach and architecture. Section III describes the interpretation of the robotic welding process and the automatic configuration of a RRT* [14] for calculation of optimal motions when collision avoidance is required. Section IV presents the simulation results on the three different welding scenarios. Finally, conclusions, potentials and the future work are discussed in section V.

II. Problem definition and approach

This section formally defines the problem and outlines the general solution approach.

A. Problem definition

To address improvement in robot manufacturing systems configurability while providing manufacturing quality the following motion problem generation is defined:

Compute automatically and offline a feasible and optimal motion σ^* (as defined in [14]) which satisfies the initial and final configuration for the manufacturing process and in which the manufacturing task is optimally accomplished as much as possible, under a specified manufacturing process optimum and having as inputs a *World* W, in this case the Cartesian space $W = \mathbb{SE}3$, in which the *Product*, *Process* and *Resource* (*PPR*) components are defined.

B. Solution Approach

In order to automatically solve the problem defined in subsection II-A, first a systematic interpretation of the relations of the *PPR* components in respect to the motion is proposed, followed by the computation of an optimal motion in a sample-based motion algorithm, performed in the architecture denominated the optimal motion generator. Fig. 1 depicts the architecture for solving the problem. Inputs for the interpreter are the semantic descriptions and mathematical models of the PPR components. Outputs of this interpreter are: (1) the state validation definition which constitutes the formulation for determining if a state could be taken into account for a motion, (2) the manufacturing motion cost (notated as RMP_{Cost}) which constitutes a quantification of the relation between interconnected valid samples and the manufacturing optimum and (3) the required configuration for a state of the art sample-based motion algorithm. The outputs of this interpreter are then run into the sample-based algorithm for computing an optimal motion which is later parsed into a robot program constituting the output of the proposed architecture and solution approach.

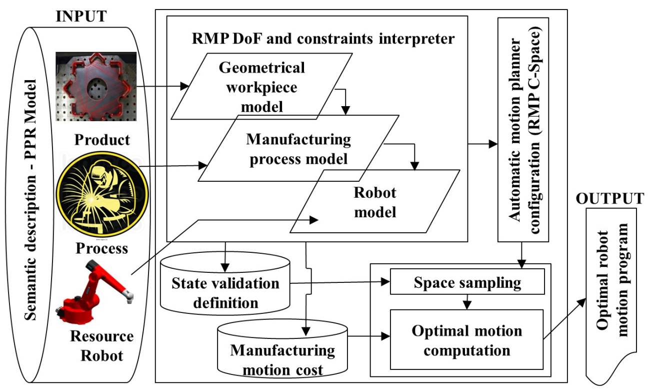

Fig. 1. Architecture for the automatic offline optimal motion generation for robotic manufacturing processes (DoF: Degree of Freedom)

For computing robot manufacturing optimal motion the approach proposes the incorporation of a new space (denominated the robot manufacturing process configuration space ($\mathbb{RMP}$) in addition to the Cartesian space (defined as $\mathbb{SE}(3)$) and the joint space (notated as $\mathbb{J}$). Functionality of this new space is to relate motions with the product, process and resource components of a manufacturing system. Main goal of the inclusion of this space is the simplification of the description of a manufacturing process and achieved by satisfying the *PPR* constraints into the space and setting as dimensions the common free DoF of the *PPR* components. Fig. 2 depicts the involved spaces in the approach.

Moreover, the approach uses inverse kinematic $\underline{j} = IK(\underline{x})$ for transforming poses $\underline{x}$ from $\mathbb{SE}(3)$ to joint positions $\underline{j}_{[mx1]}$ in $\mathbb{J}$ (where m symbolizes the number of robot joints) and the forward kinematics $\underline{x} = FK(\underline{j})$ the other way around. Similarly, the transformation from the $\mathbb{RMP}$ to $\mathbb{SE}3$ is introduced and notated as $T^{\mathbb{SE}3}_{\mathbb{RMP}}$. Exemplification of the use of this transformation is demonstrated for robotic welding in Section III.

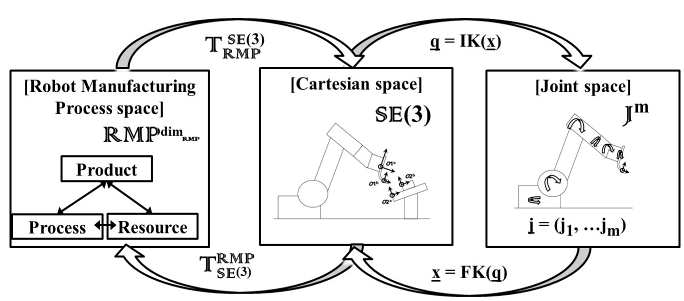

Fig. 2. Spaces for the optimal motion generation in robotic manufacturing

The solution approach for solving the defined problem then becomes the interpretation of the *PPR* model in the world for automatically finding a configuration of an sample-based algorithm for offline finding the feasible and optimal motion σ^* as notated in (1)

$$\begin{pmatrix} World, \\ Product, \\ Process, \\ Resource, \\ RMP_{Cost} \end{pmatrix} Interpret \rightarrow \begin{pmatrix} \mathbb{RMP}, \\ dim_{RMP}, \\ bounds_{RMP}, \\ RMP_{init}, \\ RMP_{goal} \end{pmatrix}, \qquad (1)$$

where the $\mathbb{RMP}$ defines the robot manufacturing process space with dimension dim_{RMP}, bounds $bounds_{RMP}$ derived from the *PPR* model and with initial and goal states notated RMP_{init} and RMP_{goal} respectively.

