TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics and Mechanics
Vol. 54, Issue I, 2011

OFFLINE PROGRAMMING OF ROBOTIC ARC WELDING SYSTEMS

Bogdan MOCAN, Mircea FULEA

Abstract: *The online programming time for industrial robots in many cases represented increased costs and downtime, which affected the manufacturer's profitability. The efficiency of the offline programming for arc welding robots depends on the complexity of work pieces and welding experience of the technician. While envisaging CAD/CAR/CAM integration, this paper reviews the current state of the art in offline programming of robotic arc welding systems.*

Key words: *robot programming, offline programming; arc welding, industrial robots.*

1. INTRODUCTION

Robotics has been an active area of research for more than three decades. Today various types of robots are in use in industry, in particular for manufacturing applications. Robots are complex machines and significant technical knowledge and skills are needed to control them [1], [2].

The range of applications of industrial robots has expanded in recent years also due to advances in programming capabilities resulting from the development of offline programming (OLP) systems [2], [3], which enable program development to take place in a virtual environment.

Due to the highly proprietary nature of robot software, most manufacturers of robot hardware also provide their own software. While this is not unusual in other automated control systems, the lack of standardization of programming methods for robots does pose certain challenges [4]. For example, there are over 30 different manufacturers of industrial robots, so there are also 30 different robot programming languages required. Fortunately, there are enough similarities between the different robots so that it is possible to gain a common understanding of robot programming without having to learn each manufacturer's proprietary language [5].

Robot software consists of the instructions that control a robot's actions and provide information regarding required tasks. When a program is written using this software, the robot is able to execute commands and perform tasks [6].

Programming robots can be a complex and challenging process, and while it has become easier over the years, the lack of cross-platform industry standards has affected the development of software tools for robots compared to other automated control systems such as programmable logic controllers (PLCs) [4].

The offline programming software applications allow manufacturers to program their next job without interrupting the one currently in progress; thus, they start to attract considerable attention from both large and small companies. In some cases, the time in which products reach the market is reduced by 50% and costs are cut by up to 30 percent, according to customers [3].

This paper reviews the current state of the art in offline robot programming systems, in the related area of arc welding robot software applications and related trends.

We do not aim to enumerate all the existing offline robot programming systems; this paper focuses only on the most used offline robot programming systems in Romania (ABB’s RobotStudio, Kuka’s Sim Pro, Fanuc’s Roboguide).

2. BASIC CATEGORIES OF PROGRAMMING LANGUAGES

Virtually all robots are programmed with some kind of robot programming language. These programming languages are used to command the robot to move to certain locations, signal outputs, and read inputs. The programming language is what gives robots flexibility. When learning any programming language, whether a robot or a computer language, one of the most difficult tasks is learning what the commands are and how to use them [7].

To get an overview of different types of robot programming languages, it is appropriate to classify them in three basic categories [4]:

1. Specialized robot languages. These languages have been developed specifically for robots. The commands they provide are mostly motion commands with minimal logic statements available. Most of the early robot languages were of this type and many still exist today. VAL is an example of such a robot language [2], [4], [8].
2. Robot libraries for a new general-purpose language. A new general-purpose programming language was first created and then robot-specific commands were added to it. They are generally more capable than a specialized language, since they tend to have better logic testing capabilities. KAREL is an example of this type [8].
3. Robot libraries for an existing computer language. These languages are developed by creating extensions to already existing popular computer programming languages.

Consequently, the robot languages resemble traditional computer programming languages, providing the same power as these widely used languages [7].

Regarding programming techniques, today, arc welding robots are programmed in one of two possible ways. In reality, these techniques are often combined, which sometimes is referred to as hybrid programming. The two main techniques are described shortly below [9].

Online programming means creating the program directly on the robots teach pendant (such as the one shown in figure 1), hence by manually steering the robot to different positions using a jog movement or similar control mechanism. Each desired position contributes to the code as a number of coordinates. An advantage with online programming is exactness and few later corrections due to programming the actual robot in its actual real-world environment. As disadvantages, time consuming and long production stops are mentioned.

