Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
*Young Researches and Scientist Paper / * Supervisor, Mentor*

OFFLINE PROGRAMMING AND SIMULATION OF ARC WELDING ROBOTIC CELL USING ROBOTSTUDIO SOFTWARE

IVAN, A[ndrei] - M[ario]*; AVRAM, G[eorgia] - C[ezara]; NICOLESCU, A[drian] F[lorin] & STRAJESCU, E[ugen] R[adu]

Abstract: *The paper presents the offline programming and simulation approach for a medium scale robotic arc welding manufacturing cell. The works include the layout development of the manufacturing cell, the offline programming process of the articulated arm robot equipped with an arc welding torch, and the validation/optimization of the developed program using simulations in virtual environments. The study was made using ABB equipment and the ABB RobotStudio software program for offline programming and simulation.*
Key words: *industrial robot, arc welding, offline programming, virtual simulation, ABB RobotStudio*

1. INTRODUCTION

The works presented in this paper illustrates the possibilities and facilities of using offline programming and simulation software for robotic arc welding industrial purposes. In order to fully explore the advantages of this approach, the ABB RobotStudio software was used for the entire process, from cell layout design and application development up to robot offline programming, as well as program optimization and validation in the virtual environment provided by the software (Nicolescu, 2010).

2. THE CELL STRUCTURE

The cell layout includes the following elements (see Fig. 1):

- an articulated arm industrial robot equipped with an arc welding torch for performing arc welding operations (Nicolescu, 2005);
- four part positioners used for orienting the parts to be processed by the robot, two types of positioners being included in the cell structure (2 with one degree of freedom and 2 with two degrees of freedom for part orienting) each positioner having two workstations in order to reduce auxiliary part setting time;
- a linear tracking module in order to increase robot's work envelope and to ensure robot's access to all positioners;
- a torch management system used to identify and allow correction of any tool positioning and orientation errors.

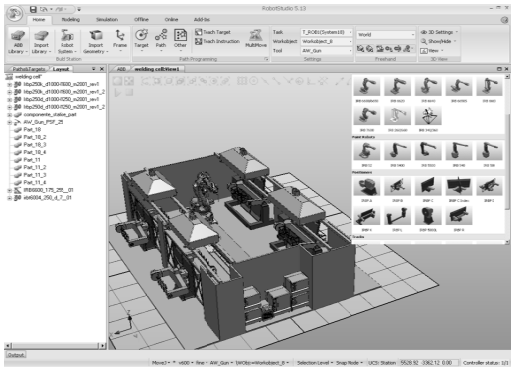

Fig. 1. Overall cell structure

Additionally, the cell includes one fume exhauster system above each positioner, in order to evacuate the gas resulted from welding operations.

The development of the station elements was partially made using Catia V5 virtual prototyping environment for the walls, the fume exhausters, the controllers, the welding source and the torch management system (Nicolescu, 2009). After these elements have been reciprocally constrained, the workspace and components distribution were validated using the DMU Kinematics menu, by defining the specific motion laws for end-effector trajectory generation and servo-assisted end-effector orientation motions. After workscene validation, the components have been imported in RobotStudio virtual environment, where the positioners, the linear track and the robot have been added, as well as the corresponding virtual controllers.

It is necessary to take into account that the manufacturing cell includes components that are moved by other components, as is the case for the robot, moved by the linear track and the parts, moved by the positioners. Thus, attachment relations were defined between these elements in order to ensure complete synchronization of different component movements.

3. DEVELOPMENT OF APPLICATION PROGRAM

After the workscene has been completely configured, a virtual system, representing the command unit of the station was created. Because there are four positioners included alongside the robot and the linear track (resulting far too many external axes for only one controller) it became necessary to assign multiple tasks for commanding all these units. Thus, the robot and the linear track were assigned (for optimal coordination) to one task and the positioners were assigned each to a different task.

Taking into account that all target points from which the paths will be interpolated are calculated and referenced with respect to the robot's baseframe, and that the robot is moved by the tracking system, the virtual controller should be able to read the position changes of the robot's baseframe. In order to ensure logical functionality of the cell (from the controller unit's point of view), the configuration files representing the virtual controller of the industrial robot-linear tracking system had to be modified so that robot's baseframe position can be updated in accordance with robot-linear tracking system movements. After solving the baseframe issues and setting the attachment relationship between the cell components, the programming sequences were defined.

Because, in order to be completely processed, the part had to be repeatedly repositioned, modular programming has proved to be the best approach. Initially, the parts to be processed were only visually represented in the virtual environment by the corresponding geometry, but they were not individually linked to a particular frame. The "Create Workobject" command allows the programmer to define a

particular frame for each processed part (***, 2007). This aspect is important from the controller's point of view, because the robot movements and the tool orientations are calculated with respect to both robots' baseframe and the frame attached to the part.

