1-1-2011

Offline programming for short batch robotic welding

Nathan Larkin
University of Wollongong, nlarkin@uow.edu.au

Aleksandar Milojevic
Kranendonk Production Systems

Zengxi Pan
University of Wollongong, zengxi@uow.edu.au

Joseph Polden
University of Wollongong, jpolden@uow.edu.au

John Norrish
University of Wollongong, johnn@uow.edu.au

Recommended Citation

Larkin, Nathan; Milojevic, Aleksandar; Pan, Zengxi; Polden, Joseph; and Norrish, John, "Offline programming for short batch robotic welding" (2011). *Faculty of Engineering and Information Sciences - Papers: Part A*. 551.
https://ro.uow.edu.au/eispapers/551

Offline programming for short batch robotic welding

Abstract

Robotics based automation has many advantages over both manual operation and dedicated hard automation. In practice however the robot programming overhead and high cost of dedicated jigging are serious impediments to its use for short production runs. Offline programming packages offer an attractive alternative to reduce programming time and robot down time, however many of the available packages are fairly complex and treat the geometric programming and welding operation separately. The paper will describe the development of alternative approaches and the evaluation of an integrated offline robotic welding package (RinasWeld) which addresses these issues and makes low volume robotic welding of complex fabrications viable.

Keywords

programming, offline, short, robotic, batch, welding

Disciplines

Engineering | Science and Technology Studies

Publication Details

Larkin, N., Milojevic, A., Pan, Z., Polden, J. & Norrish, J. (2011). Offline programming for short batch robotic welding. 16th Joining of Materials (JOM) conference 2012 (pp. 1-6).

Offline Programming for Short Batch Robotic Welding

Nathan Larkin[1], Aleksandar Milojevic[2], Zengxi Pan[1], Joseph Polden[1] John Norrish[1]
1. University of Wollongong, Wollongong, NSW, 2530, Australia
2. Kranendonk Production Systems, 10 Sir Rowland Hillstraat Tiel, JT, 4004, Holland

Abstract: Robotics based automation has many advantages over both manual operation and dedicated hard automation. In practice however the robot programming overhead and high cost of dedicated jigging are serious impediments to its use for short production runs. Offline programming packages offer an attractive alternative to reduce programming time and robot down time, however many of the available packages are fairly complex and treat the geometric programming and welding operation separately. The paper will describe the development of alternative approaches and the evaluation of an integrated offline robotic welding package (RinasWeld) which addresses these issues and makes low volume robotic welding of complex fabrications viable.

1. Introduction

Compared to manual welding or automated welding using dedicated machines, robotic welding has many advantages; Robotic welding is highly flexible, it achieves consistent welding results, and heat input can be precisely controlled. In practice however, the programming overhead and cost of dedicated jigging limit the viability of a robotic welding system for all but the most repetitive tasks. In some cases, the programming time can be 360 times the execution time [1].

For industrial welding applications, there are two main programming methods; online programming, where a teach pendant is used to move the robot to the required position and orientation for each step of the program and Offline Programming (OLP), where the program is developed in a 3D computer environment.

The concept of the online programming method is simple, however it is requires significant time and effort to generate robot programs for even the simplest processes and geometries. The OLP process is more complex than online programming methods and typically does not significantly reduce the programming time. The main advantages of the OLP method result from freeing the work cell for production while programming is taking place on a computer system. The common OLP packages are Not designed for single or low volume production and the high cost of proprietary OLP software is normally only justified for mass production applications.

This paper describes the issues facing the current generation of OLP software packages for use in programming robots and investigates some of the possible solutions available to improve the viability of robotic welding for low production volumes. An evaluation of several OLP software packages is provided. Common packages such as RobotStudio™ from ABB Robotics and Delmia™ from Dassault Systems are compared, along with a Matlab™ based OLP system. RinasWeld™ from Kranendonk Production Systems, a state-of-the-art OLP software package developed for single piece production in the shipbuilding industry is also dicussed.

