Description of manufacturing operations in 6-axis industrial robots through XML files.

A. Suelves(1), A.C. Majarena(2), J. Santolaria(3), J.A. Aguilar(4).

(1) Department of Design and Manufacturing Engineering, EINA. University of Zaragoza, 50018, Zaragoza (Spain)[arturo.suelves@gmail.com](mailto:arturo.suelves@gmail.com), (2, 3, 4) Department of Design and Manufacturing Engineering, EINA. University of Zaragoza, 50018, Zaragoza (Spain)

**ABSTRACT**

The aim of this work is to develop a methodology to define manufacturing operations in 6-axis industrial robots. This methodology is based on the use of XML files to describe the operations that the robot must perform. The generated XML file is processed in order to obtain the necessary robot programming to perform the task. The description of operations on XML files allows the user to better readability, portability and modifiability of operations.

**Keywords:** XML , manufacturing operations, robot.

**1. Introduction**

The aim of this paper is to describe manufacturing operations to be performed by an industrial robot using a descriptive file type that is easily manipulable and readable by the user.

Today the tasks performed by industrial robot are very different: welding, machining, laser cutting, [1], palletizing, etc...

A robot basically performs operations assigned using a series of commands which describe the paths to be performed. Each point of the trajectory is defined by the robot join coordinates. That is, the value that must have each joint of the robot to position its end-effector at that point. Once these positions are reached, the robot activates their tool under certain conditions [2].

Each production process has its own particularities. Thus, for example, spot welding orders sent to the end-effector of the robot when it has reached its goal would be: clamp activation, activation time, etc... For a machining process, the tool should be activated at one point and being active along a path with a certain speed.

That is why the file that describes the tasks must be flexible enough to be adapted to each type of automated process.

**2. Analysis**

To program a sequence of work of a robot, the user must program it with a series of points and orientations of the end-effector. The path through these points determines the trajectory of the robot tool.

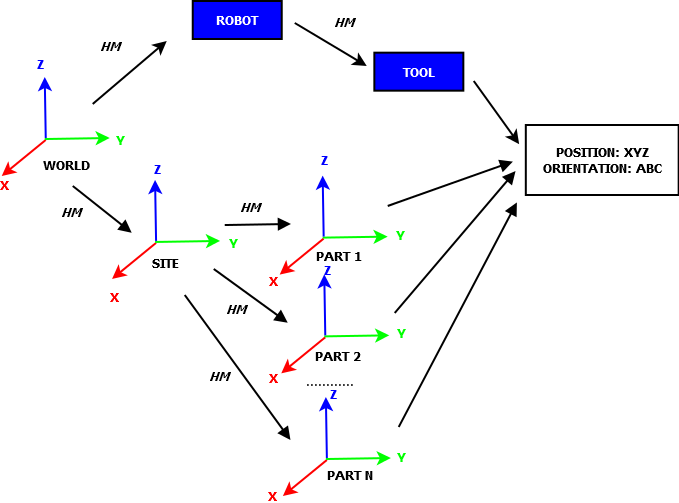
There are two basic types of displacement between two points [2]:

1. The trajectory point to point. (Point To Point. PTP). The objective is to minimize the time spent on performingthe move. The intermediate points do not matter in this type of displacement (this type must be used with caution to avoid collisions because waypoints are not defined).

2. The continuous path. (Continuous Path). It is the movement made following mathematical conditions; a line, arc, etc... In this case the intermediate points of the path are defined.

Once the end-effector is located in the desired position and orientation, the associated tool (welding, laser cutting, etc...) is actived for a particular condition (time, to follow a path, etc...).

The following scheme is proposed to describe the automated manufacturing process (see Figure 1).



***Figure 1.*** *Proposed scheme.*

The description of the operations is performed by change operations of reference systems through homogeneous matrix (HM). From the absolute reference, WORLD, to the desired robot end-effector. The bottom of the figure shows how to pass from a WORLD system (absolute) passing through a SITE system (workstation where the workpiece is placed) to the local systems each of the parts of the piece to be processed. To perform the process, the robot must reach the same position and orientation through its kinematic chain as can be seen at the top of the figure.