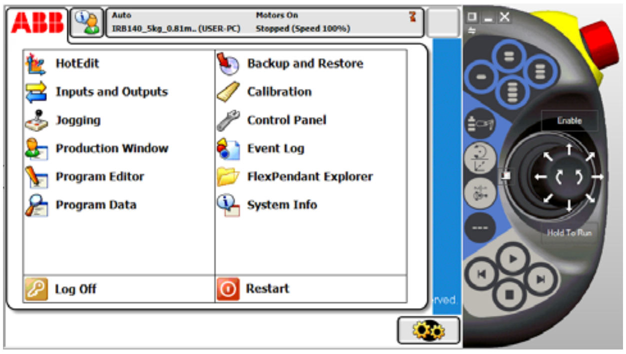

Fig. 1. An ABB Robot Teach Pendant [3].

In contrast to online programming, *offline programming* means creating the control program on a detached unit, such as a PC. This involves either manual editing of code in a text editor, or automatically generated code using for instance a CAD - model in RobotStudio [3] (as seen in figure 2) or corresponding environments.

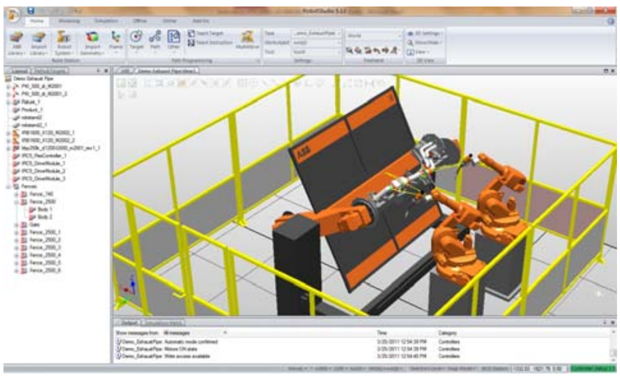

Fig. 2. A virtual robotic arc welding cell realized in RobotStudio 5.13 [3].

Once the program is ready for deployment, it is moved to the robot's computer for manual correction and tuning [3], [11].

An advantage of this method is that robots can be programmed before installation and kept in production while being reprogrammed. This means that production breaks are usually significantly shortened. On the other hand, manual correction sometimes gets very extensive, and a programmer is also required to write the code offline [2].

3. ARC WELDING DATA

Welding data consists of parameters that are used for the welding process. The welding data is composed of start data, main data, end data, and weaving data. Figure 3 shows an example of welding data [6], [12].

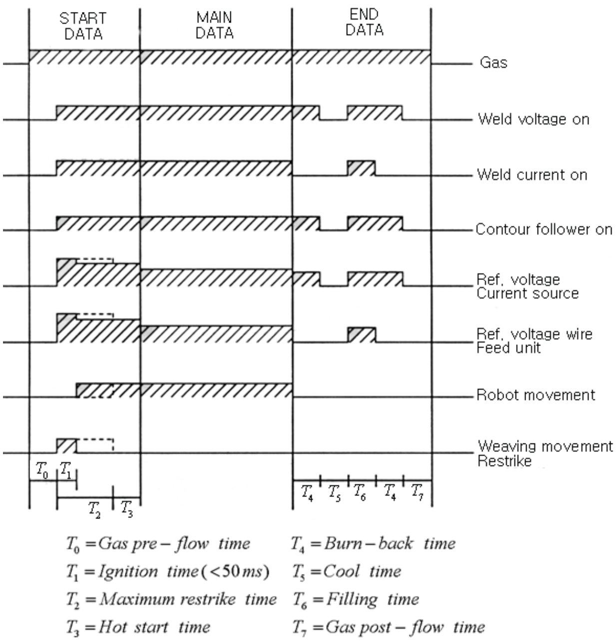

Fig. 3. An example of welding data [12].

Start data. Start data generates arc start data and stabilizes electric power. Start data contains the following parameters [2]:

- Ignition voltage and current.
- Gas pre-flow time - the time between shield gas flow and arc start.
- Hot start voltage and current - to stabilize the arc in the first stage.
- Restrike amplitude - minor change in the position of the torch to start the arc.
- Restrike cross time - time to stay in the restrike position.
- Arc welding start maximum time - the robot stops the process if the arc does not start in this time interval.

Main data. The Main data contains the following welding process parameters [6]:

- Welding voltage;
- Welding current;
- Welding speed.