III. Robotic welding interpretation and motion generator configuration

This section describes how the proposed approach interprets a welding process, and configures the RRT* algorithm for finding an optimal motion for avoiding collision while maintaining as much as possible the optimum welding angle based on the previous described solution approach. This is a typical and common problem in SMEs which consumes time when programming an industrial robot and which normally requires collision avoidance for eluding for instance clamping mechanisms. Firstly, the product, process and resource model are defined. Secondly, the state and motion cost for robotic welding are specified. Finally, the interpretation and configuration algorithm is presented.

A. Product model

The product model, notated as (p), consists of the mathematical description using homogeneous transformations of single or multiple-continuous T-joint seams of a welding workpiece, which are selected by the end user with a click from the CAD. The description of the translational component of the homogeneous matrix consists of parameterized linear equations (with k as parameter) on the Cartesian space by having as input the start and end point of each of the selected seams $\vec{P}_{\vec{S}_n}$ and $\vec{P}_{E_n}$ respectively (see Fig. 3), where n describes the numbers of the selected continuous T-weld seams as described in algorithm 1. Each T-weld seam is parameterized from $100(n-1)$ to $100n$ (see Fig. 3).

The rotational component of the welding product is modeled by obtaining a quaternion notated as $\underline{q}_n$ computed based on the normal vectors of the two T-joint plates. A percentage of the k factor in which rotation between seams may occur is introduced and notated as ζ (see Fig. 3). Rotations from seam to seam are calculated as follows $\delta\underline{q}_{([n]\to[n+1])} = (\underline{q}_n)^{-1} \cdot \underline{q}_{[n+1]}$ and transformed to the axis-angle convention with the following function $[\alpha_n, \vec{v}_n] = Q2AA(q_{[n]\to[n+1]})$. The minimum rotational angle α_n is then parameterized with the previously introduced k factor for defining the optimal orientation from corner to corner as follows,

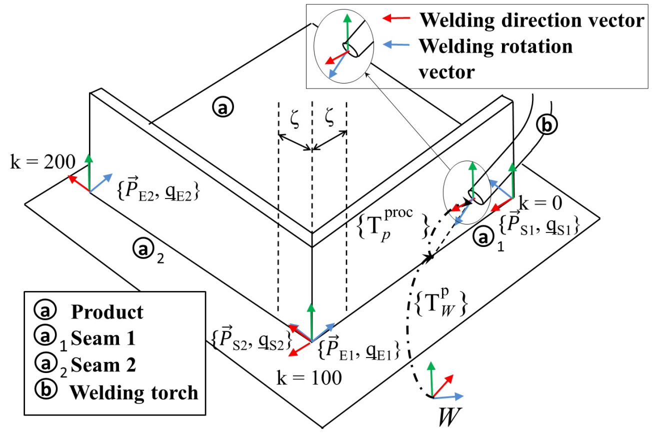

Fig. 3. Illustration of the T-Weld seam product model components and notations

$$\beta(k,n) = \alpha_n \cdot \left(\frac{k-(100n-\zeta)}{2\zeta}\right). \qquad (2)$$

The rotational component is also defined depending on the amount of selected seams as described in Algorithm 1. This algorithm uses the function $R_q(q_n)$ to calculate a rotation matrix from a quaternion and the function $R_{aa}(\beta(k), \vec{v}_n)$ for transforming from axis-angle into rotational matrix notation. The descriptions of rotation and translation are then used as in algorithm 1 for describing the homogeneous matrix for each range of the seams. Output of algorithm 1 is the transformation World to product $T_W^P(k)$ parameterized in k as described in equation (3). The DoF of the product (notated DoF_p) is then k and for this model of one dimension.

Algorithm 1 Algorithm for automatic interpretation of the multiple seam product model

Require: $(\vec{P}_{S_1}...\vec{P}_{S_n}), (\vec{P}_{E_1}...\vec{P}_{E_n}), (\underline{q}_1...\underline{q}_n),\ \zeta,\ n$

$T_W^P(\ DoF_p\ ) = T_W^P(k) =$

if $n = 1$ **then**

$\quad \begin{bmatrix} R_q(\underline{q}_1) & \vec{P}_{S_1} + (\frac{k}{100})(\vec{P}_{E_1} - \vec{P}_{S_1}) \\ 0 & 1 \end{bmatrix}, k \in [0, 100]$

else

$\quad \begin{bmatrix} R_q(\underline{q}_1) & \vec{P}_{S_1} + (\frac{k}{100})(\vec{P}_{E_1} - \vec{P}_{S_1}) \\ 0 & 1 \end{bmatrix},\ k \in [0, (100-\zeta))$

$\quad$ **for** $m = 1$ to n **do**

$\quad\quad \begin{bmatrix} R_q(\underline{q}_m) \cdot R_{aa}(\beta(k), \vec{r_m}) & \vec{P}_{S_m} + (\frac{k}{100})(\vec{P}_{E_m} - \vec{P}_{S_m}) \\ 0 & 1 \end{bmatrix},$
$\quad\quad k \in [(100m-\zeta), (100m))$