This description allows us to insert and modify operations and wean the process from the robot model. It also allows us to perform complex operations by means of its decomposition into simpler operations.

The file describing these tasks should indicate the type of process which is designed, as each production process has its own peculiarities.

The XML format is selected as it is a standard format, it has many tools for using and processing, it is independent of the platform used and the files are plain text files.

**3. XML language**

XML [3] is proposed as a standard for exchanging structured information between different platforms. XML acronym for eXtensible Markup Language, is a markup language developed by the World Wide Web Consortium (W3C) and it is used to exchange information and to store data in readable form.

It derives from SGML and allows us to define specific language grammar (in the same way that HTML is itself a plane defined by SGML) to structure large or complex documents.

The XML format is widely used in computer systems because it allows different systems to share information in a secure, reliable and agile way.

Some of its uses in the manufacturing industry are: information exchange between different CNC machines [4] and their different languages, data exchange between ERP and databases, process simulation, etc...

Some advantages of using this format are: syntax check to prevent typing errors (a file must be valid and well formed), insertion or modification operations can be performed easily, easily readable by the user, easy definition or modification of new elements (types of movements).

**4. Defining the file structure**

XML is standard, so its structure definition and validation rules must be performed by a file called XML SCHEMA. The elements, the syntax and the structure are defined in this file, as can be seen in Figure 2.



***Figure 2.****XMLSchema file structure.*

A root element, PROCESS, is defined. The process type, the units to use and the XMLSchema definition file of the structure and syntax are indicated in this element. Moreover,this element only contains GROUP and INCLUDE elements.

GROUP elements are defined to group in a logical way the elements of the file. They may contain a number of basic and complex elements.

INCLUDE elements are used to include GROUP elements of other files in the current file. So you can reuse definitions and other tasks defined in another XML file GROUP reducing the generation times.

Simple elements are the basic movement commands (TOPOINT, ARC, LINE) and activation tool(TOOL).

Complex elements are defined based on simple elements. Thus, POLILINE element can only contain LINE elements, and the element POLICURVE only contains LINE and ARC elements.

Each of these elements has a well-defined attributes or properties in XMLSchema. All elements have a name attribute whose value must be unique within the file. Other attributes refer to the point and the desired orientation of the effector of the robot as well as the type of Euler angles chosen to represent the orientation (ZXZ, ZYZ or RPY). In the case of GROUP and INCLUDE elements, they include too the homogeneous coordinate matrices for the transformation.

The elements of the file should describe robotend-effector motions but always bearing in mind that the end point of a movement must be the beginning of the next movement. That is why the attributes of the elements, except the element TOPOINT, refer to an end of the movement direction. This is intended to prevent the robot is in an uncontrolled position and engaged an unintended movement that may be at risk (collisions).

The definition of new elements in the XMLSchema can be done in two ways.

1. By creating a complex element from simple elements.

Example: triangle definition.

<TRIANGLE name="...>

<LINE name =.. xyz2=... angles2=... angles\_type=.../>

<LINE name =.. xyz2=... angles2=... angles\_type=.../>

<LINE name =.. xyz2=... angles2=... angles\_type=.../>

</TRIANGLE>

1. Through direct definition of a single element.

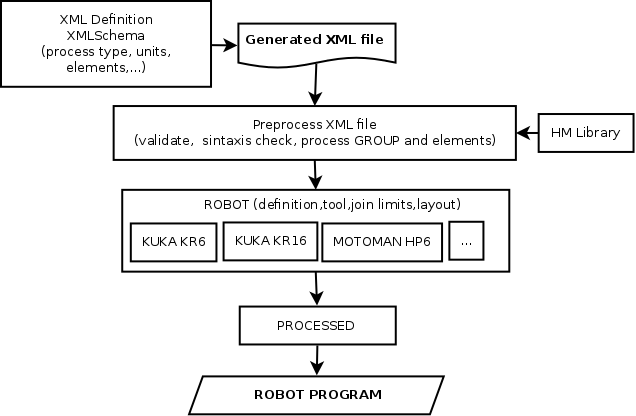
Example: triangle definition.

