ABCM Symposium Series in Mechatronics - Vol. 4 - pp.364-373

PETRI NETS AND GRAPHIC SIMULATION FOR THE VALIDATION OF COLLABORATIVE ROBOTIC CELLS IN AIRCRAFT INDUSTRY

Adriano José Cunha de Aguiar, ajca@ita.br
Emília Villani, evillani@ita.br
Instituto Tecnológico de Aeronáutica – ITA
Divisão de Engenharia Mecânica-Aeronáutica
Pça. Mal. Eduardo Gomes, 50 - Vila das Acácias
CEP: 12228-900 – São José dos Campos/SP, Brazil

Fabrício Junqueira, fabri@usp.br
Escola Politécnica, University of São Paulo, São Paulo, Brazil.

Abstract. *Today, the design of new robotic cells requires the use of graphic simulation tools that make possible the verification and definition of a number of key design points before having the physical devices (robots) and the workcell. Examples are to test the robot reachability, visualize the trajectory, detect collision points, support the definition of layout, determine production times, etc. In this context this paper addresses the issue of designing supervisory control systems for flexible collaborative robotic cells in aircraft industry. For this purpose a mixed approach based on Petri net and graphic simulation is proposed. Petri net is used to model the supervisory system that coordinates the activities of robots. Graphic simulation is used to illustrate the behavior of the robots in a 3D environment. The flexibility of the application requires that the robots work as 'slaves' of the supervisory system. Its trajectory is not previously programmed in the robot controller but is defined, in real time, by the supervisory system. As a consequence, in order to validate the collaborative robotic cell, both simulation tools must be integrated. The motivation of this work is the design of a workcell for the aircraft industry. The workcell automates the processes of drilling and clipping fuselage parts. The workcell is composed of two robots with must work in cooperation.*

Keywords*: keyword Petri nets; Graphic simulation; Collaborative robotic; Validation; Aircraft industry; Robots; Supervisory system*

1. INTRODUCTION

The use of multiple robots in industrial environments has become increasingly frequent. In these systems, the movement and synchronism of these robots became critical factors. In this context, so that a robot performs an action, there must be communication between it and the other robots, and the supervisory system in order to perform a task, preserving its integrity and, also, the integrity of the entire system.

According to Chaimowicz (2002), there are certain types of tasks, called "strongly coupled tasks," which can only be implemented using multiple robots cooperatively. To run this type of task, the robots must act in a coordinated manner, and that in many cases requires some knowledge in relation to the states and actions of other robots and the environment. Furthermore, the occurrence of failures in at least one of the robots can harm/hinder the fulfillment of the task of all.

In this context, this paper addresses the problem of design of flexible robotic cells composed for multiple robots, where a supervisory system is responsible for integration and coordination of robots.

The flexibility in the design of robotic cells is characterized by the following conditions:

- The definition of the trajectory is performed in real time by a system external to the controller of the robot. This is the case for applications where the precision required for the positioning of the tool is higher than the precision provided by the robot. In this case, the tool requires a system for correction of position and orientation that interacts and modifies the position of the robot;
- The cooperation between multiple robots results in an undefined trajectory, needing a system supervisor to coordinate the activities;
- The application requires the processing of exceptions and situations of failure of manner more elaborate than simply suspending the operation of the robot.

Thus, this paper proposes a new approach to the project supervisory systems for flexible robotic cells using the Digital Manufacturing, as the graphical simulation of robots. This approach combines the use of Petri net for modeling and simulation of the logic control of the supervisory system with the graphical simulation of robot, which displays the behavior of robots in a three-dimensional environment.

This work is motivated by a need for the Brazilian aircraft industry: the automation of the structural assembly of aircraft using flexible robotic cells.

This article is organized as follows. Sections 2 and 3 introduce the two techniques used in the proposed approach: graphical simulation of robots and Colored Petri Nets. Section 4 presents the proposed approach and illustrates its

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.364-373

application through a case study of the aircraft industry. Finally Section 5 presents some conclusions and discusses future work.