$\quad\quad \begin{bmatrix} R_q(\underline{q}_{m+1}) \cdot R_{aa}(\beta(k), \vec{r_m}) & \vec{P}_{S_{m+1}} + (\frac{k}{100})(\vec{P}_{E_{m+1}} - \vec{P}_{S_{m+1}}) \\ 0 & 1 \end{bmatrix},$
$\quad\quad k \in [100m, (100m+\zeta))$

$\quad\quad \begin{bmatrix} R_q(\underline{q}_{m+1}) & \vec{P}_{S_{m+1}} + (\frac{k}{100})(\vec{P}_{E_{m+1}} - \vec{P}_{S_{m+1}}) \\ 0 & 1 \end{bmatrix},$
$\quad\quad k \in ((100m+\zeta), (100(m+1)-\zeta)]$

$\quad$ **end for**

$\quad \begin{bmatrix} R_q(q_n) & P_{S_n} + (\frac{k}{100})(P_{E_n} - P_{S_n}) \\ 0 & 1 \end{bmatrix}, k \in ((100n-\zeta), 100n]$

end if (3)

B. Welding process model

The welding process model (*proc*) is defined with a homogeneous transformation notated as ($T_p^{proc}(\cdot)$), meaning the transformation from product to process, using the Euler convention which is more intuitive for the end user when defining constraints and free DoFs of the manufacturing process. The description is performed locally, meaning with respect to the coordinate system of the seams of the workpiece (see Fig. 3).

Position and rotational constants for this transformation are values obtained from standards, for the case of welding ISO 22553, and notated as follows $(x,y,z,a,b,c)_{proc}$. DoF of the process DoF_{proc} are defined depending on its nature and notated as $(\delta x, \delta y, \delta z, \delta a, \delta b, \delta c)_{proc}$, please note that depending on the process some of these DoF do not exist. The following equation states the homogeneous transformation from product to process of a general manufacturing process. Detailed description for the simulated welding process model is given with its semantic description in IV-A.

$$T_p^{proc}\left(\; DoF_{proc} \; \right) = T_p^{proc}\begin{pmatrix} \delta x, \delta a, \\ \delta y, \delta b, \\ \delta z, \delta c \end{pmatrix}_{proc} = \begin{bmatrix} R_{eul}\begin{pmatrix} a+\delta a \\ b+\delta b \\ c+\delta c \end{pmatrix}_{proc} & \begin{bmatrix} x+\delta x \\ y+\delta y \\ z+\delta z \end{bmatrix}_{proc} \\ 0 & 1 \end{bmatrix}. \tag{4}$$

C. Resource model

The resource model, notated as (r), in this case an industrial robot, is defined with the forward kinematics and inverse kinematics as defined in II-B. In this paper, only the kinematic is studied but further research could also be done for optimizing robot dynamics in respect to process models. For instance, optimizing stiffness for robotic milling. The dimension of the robot (DoF_r) in the $\mathbb{RMP}$ for this study is considered null because Cartesian poses are determined from the product-process relation. However, please note that the null space of the robot could also be embedded into the approach for optimization purposes.

D. State and motion cost for robotic welding

The welding process is predefined as optimal if non uncertainties, such as collisions, are present taken into account the specified process parameters. In the case of non uncertainties, sampling on the $\mathbb{RMP}$ is required. Interpretation of the optimal welding is mathematically defined when the variables of the welding process model are null $T_p^{proc}(\varnothing)$. Please note that for other processes and cases it may occur that the optimum for manufacturing is not previously known. For instance, optimizing the robot stiffness in a path similarly as in one of our last works [22]. Based on this interpretation, the optimal function for the welding process, in respect to the world, of one seam or multiple continuous seams is determined with the function $ComputeOptimal(\cdot)$ as follows,

$$T_W^{proc_{opt}}(k) = T_W^p(k) \cdot T_p^{proc}(\varnothing). \tag{5}$$

Based on the optimal function for the welding process, the state cost is defined as the required angle (calculated using the axis-angle convention) of a non-collision generated sample α_k for reaching the optimum α_{opt} on a certain k of the parameterized T-weld seams as follows $\mid \alpha_{opt} - \alpha_n \mid$. Moreover, The motion cost in respect to optimum (RMP_{Opt}) between two non-collision samples $S_1 = [k_{S1}, \alpha_{S1}]$, and $S_2 = [k_{S2}, \alpha_{S2}]$ is calculated by computing the area in between the line which connects those two samples $Line(k) = Line(S_1, S_2) = \frac{\alpha_{S2}-\alpha_{S1}}{k_{S2}-k_{S1}} k + \alpha_{S1} - \frac{\alpha_{S2}-\alpha_{S1}}{k_{S2}-k_{S1}} k_{S1}$ and the optimal definition of the motion $T_W^{proc_{opt}}(k)$, mathematically formulated as described in lines 2 and 5 of algorithm 2.

The function $FindIntersection(\cdot)$ is implemented for finding the crossing point between two functions required for differentiation between the two different integration cases. The motion cost in respect to optimum evaluates the closeness of the orientation in respect to the optimal welding angle. Please note that just the kinematic is taken into account and that further specification of dynamic constraints in respect to the welding process could improve the computed motions. Moreover, a motion cost policy is also implemented in order to penalize backward motions $RMP_{Backward}$ which are undesired for the welding process. The motion final cost is then calculated as a sum of the last explained two motion cost as notated in (6). Algorithm 2 describes the calculation of the proposed motion cost.