<TRIANGLE name=... xyz2=... angles2=... xyz3=... angles3=... xyz4=... anles4=... angles\_type=..../>

Modifying the definition of the elements is also possible to better suit different tasks. Due to the particularities of each business process elements defined in the XMLSchema may be suitable for some processes and not others. For example; MIG / MAG welding and spot welding. In the spot welding due to modify the element TOOL to incorporate an attribute to specify the time period that must be active. Therefore, it is desirable to define an XMLSchema for each production process because one has its own characteristics and validation rules will be different.

**5. Description of the process**

The process scheme for the robot program from XML file is as follows.



***Figure 3.****Process scheme.*

5.1. Generation of XML file

In this part, the descriptive XML file of the tasks is generated. The XMLSchema that will validate the file based on the specificities of the process described is defined. For example; spot welding, plane laser cutting, etc...

It is found that the file is syntactically valid and well-formed by some XML editing program (in our case xmlcopyeditor).

For testing, we used an XMLSchema for welding processes MIG / MAG.

5.2. Preprocessed

In this part, a preprocessed of the file elements to single values decomposition in position and orientation is performed. A file of values with which to perform the next stage in the inverse kinematic calculation is generated depending on robot, tool, etc...

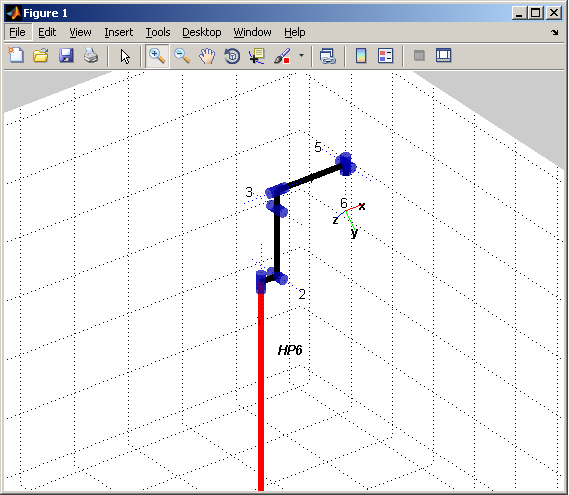
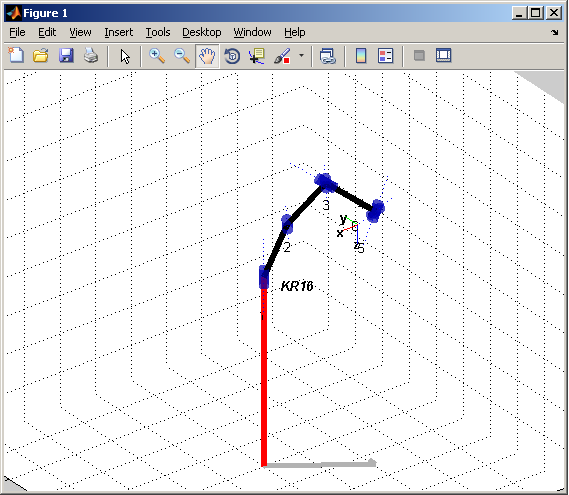
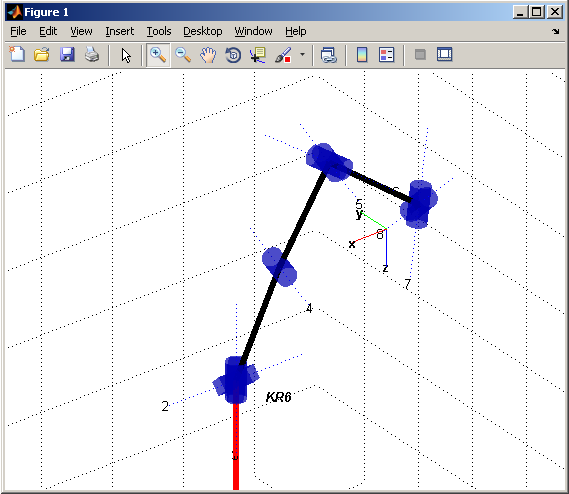
To assist calculations,a library (Library MH) that allows us to do transactions between homogeneous matrices has been developed. This library (or module) has been programmed in Python. Some of the operations that can be performed with this module are: creating homogeneous matrices, multiplication, verification and transformation types.