2. The OLP process

The OLP process can be broken down into several steps:

1. 3D computer model generation. Typically generated during the design stages of a product.
2. Tag generation, where process start and end points are identified.
3. Path planning, where robot motion paths are planned and the reachability and potential for collision of a robot is assessed.
4. Process planning, where each individual process is sequenced and optimized.
5. Post processing, where the required process I/O is added and the program is converted into the native programming language of the robot.

In addition to these steps, the program will usually require calibration to account for differences in the real world geometry of elements compared to the nominal geometry of the elements modelled in the software package. OLP packages often offer a simulation system, where programs can be simulated to verify them before execution on the physical robot system. A block diagram of the typical OLP process is shown in Figure 1.

In most OLP software packages that are currently available, a robot programmer completes each of these steps manually, with perhaps a small amount of computer assistance. For example, tags can be generated along a complex path by selecting the modelled geometry, rather than being generated individually. To allow cost effective robot programming for low production runs, these manual steps need to be automated wherever possible.

2.1 Tag Generation

In robotic welding, the tag generation can be as simple as identifying geometrical elements that are in contact and defining the start and end locations as process start and end tags. However, since several welding process variables are linked to geometry, it is convenient for the tags to contain extra geometric information. This tag data can then be used at a later stage to automatically generate optimal process variables. For example, if the plate

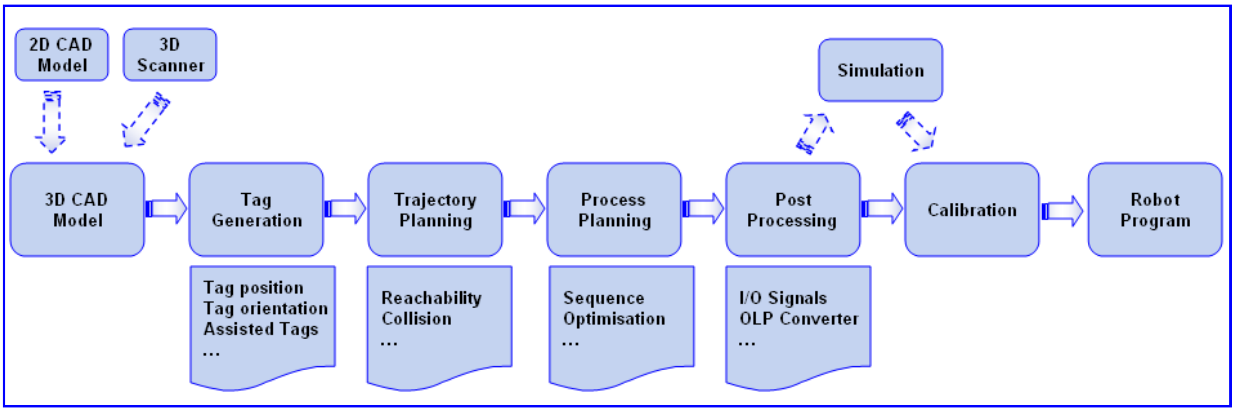

Figure 1: Block diagram of the OLP process.

thickness and material is known, welding speed, arc voltage and wire feed rate for a Gas Metal Arc Welding (GMAW) process can be generated.

Tags are also required for weld position calibration. Position calibration is important in robotic welding. A small error between the robot path and intended weld location can lead to poor weld quality. Typically, robotic welding systems utilise touch sensing capability built into modern welding power sources, or a structured light 2-dimensional profile sensor to correct such positional errors.

2.2 Path Planning

Path planning, involves determining the paths that the robot will take to get from each process point to the next. This process takes into account robot reachability, robot configuration, and collision avoidance to generate robot position steps that will lead the robot to weld start and end points, any calibration points required, and other locations in the cell such as wire cutting or touch cleaning stations.

2.3 Process Planning

In welding, the order in which each weld is completed is important to optimise robot utilisiation while adhering to weld procedure specifications. In some cases it may be acceptable to simply order welds arbitrarily. However for preheat or interpass temperature control, and sequencing with other process activities, it may be necessary to order welds accordingly.