For higher productivity, the welding speed should be increased to the maximum value.
Therefore, a new system should be put through a number of tryouts until the parameters for maximum speed are determined. The above three parameters have interrelations with each other.
End data. At the end of the welding process there are craters, or cracks, that may be the cause of welding defects. Therefore, several parameters for appropriate finishing are required.
These parameters include [7]:

- End voltage and current;
- Gas postflow time;
- Burnback time;
- Cool time;
- Fill time.

Weaving data. A large work piece, to be welded with large penetration repetitive welding, demands a long time and requires a motion plan to the weld start position. Carrying out this type of welding in one pass is possible due to weaving. Weaving has various patterns such as zig-zag, V-shape, triangular, and wrist weaving. Wrist weaving uses only the sixth-axis of the robot and enables weaving in a narrow space where weaving with the lower arms of the robot is impossible. Also, it is useful when high frequency weaving is necessary. Therefore, small amplitude weaving is recommended [4].

4. OFFLINE ROBOT PROGRAMMING SOFTWARE APPLICATIONS

In this the study we will consider the most used offline programming software applications in Romania (i.e. RobotStudio, Kuka Sim Pro, and Roboguide).
RobotStudio is built on the ABB VirtualController, an exact copy of the real software that runs on ABB robots in production. It thus allows very realistic simulations to be performed, using real robot

programs and configuration files identical to those used on the shop floor [3].

ArcWelding PowerPac is a dedicated programming tool for generating arc weld programs and is based on RobotStudio [5]. Both of them use the CAD geometry as the basis for all robotics programming.

The programmer defines the weld locations in the CAD geometry (Fig. 4) and creates robot positions in relation to the geometry. This method is known as geometry-based offline programming [11].

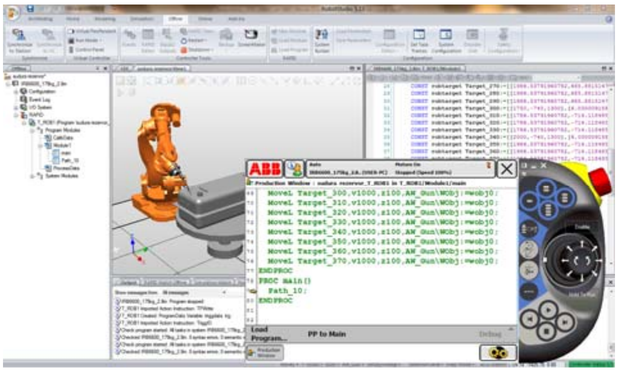

Fig. 4. A capture of Robotstudio v. 513 application software [3].

To create a weld, the user starts by picking the start and the end of the weld on the geometry. The next step is to define the process parameters such as Torch Angles – Work Angle, Push/Drag and Spin angles, Weld Parameters – Seam, Weld and Weave data, Motion Parameters – Speed, Zone, Motion Type – Linear, Circular and Joint motion, Instruction Type – Move, Arc, Search [3].

The arc welding data that is to be used needs to be defined as well. This data is divided into three types [9]:

- seamdata: describes how the seam is to be started and ended.
- welddata: describes the actual welding phase.
- weavedata: describes how any weaving is to be carried out.

The exact components of the above data depend on the configuration of the robot at the time [3].

The arc welding instructions will be added directly to the program. The structure of an arc welding instruction in RAPID, the ABB robot's programming language, is illustrated in figure 5 [3], [11], [2].

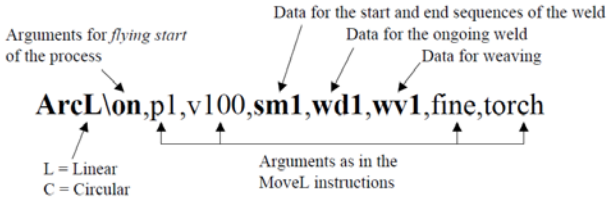

Fig. 5. The arc welding instructions in RAPID [3].

The arguments are set in relation to the last arc welding instruction that was programmed. However, if an argument needs to be changed, it can be replaced by another one.

When finished adding arc welding instructions, it is time to go on with the arc welding topics. The topics contain parameters that define the arc welding functions [11]:

- The units used when the parameters are entered;
- The process functions used;
- The current equipment;
- The weld guide sensor being used.