Algorithm 2 Optimal motion cost for collision avoidance for robotic welding

Require: $S_1 = [k_{S1}, \alpha_{S1}]$, $S_2 = [k_{S2}, \alpha_{S2}]$, $T_W^{proc_{opt}}(k)$

{Cost calculation for closeness to optimal welding angle}

1: **if** $[(Q2AA(T_W^{proc_{opt}}(k_{S1})) > \alpha_{S1}) \vee (Q2AA(T_W^{proc_{opt}}(k_{S2}) > \alpha_{S2}))] \wedge [(Q2AA(T_W^{proc_{opt}}(k_{S1})) < \alpha_{S1}) \vee Q2AA((T_W^{proc_{opt}}(k_{S2})) < \alpha_{S2})]$ **then**

2: $RMP_{Opt} = \int_{k_{S1}}^{k_{S2}} Line(k) \cdot T_W^{proc_{opt}}(k) dk$

3: **else**

4: $[k_c, \theta_c] \leftarrow FindIntersection(Line(k), T_W^{proc_{opt}}(k))$

5: $$RMP_{Opt} = \left| \int_{k_{S1}}^{k_c} Line(k) \cdot T_W^{proc_{opt}}(k) dk \right| + \left| \int_{k_c}^{k_{S2}} Line(k) \cdot T_W^{proc_{opt}}(k) dk \right|$$

6: **end if**

{Cost policy if backward motion is computed}

7: **if** $(k_{S2} < k_{S1})$ **then**

8: $RMP_{Backward} = \infty$

9: **else**

10: $RMP_{Backward} = \varnothing$

11: **end if**

12: $$RMP_{Cost} = RMP_{Opt} + RMP_{Backward} \tag{6}$$

E. Welding process interpretation and automatic optimal motion planner configuration

For automatically configuring the RRT* algorithm, some functions are required. One of these functions $DetermineInitGoalStates(p, proc, r)$ determines the initial (RMP_{init}) and goal state (RMP_{goal}) parameterized in k before and after the collision taking into account a specific motion percentage among k for allowing the reaction. For finding these collisions, the function $Collision(\cdot)$ has been implemented using the triangle-triangle collision algorithm for cross checking all CADs involved in the robot scene. Collisions are iteratively checked for the optimum $T_W^{proc_{opt}}(k)$ for each discretization step defined for k (δk).

Another required function is $DetermineDim(p, proc, r)$. This function defines the dimensions of the $\mathbb{RMP}$ ($dim_{\mathbb{RMP}}$) by summating the number of variables of each of the mathematical description of the PPR components (i.e. DoF_p, DoF_{proc}, DoF_r). The number of dimensions is automatically configured into a RealVectorStateSpace object in OMPL [15] and should be a natural number $dim_{\mathbb{RMP}} \in \mathbb{N}$ and $dim_{RMP} \geq 2$ as defined in [14]. The bounds of the $\mathbb{RMP}$ are defined with the minimum and maximum value of the defined variables in the PPR components as in (7) for the amount of DoFs of the manufacturing process. Finding of these bounds is calculated with the function $DetermineBounds(p, proc, r)$. The transformation from robot manufacturing space to Cartesian space is then finally defined as in (8)

$$bounds_{RMP}\begin{pmatrix} RMP_{init}, \\ RMP_{goal} \end{pmatrix} = \begin{cases} k, \begin{cases} k_{ini} = RMP_{init} \\ k_{goal} = RMP_{goal} \end{cases} \\ \theta, \begin{cases} \theta_{ini} = \delta b_{min} \\ \theta_{goal} = \delta b_{max} \end{cases} \\ \vdots \end{cases}, \tag{7}$$

$$T_{\mathbb{RMP}}^{\mathrm{SE3}}\begin{pmatrix} DoF_p, \\ DoF_{proc}, \\ DoF_r \end{pmatrix} = T_W^p\left(DoF_p \right) \cdot T_p^{proc}\left(DoF_{proc} \right) \cdot T_p^r\left(DoF_r \right) \tag{8}$$

The interpretation function defined in (1) is then composed of the functions $DetermineInitGoalStates(p, proc, r)$, $DetermineBounds(p, proc, r)$, $DetermineDim(p, proc, r)$ and $ComputeOptimal(T_W^{proc_{opt}}(k))$ for determining the configuration of the Rapidly-exploring Random Trees RRT* [14] as described in algorithm 3. Finally, after the interpretation, the robotic manufacturing process is configured into the state of the art RRT* sample-based motion generation algorithm, which provides the computation of a provably asymptotically optimal [14], as described in algorithm 3 for finding the optimal motion σ_{opt}^*.

IV. Results

The following section describes the robot system and its semantic modeling, which is a key factor for the changeover abilities of the presented approach. Moreover, simulation for three different welding scenarios requiring collision avoidance in robotic welding are presented.