2.4 Post Processing

Once the welds are identified and the motions and processes are planned, a post processing step is completed. This involves adding the required robot I/O, such as communication to welding power sources, and conversion to the robots native language. With the exception of selecting weld parameters, the post processing step is completed with little user input in most OLP software available.

2.5 Gaps in OLP Software

The current generation of OLP is capable of completing the OLP process defined above. However, the user input required limits the viability of this programming method for low volume part production. Minimising the user input, or programming time, required by OLP software is crucial if smaller batch sizes are to be considered. Of the OLP process steps described, the tag generation and motion planning steps are the most suitable for automation. The selection of weld process parameters during post processing can also be automated, provided weld geometry is known. Process planning requires detailed knowledge of weld procedures that are not typically handled by OLP software.

Delmia™ provides several tools that allow efficient tag generation of weld seams. However, each seam requires several operator inputs and can only be programmed individually. Whilst this facility reduces the programming time, it is not ideal for low volume production. RobotStudio™ has similar tools that allow paths to be generated from geometrical features in the model. The tags generated in these programs do not contain weld geometry information, and weld settings need to be generated manually.

In Delmia™ and RobotStudio™, path planning requires the manual placement of fly-by points to ensure collision free paths between processing points. For robots with external axes, Delmia™ offers a tool to allow optimal robot placement for reachability of weld seams. Like the tag generation tools, this requires additional user input.

3. OLP Solutions for Low Volume Production

3.1 Customisation of an OLP package

Some OLP software packages provide the ability to write software scripts to automate functions within the package. Delmia™ is an example of an OLP software package with this functionality. This allows a level of software customisation to minimise the required user input for each step of the OLP process. These facilities have been used in the current work to assess the feasibility of reducing programming time.

To generate tags for process start and end locations, plates in the model were checked for intersections. A surface was then generated from the intersecting area, and the perimeter edges treated as weld locations. An analysis of the direction normal to the plates, and the weld edge direction yielded the orientation of the tags. This process is shown in Figure 2**Error! Reference source not found.**.

Path planning was treated as two separate steps. The first step was to determine if the process path is reachable by the robot, in order to select an appropriate robot configuration and, the position of external axes. The second step was to plan the motion path between each of the process paths.

The first step to check reachability, configuration, and select the position of external axes, was completed by generating small robot programs that executed the weld path, as shown in Figure 3. During execution of the program, Delmia™ is able to report the reachability of the weld path and if the robot is in collision. Based on this output, the robot configuration and external axis position is modified until suitable positional parameters are found. On investigation it was found to be too difficult to implement this step with the scripting functions available in Delmia™. However, a third party software toolbox, the Kineo™ Robotic Path Planner, was found for Delmia™ that was able to generate the paths required.

Although this investigation indicated the potential to reduce programming time by customising the Delmia™ OLP software package, many limitations were encountered. In particular, the processing speed in Delmia™ was deemed to be too slow, possibly due to the generation of simulation graphics during program execution. Various software ‘bugs’ such as the robot ‘disappearing’, also led to incorrect and inconsistent results. Delmia™ is a powerful OLP software package, however, at this stage it was not considered suitable for the envisaged low volume production.

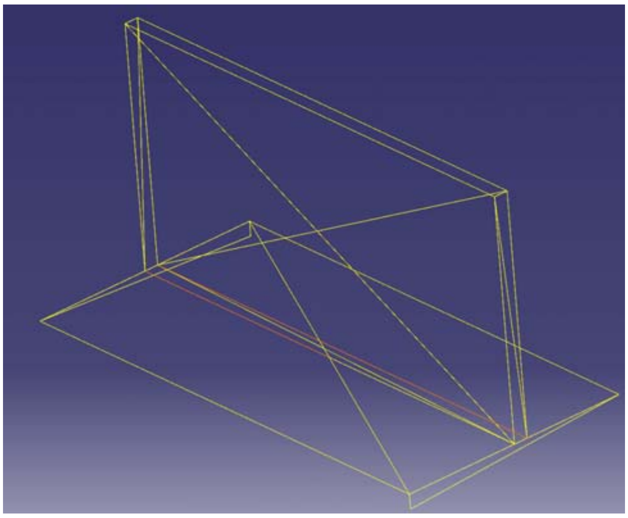

Figure 2: Plate Analysis for Tag Generation in Delmia.