When the setup is complete, and the program is running, there are two ways of tuning the weld data components.

Certain weld data components (*weld_speed*, *weld_wirefeed* and *weld_voltage*) can be tuned using the weld data tuning function. When tuning this way, it is always the present value that is changed, however the original value can also be updated (i.e. it can be set to the same value as the present value). Certain data can also be tuned while it is active (i.e. when the program is executing), however only the present values can be tuned. The original values can be altered only when program execution has been stopped [11].

KUKA Sim Pro is designed for offline programming of KUKA robots. It allows cycle time analysis and robot program generation and is connected in real time to KUKA OfficeLite (Fig. 6), the virtual controller of KUKA's robot [10].

KUKA.ArcTech PC is an extension to the KUKA SIM Pro solution for the offline programming of arc welding robots. Using KUKA.ArcTech PC the robot motions for the welding seams are generated. The robot motions can be verified for robot reachability and collisions.

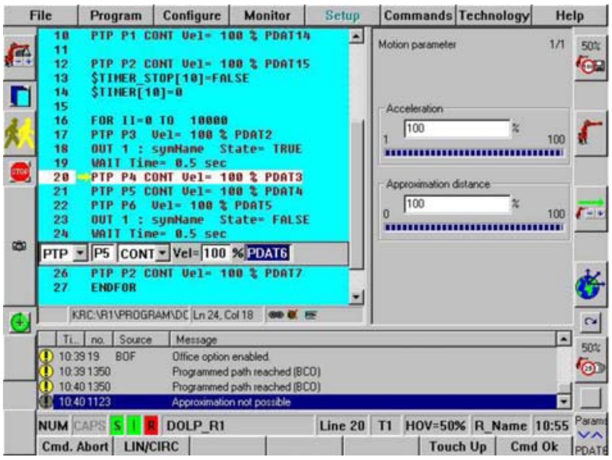

Fig. 6. The KUKA programming environment and robot programming language [10].

Once the programs are validated, the robot program is generated and transferred to the robot controller [1].

WeldPRO, FANUC Robotics' latest plug-in to the Roboguide® offline programming tool, allows users to simulate a robotic arc welding process in 3-D space [8]. Driven exclusively by a FANUC Robotics Virtual Robot Controller, WeldPRO is empowered with the most accurate program teaching tools and cycle time information available in any simulation package [8].

A user can easily navigate through WeldPRO to create complete workcells by importing actual tooling and workpiece CAD files. Anyone familiar with programming a FANUC robot will be able to easily create new weld paths with proper torch angles and process parameters [2].

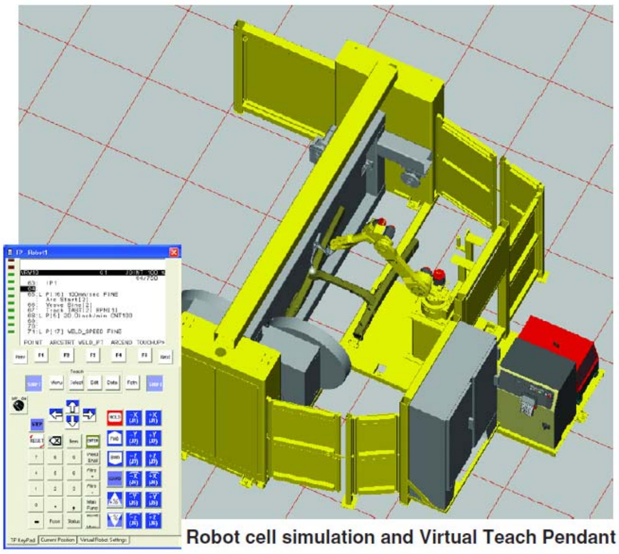

Fig. 7. A capture of the WeldPRO software [8].

All programs and settings from the virtual workcell can be transferred to the real robot to decrease installation time (as seen in figure 7).

WeldPRO is recommended for GMAW Welding, Plasma Welding, GTAW Welding and Plasma Cutting [8].

5. CONCLUSIONS

As robot usage increases and their applications and components to be produced become more sophisticated, it is essential to look beyond traditional robot programming method. The trend is that robot cells are simulated and programmed using robot simulation software.