A. System Description and Semantic Modeling

The robot system is conceived and implemented for studying and improving the cognitive and collaborative abilities of robot welding systems in SMEs. The welding robot cell consists of an industrial robot Reis RV30-26 provided with a welding table and a Schunk tool changer reference SWK-040Q. The robot cell is provided with two endeffectors. In the first one, an Ensenso 3D stereo camera is installed. In the

Algorithm 3 Automatic interpretation and calculation optimal collision avoidance for robotic welding using RRT*

Require: Interpret (*World*, *Product* (eq. (3)), *Process* (eq. (4)), *Resource* (as defined in III-C), RMP_{Cost})

$ComputeOptimal(T_W^{proc_{opt}}(k))$ (as in eq. (5))
Check collision for the optimal path $j = 1$, $k_{ini} = 0$
for $k = 0$ to $100n$ each δk **do**
  if Collision($T_W^{proc_{opt}}(k)$) **then**
    $DetermineDim(p, proc, r)$
    return $dim_{\mathbb{RMP}}$
    $DetermineBounds(p, proc, r)$ as in (7)
    return $bounds_{RMP}$
    $DetermineInitGoalStates(p, proc, r)$
    return (RMP_{init}, RMP_{goal})
    Sample and calculate optimal motion (as in [14] Algorithm 6)
    RRT*($\mathbb{RMP}$, $dim_{\mathbb{RMP}}$, $bounds_{RMP}$, RMP_{init}, RMP_{goal}, $RRTCost()$)
    return Optimal path $\sigma_{j+1}^* \in [RMP_{init}, RMP_{goal}]$
    and optimal path without collision $\sigma_j^* \in [k_{ini}, RMP_{init}]$
    $j = j + 1$, $k = k_{ini} = RMP_{goal}$
  end if
end for
$\sigma_{opt}^* = \sum_{i=1}^{j} \sigma_j^*$

second one, the welding torch is mounted. Fig. 4 depicts the welding robot cell and its components. A TCP/IP communication application is implemented with the robot controller in order to transfer automatically programs generated with the previous described interpretation and configuration of the RRT* algorithm in OMPL [15]. A referencing application using the 3D stereo camera is intended, similarly than in [23], in order to localize precisely the workpiece in the robot cell before the computation of optimal motions.

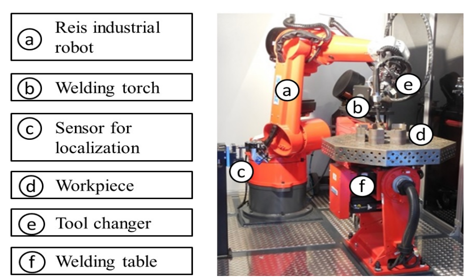

Fig. 4. Welding robot cell

The robot cell is simulated using the CAD files of its elements into an internal developed software configured using the PPR model semantically described using the Automation Markup Language (AML) [24]. The robot, peripherals and endeffectors (tools) are described under the resource classification. The CAD files of the robot joints and its Denavit-Hartenberg kinematic parameters are associated with the resource structure. Links to the forward kinematics and inverse kinematics are also specified in this classification. The CAD file of the product (workpiece) is also linked in this AML format.

The models described in Section III are coded in Python and depending on the scenario (number of edges, edges position and orientations). A C file containing the configuration for the RRT* is automatically generated. The motion generator is programmed in C++ and compiled into a Dynamic-link library linked with the simulation software. This semantic description and software structure facilitates the changeover when facing different requirements of the PPR components. Fig. 5 depicts the first levels of the welding robot cell semantic modeling in the AML editor, highlighting the components of the PPR model.

Fig. 5. Robot system semantic description in AML using the PPR model

The process is modeled into an XML file also linked to the AML, it describes the geometrical transformation from workpiece to process, as depicted in Fig. 3. Parameters defined under the semantic specification (<FrameConstraintsDoF>) specifies the constraints (type="fixed") and free degrees of freedom (DoF) of the manufacturing process (type="range"). These values are directly related with Eq. (3) where values, and its units, are assigned to the constants and the min and max values (See listing 1) of the ranges are assigned to the range of the variables. Listing 1 depicts the semantic model of the MAG-welding process in the XML format.

```
<Process name="Welding">
 <FrameConstraintsDoF>
  <GeoPar name="x" unit="mm" type="Fixed" value="0"/>
  <GeoPar name="y" unit="mm" type="Fixed" value="0"/>
  <GeoPar name="z" unit="mm" type="Fixed" value="100"/>
  <GeoPar name="a" unit="deg" type="Fixed" value="0"/>
  <GeoPar name="b" unit="deg" type="Range" min="−45"
        max="45" value="45"/>
  <GeoPar name="c" unit="deg" type="Fixed" value="0"/>
 </FrameConstraintsDoF>
</Process>
```

Listing 1. Semantical modeling of the welding process in XML format

B. Simulation results

To evaluate the proposed concept, mathematical description and architecture; three different scenarios of collision avoidance for robotic welding were tested using the solution approach described in Section II-B and the interpretation for the welding process as described in Section III. The first scenario evaluates a collision avoidance for a T-weld seam by configuring as DoF of the manufacturing process the rotation around the welding direction vector (see Fig. 3)(i.e. δb) as semantically described in Listing 1. By selecting one edge, the software automatically determines the transformation $T_W^P(k)$ as defined in (5). The algorithm automatically finds the collision and interprets the start and end configuration of the sample-based planner, it automatically configures the $\mathbb{RMP}$ with two DoF (k and δb) as described in algorithm 3.