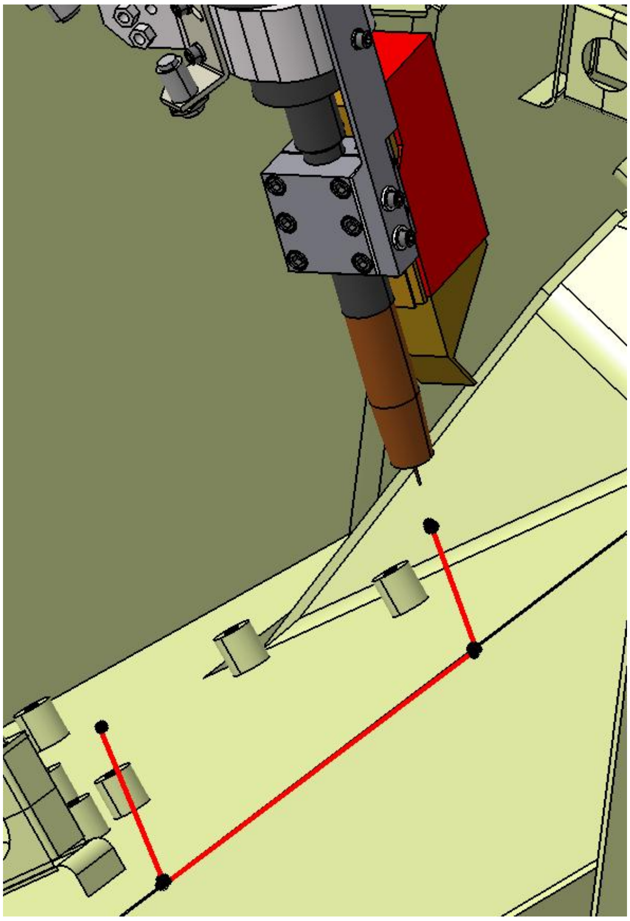

Figure 3: Delmia assessing the rechability of a weld seam.

3.2 Development of an Automated OLP System

A key requirement of an OLP software solution for low volume production is to limit the amount of user input required. Typical OLP software relies on a 3-dimensional computer modelled environment for user input and feedback. In fact, such an interface would not be necessary for low volume production and a simplistic OLP system can be developed without the computational overhead that is required to generate the visual environment. As automated OLP is calculation intensive, the Matlab™ software package was chosen as a suitable platform for an alternative system.

The automated OLP system has two key low level components on which the higher level algorithms are built upon. These are; the robot kinematic model which is used to calculate the various joint angles for a given robot position and orientation; and, the collision model which determines if the robot components are in collision with the other workcell components. Calculation time is an important consideration for an automated OLP system, particularly for the collision model. To check a weld path, the collision model can be run up to ten thousand times for a simple structure or more than one million times, for a complex structure. To accommodate this, the robots in the collision model were represented by a collection of spheres in order to minimise the complexity of the collision calculation as shown in **Error! Reference source not found.**.

