Abstract—Friction Stir Welding (FSW) in 3-dimensions, due to process constraints, requires an off-line programming approach. The creation of tool paths based on computer aided models for cutting, machining or traditional welding, exists in numerous of applications. But since FSW, until recently, have not been a 3-dimensional application, no proper solution for this process exists. In this paper we propose solutions on how to create FSW tool paths based on the geometric description of CAD models to auto create and export such to the executing control system. The emphasis is on extracting and evaluating the weldability of the defined segments in order to perform robust welding on complex weld seam geometries.

I. INTRODUCTION

Friction Stir Welding was invented at The Welding Institute (TWI), UK, in 1991 (patent filed in 1992 [14]) and is a solid state joining process. This, which implies welding without melting, has several advantages compare to traditional fusion welding. Except for the superior quality of the weld, there are environmental benefits (no light or smoke) as well.

In terms of welding material, the main focus has been on aluminium and copper, two materials often considered being un-weldable. But the absence of filler material and lack of melting, welding is performed with high reliability and repeatability. The process eliminates, to a great extent, shrinkage, porosity and distortion, which is often associated with traditional fusion welding [7], [8].

During the last decade there have been several industrial applications for which FSW have been the selected joining method. Since the joint strength and the process reliability are trademarks of the process, it has appealed extreme applications with no margin for error. The aerospace industry (welding of fuel tanks) and nuclear waste management (sealing waste canisters [1]) are two successful implementations of the FSW process.

There exist, however, drawbacks with the FSW process. It does require high forces, applied by the machine carrying the welding equipment, to produce the seam. Theses forces and torques does, of course, cause influence on the machine in terms of compliance and it also becomes necessary to use fixtures and backing plates supporting the welded object. The effect of these drawbacks has lead to (i) customized welding aperture and (ii) welding of non-complex seams, limiting a widespread use of FSW in areas such as the automotive industry due to its lack of flexibility.

During recent years, research has been aimed to solve the machine problem using an industrial robot. In [12] and [10] a serial designed manipulator was used, while in [13] a parallel designed robot was used. To handle compliance issues, all three solutions implemented force control. A force/position hybrid control allows the manipulator to move of position controlled basis in one or more direction while the other directions are force controlled, aiming for a desired force value through positional offsets. This means that a reference path always exists, but can be altered in the force controlled directions.

The robot solutions have been proven a valid alternative for welding of aluminium 5xxx and 6xxx plates with thickness up to 5 mm, with no actual constraints in terms of seam complexity other than the process related constraints. Another benefit, striving for new markets, is the relatively low cost and the already widespread use in the manufacturing industry.

The hybrid controlled robot solution is truly beneficial for FSW, since a seam is to be followed while a constant force is to be applied on the workpiece. But there is, however, another issue to consider, namely the tool’s orientation. The orientation is, of course, embedded in the path description, but needs to be well investigated before the welding motion is executed. A proper orientation is normally defined through the workpiece surface’s normal, then added a few degrees tilt (Fig 1). This allows the tool to smoothly slide the surface without causing a material flow in front of the tool.

The procedure of creating a proper weld path can in the 1 and 2-dimensional case be solved on-line, due to the object’s pre-defined planar surface [11]. In those cases the object’s reference frame can be used in defining the tool’s orientation. But on a more complex object with a curved surface, such solution is invalid. In those cases the welding path needs to be planned off-line, using CAD data references to create a properly defined reference path.

II. BACKGROUND

The use of geometric models for creation, simulation and verification of robot paths, have been in use for many years. Off-line programming (OLP) tools, in which the operator can develop a robot program without accessing the actual robot, have been a great aid in many applications (e.g. welding and painting) due to difficulties in both modelling the manipulators environment and the creation of complex motions.

But in many of today’s commercial OLP tools, the object of manipulation is generally not the focus of attention. It is rarely used as an input source for path planning, but rather as an obstacle in collision detection. In our system we emphasis on just that, using the detailed
geometric model of the object for path planning of in-contact robot operations. When reviewing other research approaches related to the FSW processes, we find:

- Robotic grinding [9], [16]
- NC milling [4], [2]
- NC cutting [15]

which all, in one way or the other, addresses the same types of problem, namely to approximate a (optimal) path from a sophisticated free-form CAD geometry.