After the automatic configuration, the RRT* finds an optimal path for avoiding the collision as depicted in Fig. 6. The results in joint space and a sequence of robot movements are also depicted. The sample cost of each valid sample is observed thanks to the color map. The right part and left part of the motion were not sampled due to the pre-definition of optimum for this specific manufacturing process. Fig. 6 depicts 1918 valid states. The motion generator evaluated 9068 collision states which are not depicted. Sampling was evaluated and plotted over ten hours. Solutions for the avoidance of this obstacle without plotting functions are obtained within 10 min of sampling and computation which is reasonable for its implementation in SMEs[2]. Longer sampling results in less motion cost due to the provably asymptotically optimal nature of the sample-based techniques [14]. This behavior is observed on the motion cost obtained for different sampling times as in table I.

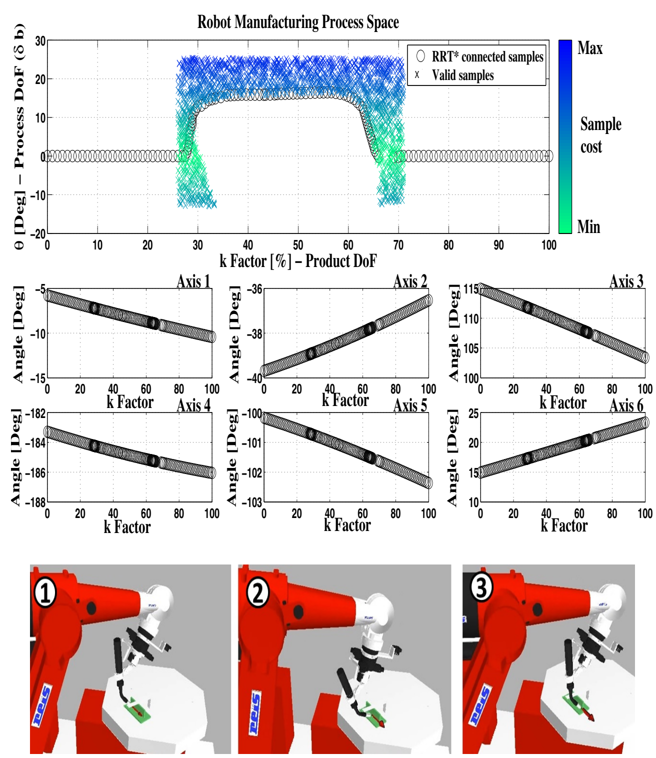

Fig. 6. (Up) Evaluated valid samples and optimal interconnected solution in the Robot Manufacturing Process space for avoiding a collision using the rotation around the welding direction vector (Middle) Joint space with the found optimal solution (Down) Sequences of robot movements in the simulated robot scene demonstrating the optimal collision avoidance

TABLE I

AVERAGE COST FOR DIFFERENT SOLVING TIMES - SCENARIO 1

Sampling time [min]	10	20	30	40	50
Average cost [$Deg \cdot \%$]	1030	945	934	837	791
Number solutions from 5 trials	3	5	4	5	5

[2]Please note that the fastest achievement of the computation of the optimal is out of the focus of this paper.

For testing the re-configurability of the motion generation in respect to changes of the welding process model, a second scenario is evaluated. The free degree of freedom of the process is changed to the welding rotation axis (see Fig 3). This rotation has less influence on the welding process quality and could be easily prioritized by an expert welder without having experience with robotics, which is one of the main advantages of the proposed approach. For re-configuring this feature, changes of lines 7 and 8 of listing 1 are performed as specified in listing 2. Simulation for two different amounts of valid times and samples (2502 and 225) results in optimal motions as depicted in Fig. 7. It is observed that the motion does not vary significantly between both cases for this case.

```
7   <GeoPar name="b" unit="deg" type="Fixed" value="−45"/>
8   <GeoPar name="c" unit="deg" type="Range" min="0" max=
        "45" value="45"/>
9  </FrameConstraints_DoF>
10 </Workpiece2Process>
```

Listing 2. Semantical changes for adapting process DoF

Further heuristics could be implemented based on the delta of the manufacturing motion cost, taking into account the provably asymptotically optimal behavior of the RRT*, for automatically finalizing the sampling and not having to specify sampling times beforehand. It is worth noting that if a motion is not possible the motion generator returns a message to the end user stating that no interconnection between the generated samples was possible. It is important to point out that the rotational component of the start and goal state of the robotic welding process could have also been optimized. However, this is a motion planning problem which, based on the knowledge of the authors, it has not still been solved and which has significant potential for planning robot motions.