The path followed by the robot's tool characteristic point was defined by creating a set of targets. Any target has to be attached to a workobject. Knowing that the targets created to interpolate the desired path are positioned with respect to robot's baseframe, but are attached to the workobject, a different workobject was created after each reorientation of the processed part, due to the new position of its geometry.

Each set of targets corresponding to each workobject were grouped to form a programming module. To develop a programming module based on a set of targets, the following steps were followed:

- for each target a set of move parameters were created, thus resulting a move instruction, that included movement speed and accuracy, type of movement (joint, linear or circular), tool orientation and robot's configuration;
- the move instructions were grouped to form a path, that was interpolated based on the targets corresponding to each instruction;
- a set of signals were configured to ensure communication and feedback between the robot's controller and the other components of the cells (see Fig. 2). Particularly, two types of signals were needed for the developed application: digital output signals, used to command the repositioning of the processed part, and digital input signals, used to inform the robot's controller that the new position of the part is reached, and the welding process can be restored;
- a set of action instructions were created based on the signals and the occurring events in the station (such as reorienting a part), these being necessary for conditioning the robot actions by the different signal status.

After creating a target, which represents a specific position for the tool characteristic point, the tool orientation and the robot configuration for that target were specified (see Fig. 3 and Fig. 4). These settings are critical to avoid singularity situations, especially in the case of robot's configurations. The starting configuration is very important for subsequent movements, as it limits the options for the next target possible configurations, and this restriction can propagate to the point that there are no valid configurations for the next target of the path. Another important aspect of the configuration settings is task optimization, considering that useless robot reconfiguration between two targets increases process time (***, 2010).

After all paths have been configured, containing the necessary move and action instructions, the final program was created by synchronizing the paths with the virtual controller. This generated separate procedures corresponding to each path, which could be called by creating a main sequence of the program in the Simulation menu. The program can be debugged and edited in text mode using the Offline menu of the RobotStudio software (see Fig. 5).

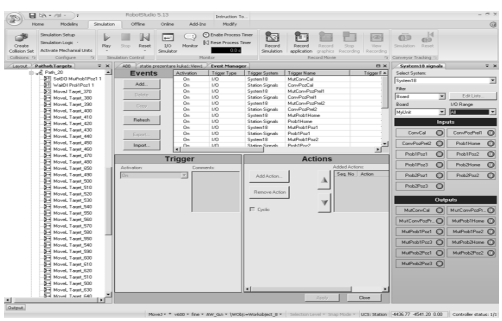

Fig. 2. Creating the station input and output signals and setting the correspondence to possible events

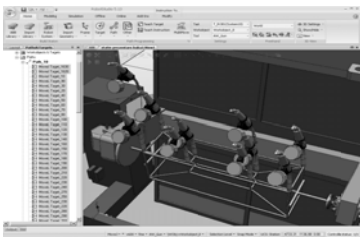

Fig. 3. Setting the tool orientations for each target of a path

Fig. 4. Setting the robot configuration for the first target of a path

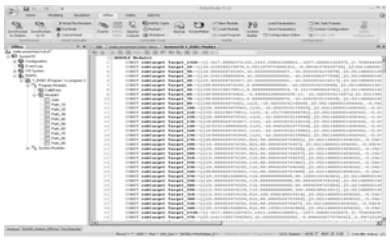

Fig. 5. The final program displayed in text mode

4. CONCLUSION

The works included in this paper represents a part of a larger project oriented towards optimization of industrial robotic applications based on off-line programming and simulation techniques. Present research explores the possibilities offered by RobotStudio offline programming and simulation software for robotic arc welding applications optimization, using block teaching and modular programming. Future research will be performed in order to continue optimization for industrial robotic assembling processes, the research strategy including applications and simulations performed with various software, in order to analyze and compare different software structures and programming approaches. After fully exploring the facilities offered by the analyzed software, methods of addressing some issues of robotic specific applications and optimizing the different processes will be elaborated. This stage of the research also includes validation of the simulation results using real robotic systems.

5. REFERENCES

Nicolescu, A. (2005). *Industrial Robots (in Romanian*), EDP Publishing House, Bucharest, Romania

Nicolescu, A. & Ivan, A. (2009). Robotic manufacturing cells virtual prototyping, *Proceedings of the 18th International Workshop on Robotics in Alpe-Adria-Danube Region*, May 25-27, Brasov, Romania

Nicolescu, A., Ivan, A. & Marinescu, D. (2010). Advanced Part Manufacturing using Kawasaki Robot and Off-Line Programming and Simulation software PC Roset, *Proceedings of CNC Technologies Workshop*, May 5-7, Bucharest, Romania

*** ABB (2007). *RobotStudio operation manual*

***(2010)http://www.abb.com/product/ap/seitp327/6230bcade7a9d7d8c12573f50042d0a9.aspx - ABB RobotStudio Community page, *Accessed on: 2010-05-10*