5.3 Calculation of joint values of the robot

Depending on the robot and its environment (external axes, tool selection, etc...) the necessary robot joint values are calculated to accomplish the tasks.

In our case, we have made a program in MATLAB with the Robotics Toolbox [6] to calculate it. This program allows robot positions to be visualized and possible singularities that may occur in the calculation of the inverse kinematics to be managed.

The input parameters to be defined in this program are: tool, axis robot BASE, starting position (HOME) and ROBOT model. The definition of the robot is done through the Denavitt-Hartenberg parameters. Robots used for testing were: KUKA KR3, KUKA KR6, KUKA KR16 and Motoman HP6.



***Figure 4.*** *Calculating joints values for different robot models in MATLAB.*

5.4 Processed

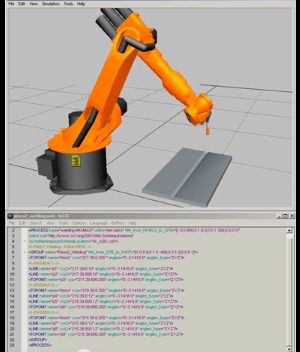
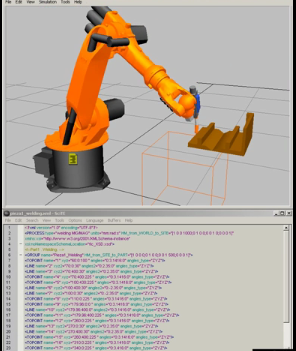
At this stage, the generation of the program for the robot selected is performed.

In our case, the program has been developed for a KUKA robot. Processing was performed for the simulation environment of the manufacturer, but it could be converted to other simulation environments or even the language of the robot directly.

Here, aspects of velocities, accelerations and interpolations needed are defined to then generate robot programming for these values depend on each model.

5.5 ROBOT program

This step verifies that all the movements of the robot are correct and small corrections and modifications are made if necessary.

After verifying the programming, we can proceed to its implementation in the robotic cell (see Figure 5).

***Figure 5.*** *Validation of the program.*

**6. Conclusions**

The description of operations through XML files allows us a greater degree of portability because operations are not subject to a kind of robot or model. By varying the processing part any 6-axis robot can perform the tasks described. Moreover, specific XMLSchema files for the type of process to be performed (milling, welding, drilling, etc...) with the modification or addition of new elements in the definition can also be defined.

The generated XML files are quite readable and let be treated easily because they are plain text files.

Modifications of operations (For example; include a part to an existing piece) are also easy to perform.

LINKS.

XML file example for MIG / MAG.

<https://www.youtube.com/watch?v=uGQ6-c1KHps>

XML file example for milling.

<https://www.youtube.com/watch?v=LEE9Y2mb-hA>

XML file example MIG/MAG welding for a union in V with several passes.

<https://www.youtube.com/watch?v=pR3chIsbw5w>

**7. References**

[1] J. Pérez-Turiel, J.C. Fraile-Marinero, J.R. Perán-González. Aplicaciones de la robótica: últimas tendencias y nuevas perspectivas. Dyna Ingeniería e Industria. Vol.77-3 p.61-68 (2002).

[2] A. Barrientos. Fundamentos de robótica (2º Edición). S.A McGraw\_Hill / Interamericana.

[3]<http://www.w3.org/standards/xml> and <http://www.w3schools.com/>

[4] R. B. Jerard. NCML: An XML Based NC Programming Language, International Conference on Smart Machining Systems, NIST, Gaithersburg, MD, March 13-15 (2007).

[5] A. Chen, F. Dong. Robot machining: recent development and future research issues. The International Journal of Advanced Manufacturing Technology. Vol.66-9-12 p.1489-1497. (2013).

[6] Peter Cooke. Robotics Toolbox for MATLAB

[http://www.petercorke.com/Robotics\_Toolbox.html](http://www.petercorke.com/Robotics_Toolbox.html%20)