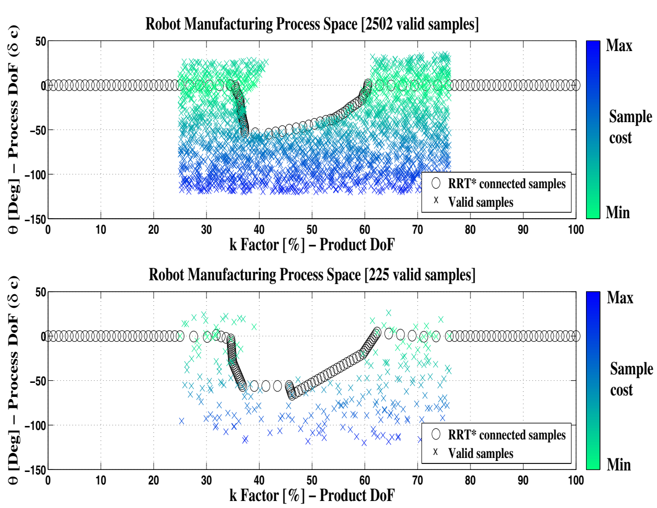

Fig. 7. Evaluated valid samples and optimal interconnected solution in the Robot Manufacturing Process space for avoiding a collision using the rotation around the welding rotation vector for (Up) 2502 and (Down) 225 valid samples

The third scenario evaluates the optimal motion generator by simulating a collision avoidance in the corner between the first and second seam. For setting this scenario, the end user selects more than one continuous edge on the product. The transformation $T_W^P(k)$ is then defined for more than one edge as specified in algorithm 1. The welding direction vector is defined as the DoF of the process.

Fig. 8 depicts the results in $\mathbb{RMP}$. The position and orientation of the robot are also depicted in $\mathbb{SE}(3)$ showing the changes of the manufacturing-process coordinate frame when colliding and the optimal rotation from edge to edge as defined in the product model. Sequences of the robot simulation are also depicted demonstrating the complexity of robot movements achieved by interpreting the *PPR* model for optimal motion generation. The different ranges on the sampled $\mathbb{RMP}$ space are due to the geometry of the workpiece on the corner. The changes of orientation when avoiding the collision are due to the lack of specification of kinodynamic relations between the product the process and the robot, which is part of the proposed further work of this research.

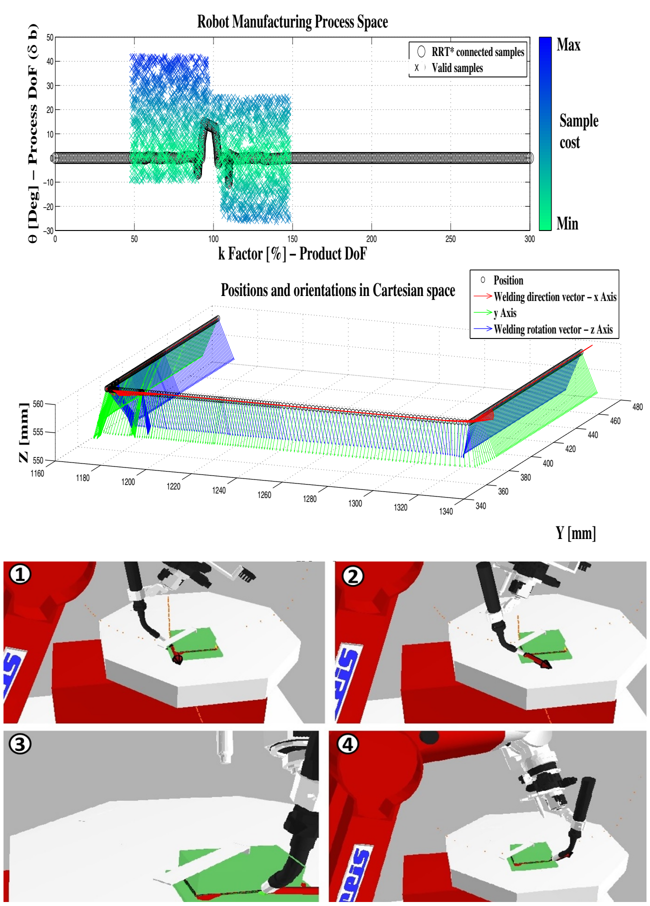

Fig. 8. (Up) Evaluated valid samples and optimal interconnected solution in the Robot Manufacturing Process space for avoiding a collision using the rotation around the welding direction vector for multiple seams (Middle) Cartesian space with the plot of the found optimal welding direction vector (Down) Sequences of Robot movements in the simulated robot scene for the optimal collision avoidance

V. CONCLUSIONS AND FUTURE WORK

This paper describes a solution approach and architecture for the automatic computation of optimal motions for robotic manufacturing processes considered a key factor for facing current re-configurations deficiencies of robotic manufacturing systems leading to simplification of the robot program generation process and avoidance of dependency from manufacturing process and robotic experts.

The paper proposes an interpretation of the manufacturing process by semantically describing and mathematically modeling the product, the process, the resource (a robot) and a manufacturing motion cost, which determines the relation between robot motions and process quality. The interpretation determines the automatic configuration of an Rapidly-exploring Random Trees sample-based algorithm used for the computation of optimal motions when uncertainties are present. The interpretation and automatic configuration is simulated for the optimal avoidance of collision in robotic welding considered a typical and initial scenario for testing the proposed approach, three scenarios varying the degrees of freedom of the manufacturing process and the number of selected T-weld seams are analyzed. Simulation results demonstrate the functionality of the approach and the simplicity when re-configuration is required.

The presented approach is conceptualized for easy re-configuration and therefore further product, process and resource mathematical models could be integrated for computing optimal motions for other manufacturing processes. For instance, stiffness model of industrial robots could be embedded as motion costs for optimizing motions for machining in joint space; robot velocities optimization for welding could be approached by embedding robot dynamic models, using sample-based kinodynamic planning and velocity welding models.