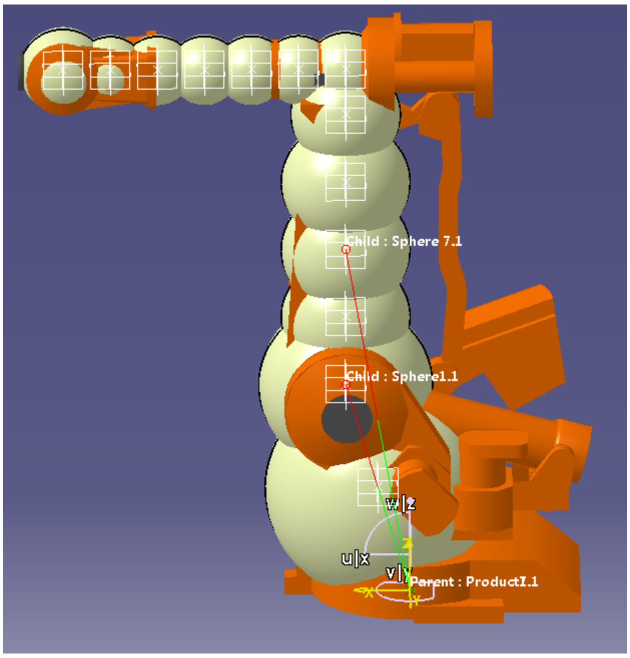

Figure 4: Sphere Model of an ABB IRB4400 Robot.

Weld seams were identified in the Matlab™ system by searching for plate edges that align with other plate surfaces. The start and end locations of the intersection edges along with the normal direction to the surface were used for tag placement and orientation. Once the weld tags were generated calibration points were added along the weld direction, and in the case of a corner, on other required geometries.

Linear paths between the weld and calibration points were then checked, and optimal robot configuration and external axes position selected. Points were also added using a probabilistic Roadmap (PRM) planner for robot paths that are difficult to access. **Error! Reference source not found.** shows a robotic path planned using the PRM planner.

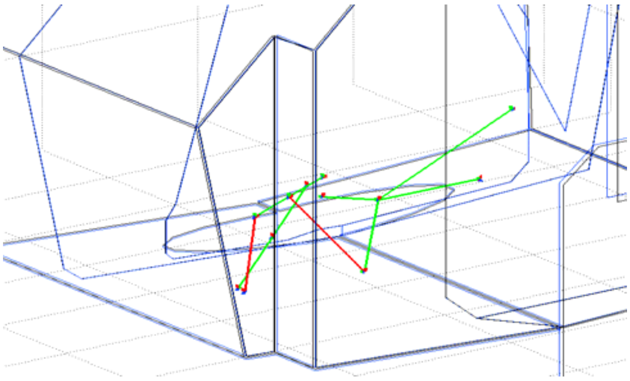

Figure 5: Path Planned using a PRM

After the robot motions are planned, the system converts the generated paths into code that can be executed by the robot controller. I/O is also added for the ancillary functions that are required by welding, torch cleaning and calibration systems.

This system is very effective at producing robotic welding programs. It takes approximately 1 minute of user time to produce a weld seam for a complex structure, this compares to an estimated time of 60 minutes using conventional online programming techniques. However, it should be noted that this system was designed for a specific robotic welding application. Further development is required to make it suitable for programming a generic robotic welding system.

3.3 RinasWeld

RinasWeld™, developed by Kranendonk Productions Systems (KPS), is an OLP software package developed originally for the shipbuilding industry. It is able to autonomously generate robot welding programs directly from CAD information with only a small amount of user input. The software identifies weld seams and generates weld tags and touch sense calibration points as required. It also plans collision free motion paths, and selects optimal robot placement within the external axes.

Downhand, vertical and overhead welds are identified and can be handled differently, along with intricate weld path details such as those required around plate edges. RinasWeld also links weld geometry with weld settings, further streamlining the OLP process.

As RinasWeld has been developed specifically for industrial applications, it differs from existing OLP software. Robot type(s), position, configuration of external axis, and tooling are all 'hard coded' into a specific version of the software supplied for the end user. Although RinasWeld has some software limitations around workcell customisation, the OLP functionality is well developed, and is the only commercially available software currently suitable for generating robot programs specifically for low volume production. Kranendonk claims that each hour of software use can generate 15 hours of robotic weld programs [3].