As the material, during the FSW process, is not melted, gravity has no influence. This key feature makes the FSW process more similar to e.g. grinding than traditional fusion welding, in terms of path modelling and restraints. And it is rather a tool contact evaluation, as in milling, than a robot joint motion optimization, that should be the emphasis of such modelling dilemma.

III. IMPLEMENTATION

As mentioned earlier in the introduction, the pure object frame does not consist with enough information in case of a curved welding surface. Instead we rely on extracting data from the CAD drawing (or workpiece model) to produce a well defined path, including tool orientations.

The geometric model of the object to be welded (or objects to be joined), often consist of valid information for the seam. This since seams, not rarely are placed on boundaries shared by two entities, as in a butt-joint where the two mating edges defines the location of the seam. And even if the model consists of two separate objects, and thereby no shared edge, it is likely that the two separate edges are geometrically related. In Fig 2 we demonstrate an object in which the two surfaces (TF1 and TF2) are to be joined at the shared edge (TE4).

The strategy described and evaluated in this paper can be divided into three sub implementations, namely:

- Topological extraction in which process of extracting sub shapes (topologies) takes place
- Curve representation in which the entity (edge) is represented as a bounded curve.
- Path feature extraction in which the path is being modelled based on features of the curve.

The procedures are visualized in Fig 3.

A. Topological extraction

When representing complex solid models, there exist different approaches. One is a technique called constructive solid geometry (CSG), in which solids are expressed as Boolean combinations of simpler solids. Another approach is the so called boundary representation (B-Rep) in which the object is represented as a organized collection of surfaces, bounded by edges, which in turn is bounded by vertices.

To visualize this we again use Fig 2 as reference and review the content and the internal connectivity Fig 4. It should of course be noted that the object in Fig 2 is not a solid object since it is not dimensionally homogeneous, but merely a simplified example of a topological data structure.

It should however be noted that topological entities sometimes are shared, which both enables benefits and drawbacks. As mentioned earlier, a shared entity can be a good location for welding. But when doing a top-down
In the process of creating FSW path segments from curves, there are two main questions to consider, namely:

- Does the curve represent an executable motion from a manipulator point of view?
- Is the path valid from a process point of view?

These questions can of course be subdivided into specifics, considering e.g. manipulator inverse kinematics, motion optimization and process dynamics, but in this study we focus on just one thing, namely in creating a path using the available motion instructions of the manipulator, applying proper FSW tool orientations. Such scenario states the following restraints:

- Motions can only be expressed as straight lines or arcs.
- Orientation must be applied based on the underlying surface geometry and the traverse tilt angle.

Considering the type of motion, straight or circular, at first, we apply two strategies. One being adopting to the user’s choice and the other to self-detect the type of motion best suited for the given curve, $C(u)$. In the latter case, an examination of the curves characteristics will classify its type, e.g. line, circle etc.

In all cases, the curves boundary points are used as end-points (boundary control points) for the new segment. These are easily extracted using the direct point solution at $u = 0$ and $u = 1$. If the curve is classified as a line or the user has requested a linear segment, this is all that is needed, in terms of positional data.

If the curve, on the other hand, is arc-shaped and therefore represented as a circle, a dilemma occurs. Depending on the direction of travel, a circle with two boundary points represents two arcs. One in the positive direction and one in the negative. Based on the topological representation of the edge, there is no way of telling which direction to use.

There is, however, a solution available. Recalling the topological scanning procedure, which has a hierarchic structure, the edge is a part of a face. The face, modelled as a surface, holds the information needed to determine which arc represents the "true" curve.

By calculating the positional data at $u = 0.5$ for both curves, denoted as $C_{pos}(u)$ and $C_{neg}(u)$, and then calculate the distance to the surface based on the orthogonal point-to-surface projection for both points, the closest distance will tell which arc to use. The point $C(u)$ is then used to complete the circular segment.