Moreover, different sorts of product variants with other features such as splines or circles could be modeled in order to optimize manufacturing for more complex products. The generation of motions in the robot manufacturing process space could be also linked with state of the art joint space optimization for avoiding singularities or maximal joint movements or optimizing in the null space, for instance. Future work is also the evaluation of the motion generator in the described robot cell using the intended sensor-based workpiece referencing system with which the workpiece could be accurately located in the robot cell for its later automatic programming as detailed described in this paper.

VI. ACKNOWLEDGMENT

The authors would like to acknowledge Mr. Ulrich Schneider and Mr. Milad Geravand for their valuable feedback, Mr. Arjun Sridhar for the improvement of the performance of the collision checking algorithm and Mrs. Luzia Schuhmacher for proofreading this paper at IPA.

REFERENCES

[1] IFR Statistical Department, "World robotics 2014 industrial robots," 2014.

[2] Jeremy A. Marvel, Elena R. Messina, Brian Antonishek, Karl Van Wyk, and Lisa J. Fronczek, "Tools for robotics in sme workcells: Challenges and approaches for calibration and registration: National institute of standards and technology u.s. department of commerce," 2015.

[3] Matthew T. Mason, "Compliance and force control for computer-controlled manipulators," *IEEE Trans on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[4] T. D. Laet, S. Bellens, R. Smits, E. Aertbelin, H. Bruyninckx, and J. D. Schutter, "Geometric relations between rigid bodies (part 1): Semantics for standardization," *IEEE Robot. Automat. Mag.*, vol. 20, no. 1, pp. 84–93, 2013.

[5] J. D. Schutter, *Task Specification and Estimation for Sensor-Based Robot Tasks in the Presence of Geometric Uncertainty*. PhD thesis, Katholieke Universiteit Leuven, Leuven, 2007.

[6] J. Nielsen, *Information Modeling of Manufacturing Processes: Information Requirements for Process Planning in a Concurrent Engineering Environment*. PhD thesis, Royal Institute of Technology, Stockholm Sweden, 2003.

[7] M. Schleipen and R. Drath, "Three-view-concept for modeling process or manufacturing plants with automationml," *International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–4, 2009.

[8] S. C. Sang, H. Y. Tae, and D. N. Sang, "Xml-based neutral file and plm integrator for ppr information exchange between heterogeneous plm systems," *Int. J. Computer Integrated Manufacturing*, pp. 216–228, 2010.

[9] A. Staranowicz and G. L. Mariottini, "A survey and comparison of commercial and open-source robotic simulator software," in *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments*, PETRA '11, (New York), pp. 56:1–56:8, ACM, 2011.

[10] E. Tom, T. Yuval, and T. Emanuel, "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx 2015, seattle, wa, usa, 26-30 may, 2015," in *IEEE International Conference on Robotics and Automation 2015,*, pp. 4397–4404, IEEE, 2015.

[11] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[12] S. M. LaValle, "Motion planning: The essentials," *IEEE Robotics and Automation Society Magazine*, vol. 18, no. 1, pp. 79–89, 2011.

[13] K. I. Tsianos, I. A. Şucan, and L. E. Kavraki, "Sampling-based robot motion planning: Towards realistic applications," *Computer Science Review*, vol. 1, no. 1, pp. 2–11, 2007.

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[15] Ş. Ioan A., M. Mark, and K. Lydia E., "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[16] G. Bolmsjö, M. Olsson, and P. Cederberg, "Robotic arc welding – trends and developments for higher autonomy," *Industrial Robot*, vol. 29, no. 2, pp. 98–104, 2002.

[17] A. Dolgui and A. Pashkevich, "Cluster-level operations planning in arc-welding robot cell with positioning table," 2004.

[18] L. Huo and L. Baron, "The joint-limits and singularity avoidance in robotic welding," *Industrial Robot*, vol. 35, no. 5, pp. 456–464, 2008.

[19] L. Baron, "A joint-limits avoidance strategy for arc-welding robots," in *Département de génie mécanique: Ecole Polytechnique*.

[20] B. Henriques, R. Ventura, C. E. Pereira, A. Queiroz Bracarense, R. Guenther, A. O. Dourado, and D. Martins, "Robot welding trajectory planning using screw theory," in *ABCM Symposium Series in Mechatronics*, pp. 196–201.

[21] P. Kim, S. Rhee, and C. H. Lee, "Automatic teaching of welding robot for free-formed seam using laser vision sensor," *Optics and Lasers in Engineering*, vol. 31, no. 3, pp. 173–182, 1999.

[22] U. Schneider, J. R. Diaz Posada, and A. Verl, "Automatic pose optimization for robotic processes," in *IEEE International Conference on Robotics and Automation 2015,*, pp. 2054–2059, IEEE, 2015.

[23] A. Kuss, M. Drust, and A. Verl, "Detection of workpiece shape deviations for tool path adaptation in robotic deburring systems (to be published)," in *49th CIRP Conference on Manufacturing Systems (CIRP-CMS 2016)*.

[24] R. Drath, A. Lüder, J. Peschke, and L. Hundt, "Automationml - the glue for seamless automation engineering," in *International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 616–623, IEEE, 2008.