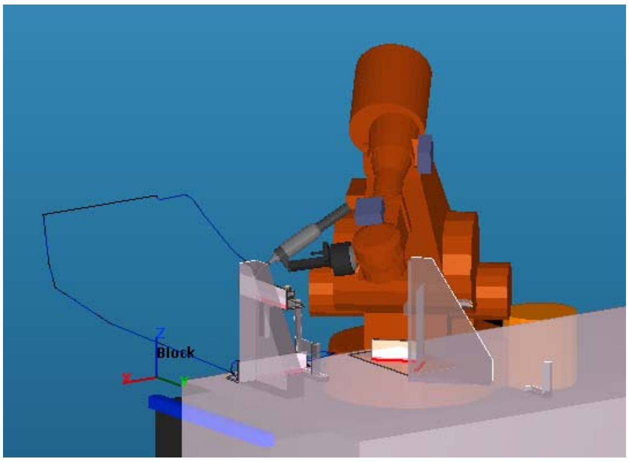

Figure 6: Simulation of a weld Program in RinasWeld.

4. Application Examples

4.1 Welding Research

Welding research requires the precise control of weld parameters and consistent, repeatable welds to ensure optimal results. Although this can be obtained using hard automation, the flexibility offered by a robotic manipulator significantly improves welding capability. Although programmed weld paths can be repeated several times, weld lengths are short compared to industry and programming time is a significant overhead. Also, conventional robot programming techniques limit the utilisation of the facility to only those with robot programming skills.

Automated OLP software reduces the programming time to a few minutes per weld path and unlike online programming methods complex geometries and difficult to navigate obstacles do not add significant programming time. Figure 7 shows a simulation of a robot navigating around a clamped plate for welding. OLP software, such as RinasWeld™ has now been applied to many welding research tasks at the University of Wollongong. It has been applied to fabrication of test samples for metallurgical analysis, temperature profile measurements on welded plates and weldability studies on coated steels. It allows researchers not skilled in robot programming to create usable robot programs.

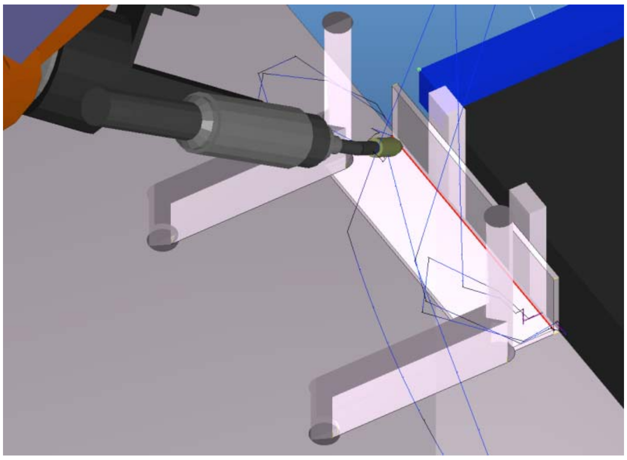

Figure 7 Robot Path Navigating Around Clamp Obstacle

To create a program, the parts to be welded are first clamped into position using a flexible fixturing system. Once the parts and in position, the fixturing and parts are modelled in a CAD software package. Each fixture comes from a standard library of parts, so this is a simple process. After the work environment has been modelled, it is imported into the automated OLP software package, where the robot program is generated.

4.2 Shipbuilding

. The large number of long and complex welds required in the shipbuilding industry are ideally suited to robotic welding, however the small production volumes limit the viability of robotic production using conventional programming techniques. RinasWeld™ is an automated OLP package which was originally developed for shipbuilding applications..

Typical applications for RinasWeld™ in shipbuilding are the welding of webs, stiffeners panels through to double hull blocks. It supports various CAD formats to allow product models to be directly imported. As ships can be very large structures, the import function can automatically split the construction into smaller segments for further efficiency.