In terms of applying a proper orientation to the control points, the surface once again come in use. Consider a parameterized surface as:

$$ S(u, v) = (x(u, v), y(u, v), z(u, v)) $$

having its tangential vectors defined as:

$$ \frac{\partial f}{\partial u} = \left( \frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right) $$

$$ \frac{\partial f}{\partial v} = \left( \frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v} \right), $$

for $C(u)$, which represents the lowest degree polynomial for describing non-planar curves. And by bounding the curve, $C(u)$, using the location of the vertices on the edge, we define $u \in [u_{min}, u_{max}]$, which then is normalized for a more convenient notation as $u \in [0, 1]$. The parametric variable is a process known as direct point solution, which requires an evaluation of the polynomial. One straightforward method, which applies on vector polynomials, is Horner’s rule [3]. To evaluate the cubic polynomial given we get:

$$ C(u) = [(a + b)u + c]u + d $$

from which we calculate the position by only six operations (three additions and three multiplications).
and its normal vector as:

\[ n_s = \frac{\frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v}}{|\frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v}|} \]  

(9)

which we recall from Fig 1.

A quaternion based representation of the orientation is based on a unit vector and an angle of the rotation. By denoting the unit vector as \( \hat{k} \) and the rotation scalar as \( \vartheta \), we get the quaternion vector \( q = \langle \eta, \epsilon \rangle \), where:

\[
\eta = \cos \frac{\vartheta}{2} \\
\epsilon = \sin \frac{\vartheta}{2} \hat{k},
\]

By calculating the quaternion rotation vector using the surface normal \( n_s \) and the tangential vector of the curve \( t_c \), as

\[
t_c = \frac{dC(u)}{du} = \left( \frac{dx(u)}{du}, \frac{dy(u)}{du}, \frac{dz(u)}{du} \right)
\]

we form the rotation vector as:

\[
\hat{k} = t_c \times n_s
\]

(12)

Considering the \( \hat{k} \) vector being the bi-normal vector in a Frenet-frame [5], we now apply the rotation. By introducing a tilt parameter, \( \alpha_t \), in FSW normally a few degrees, and the 180 degrees needed to flip the tool (pointing in the opposite direction than the surface), we get:

\[ \vartheta = 180 - \alpha_t \]  

(13)

from which we form the rotation quaternion.

IV. EXPERIMENT

The experiment is based on using the CAD drawing of a machined object from which a highly defined path should be obtained. The path should be executed with a continuous motion of an ABB 7600-500 robot, specially designed for FSW [12].

A. Path extraction

The object, which have been created in a standard CAD system and stored as a Step [6] model, is loaded onto the path planning system’s 3D context. In Fig 5 we see the selection of a face (highlighted in blue) from which we create a matching circular segment. To complete the path, we continue by selecting the next face and edge throughout the complete object.

By the use of a local coordinate system in the simulation, it is not necessary to do adjustments in the actual
prone to failure in uprise the location of the local coordinate system of the object. When dealing with 3D objects, the orientation of the object is as important as the location. A slight rotation will cause an erroneous path. By using a pre-defined fixture for the object, with a known location and orientation, the localization becomes less demanding.

As shown in Fig 7, the welding operation on the created path is executed successfully. By the use of a CAD model, we can extract geometrical information to perform complex FS welding with small efforts.

V. CONCLUSIONS

Off-line programming can be a time consuming operation, while still not eliminating all difficulties. Never the less, the use of CAD models to create complex paths, is probably the only approach possible, as human assistance is needed to identify and/or confirm the joint to weld.

In the approach suggested we use the topological entities in the CAD drawing from which we define the path segments and control points. The OLP tool, in which these features are implemented, is capable of simulating the path and visualize it.

In this OLP tool the focus is neither on the manipulator’s motion nor collision detection. We therefore do not simulate the manipulator or validate the path from a robot motion point of view. The disregarding of manipulator kinematics and dynamics is in this case deliberate. Of course, it is not for sure that the path can be executed since no kinematic evaluation/simulation is performed, but the emphasis here is rather on path modelling from a geometric point of view than simulation of a robot station.

Based on the experiments performed on the implementation, we state the following benefits with this tool:

- Low efforts to create complex FSW paths.
- High path accuracy.
- Tool path simulation

However, the lack of kinematic modelling states uncertainties in the execution phase. This issue can of course be overcomed by implementing such in the OLP system, but it should not overtake the computational capabilities of the system, which should rather be emphasised on modelling of the tool/object contact.

ACKNOWLEDGMENT

The first author’s research is financed by the KK-fondation and ESAB AB Welding Equipment. Their support is highly appreciated. The first author is grateful for the technical support provided by the project members ESAB AB Welding Equipment, ABB Robotics and the research centre for Applied Autonomous Sensor System (AASS) at Örebro University.

REFERENCES