The benefits of using such software applications are:

- robotic welding can be validated in a virtual environment without the time consuming and costly need to acquire an actual robot, its associated parts, tooling and welders;
- part and tooling CAD models produced with proven CAD applications can be imported into a simulation software
- offline "what if" scenario simulations are possible. Users can improve and touchup existing robotic applications without experiencing downtime and production loss;
- the most accurate cycle time information is provided, as compared to any other simulation packages available in the industry;
- Working with the Virtual Teach Pendant, operators can gain experience and training on an exact replica of their workcell, without any expensive hardware.

The Romanian industry needs to adopt such technologies (ABB's RobotStudio, Fanuc's RoboGuide, or KUKA's SimPro) in order to increase their manufacturing competitiveness.

6. REFERENCES

[1] Kitagishi, I., Machino, T., Nakayama, A., Iwaki, S. and Okudaira, M., *Development of motion data description language for robots based on extensible markup language – realization of better understanding and communication via networks*. In Intelligent Robots and System, IEEE/RSJ International

Conference on, volume 2, pages 1145-1151, 2002.

[2] Zhenyu, L., Wanghui, B., Jianrong, T., *Motion navigation for arc welding robots based on feature mapping in a simulation environment*, Robotics and Computer-Integrated Manufacturing, vol. 26, pp. 137-144, 2010.

[3] ABB, RobotStudio – Programming & Simulation. RobotStudio 5.13 product description, Hannover, 2000.

[4] Reinhart, G., Munzert, U., Vogl, W., *A programming system for robot-based remote-laser-welding with conventional optics*, CIRP Annals - Manufacturing Technology, vol. 57, pp. 37-40, 2008.

[5] Mocan B. and Brad, S., *Changing Paradigms in Robotic Arc Welding by Integrating the Lean Manufacturing Concept*, Academic Journal of Manufacturing Engineering, Vol. 7. pp. 78-83, 2009.

[6] Kim I.S., Son, K.J., Yang, Y.S., Yarlagada, P.K.D.V., *Sensitive analysis for process parameters in GMA welding processes using a factorial design method,* International Journal of Machine Tools and Manufacturing, vol. 43, pp. 763–769, 2003.

[7] Wenrui Dai and Kampker, M., *User oriented integration of sensor operations in an offline programming system for welding robots*. In Proceedings of the IEEE Intl. Conf. on Robotics and Automation (ICRA 00), volume 2, pp. 1563-1567, April 2000.

[8] Fanuc, *http://www.fanucrobotics.com*, March 2011.

[9] Elia E., *Robotic-arc-welding,* http://www.welding-advisers.com/Robotic-arc-welding.html, March 2011.

[10] Kuka, *http://www.kuka.com*, March 2011.

[11] ABB, *http://www.abb.com/*, March 2011.

[12] Grad L., Grum, J., Polajnar, I. and Slabe, J. M., *Feasibility study of acoustic signals for on-line monitoring in short circuit gas metal arc welding*, International Journal of Machine Tools and Manufacture. Elsevier Ltd, Vol. 44. pp. 555-561, April 2004.

Programarea Offline a Celulelor Robotizate de Sudare cu Arc Electric

Rezumat: *Timpul alocat programării online a roboţilor industriali, în multe cazuri, a reprezentat costuri crescute şi timpi neproductivi ridicaţi, care au afectat profitabilitatea firmei. Eficienţa programării offline a roboţilor implementaţi în procese de sudare cu arc electric depinde de complexitatea pieselor de sudat şi de experienţa specialistului în sudare. Având în vedere aplicaţiile software de modelare CAD, simulare a proceselor automatizate şi robotizate (CAM, CAR), lucrarea de faţă îşi propune să evidenţieze nivelul de dezvoltare a celor mai utilizate aplicaţii software de programare offline a roboţilor industriali implementaţi în procese de sudare cu arc electric.*

Cuvinte cheie*: programarea roboţilor, roboţi industriali, programarea offline a roboţilor industriali pentru sudare cu arc electric.*

Bogdan MOCAN, Assistant Professor, Technical University of Cluj-Napoca, Department of Production Equipments and Industrial Robotics, bogdan.mocan@muri.utcluj.ro.

Mircea FULEA, Researcher, Technical University of Cluj-Napoca, Department of Production Equipments and Industrial Robotics, mircea.fulea @staff.utcluj.ro.