Along with the identification of and generation of weld start and end points, joint properties are also recorded. User configurable rules then allow certain welding parameters to be defined. For example the bead size of the weld can be set according to plate thickness, weld orientation and other variables. Once rules are set for one product, they can be reused directly on other products. In this way welding “knowledge” is accumulated in the software system allowing the automatic OLP process to be applied on variations of the original product. The RinasWeld™ system also has high level weld identification and manipulation features. In many cases, welds are identified which do not require robotic welding. For this purpose “assembly level” rules can be setup to identify which welds do not require robotic welding. Furthermore, manual tools allow the user to directly delete, cut, or trim individual welds. This eliminates the need to manipulate CAD drawings specifically for the OLP process, minimising the changes to current drafting practices.

In RinasWeld™ the work preparation can be carried out, and results saved, for multiple products and the robot program generation can be initiated as a batch process for a number of these products. This allows the user to prepare the work during normal working hours, and then let the computer perform the unattended robot program generation outside of working hours.

In shipbuilding, the products to be welded are defined many weeks and even months in advance. This allows preparation of robot programs for welding these products to happen well in advance of production, thus securing efficient production with high utilisation of the welding robots.

4.3 Vehicle Hull

Although heavy vehicle production is not typically low volume, automated OLP techniques can be applied to reduce programming time for product changes. An example of this is the production of a defence vehicle hull, where variant changes and customer specific options are regularly required. This workcell also has other unique challenges; the hull is tacked together and internal weld seams are only accessible through openings in the hull; a robot-on-robot manipulator with 12 degrees of freedom is utilised to access the weld locations.

In this case, the OLP software has several tasks:

1. Identify the optimal position of the large carrying robot so the small robot can reach the weld path without collision.
2. Adjust the torch orientation to avoid collision, for example into a corner.
3. Determine the configuration of the smaller welding robot for the weld path along with the calibration points.
4. Plan the motion path for the small robot to each of the required locations.
5. Plan the motion path for the carrying robot to drive get the welding robot inside the tacked hull.
6. Generate the robot code required to complete the weld.

Each of these tasks was completed using the Matlab based automated OLP software package described above.

Using this system, welds can be programmed in a few minutes compared to the several hours that it takes using online programming methods. Programs for new weld seams are also easily implemented into the existing code structure where small modifications have taken place.

5. Conclusion

Utilising robotics for welding has some clear advantages. However, for short batch runs, conventional programming techniques make robotic welding an unviable option. One solution is the automation of the OLP process, such that robotic weld programs can be generated directly from a CAD model with minimal human input.

Three approaches to automating the OLP have been presented. Adding automation scripts to an existing OLP software package, developing a Matlab™ based automated OLP software system, and using a commercially available automated OLP system. Although there were problems automating an existing OLP software package, the Matlab™ based system and the commercial system were able to generate robot programs cost effectively for low production volume.

The application examples show how these software tools are being used for research, in shipbuilding, and in defence vehicle fabrication. These examples demonstrate the effectiveness of automated OLP techniques for the robotic welding process.

6. Acknowledgement

The authors acknowledge the support of the Defence Materials Technology Centre, which was established and is supported by the Australian Government’s Defence Future Capability Technology Centre (DFCTC) initiative. In addition thanks are due to DMTC partners Thales and RTA. Assistance in providing training and an evaluation licence for RinasWeld™ from Kranendonk is also acknowledged.

7. References

[1] Joseph Polden, Zengxi Pan, Nathan Larkin, Stephen Van Duin, John Norrish, “Offline programming using DELMIA Automation”, 2010 International Conference on Robotic Welding, Intelligence and Automation (RWIA'2010), Oct. 14-16, 2010, Shanghai, P. R. China

[2] Zengxi Pan, Joseph Polden, Nathan Larkin, Stephen Van Duin, John Norrish, “Recent Progress on Programming Methods for Industrial Robots”, ISR/ROBOTIK, Munich, Germany, June 7-9, 2010.

[3] Anonymous. (2009). "Panel & Web Welding Gantry." Retrieved 15/12/2009, 2009, from www.kranendonk.com.