2. DIGITAL MANUFACTURING AND GRAPHICAL SIMULATION OF ROBOTS

An enormous international competition is a characteristic of nowadays market just as products are increasingly more complex and dynamic is more and more innovative, therefore, along with increasingly reduced cycles of innovation, the product lifecycles and the time to launch a new product on the market are diminishing (Souza *et al.*, 2002).

Furthermore, with the great technological advances, companies are seeking new ways to achieve competitive advantage and bring new products to market faster and at a lower cost, using, for example, the Digital Manufacturing environment. The idea is that this new environment proposed addresses the whole process of development, simulation and manufacturing of the product, enabling the implementation of these activities on the computer virtually before it perform them in the real world, regardless of the degree of complexity of the shape and the structure of the product (Souza *et al.*, 2002).

Also according to Souza *et al.* (2002), the use of Digital Manufacturing is being addressed in some companies, especially the aerospace and automotive, and its concept has developed in scope, gaining greater importance and internationally acceptance. Some companies that already use the Digital Manufacturing as tool are Boeing, which developed the model 777 in a fully digital before doing so physically, DaimlerChrysler, which produced three vehicles using Digital Manufacturing and John Deere, which also has used this new environment in the development of its products (Wave, 2002).

In conformity to Banerjee and Zetu (2001), the Digital Manufacturing can be defined as the modeling of systems and components with the effective use of computers, audiovisual devices and sensors to simulate or design alternatives for an environment to manufacturing, primarily to predict potential problems and inefficiency in the functionality and in the manufacture of the product before the actual manufacturing takes occur.

In the national scene, many companies such as Volkswagen Brasil and ZF do Brasil already use the Digital Manufacturing with the aim to optimize processes, reduce time to deployment and investment, improve quality, standardize and reuse concepts and manage knowledge. The use of graphical simulators for off-line programming and for the accessibility analysis of the tool to pre-determined points and its interference, stands out among the various resources available in an environment of Digital Manufacturing (Zimmermann, 2007).

The graphical simulation of robots, especially for industrial manipulators, is increasingly present in the environment of manufacturing. The ease in which it is incorporated into the off-line programming makes the simulation a powerful tool in the planning of a new work cell or even aid in the programming of robots already existent (Silva, 2004).

The graphical simulation of robot aims to visualize and check the performance of robots in a manufacturing cell, determining some characteristics, such as reachability and envelope. In addition, in robotic cell, the application of graphical simulation is also used as an aid to off-line programming of robots (Silva, 2004) (Stobart and Dailly, 1985) (Aguiar *et al.*, 2007a).

Among the benefits offered by graphical simulation of robots stand out (Silva, 2004) (Aguiar *et al*., 2007b):

- Decrease the time of manufacturing products: with the whole trajectory set, it is possible to calculate the time spent in each operation by means of simulation and, thus, changing parameters and resetting trajectories, searching for more efficient production.
- Verification of accessibility: some operations performed by the robot, as the access or the reach, can not be allowed for a particular type of robot. With the graphical simulation, it is possible, by means of libraries provided by the manufacturer of the robot, to simulate the access without the need of a physical contact, reducing the cost and time;
- Adaptability of programs: the possibility of adaptation of similar operations is plausible. With the change of operations in an automotive assembly line due to a new project car, it is easy to adapt the existing trajectories.

A simulation robots system must cover from models of robots and structures to algorithms of inverse kinematics. The inverse kinematics allows the programming of the position or trajectory of the tool in Cartesian space without the need of the user manipulate the parameters of the joints, leaving the simulator responsible for calculating process of the joints angles (Aguiar *et al*., 2007b).

Thus, despite the ease that the simulator provides the user, planning trajectories for a robotic manipulator is still a complex task that involves various aspects such as model barriers, manipulate information from sensors, consider the dynamics of the robot, search for collision-free trajectories and avoid settings that cause singularities in the robot (Leng and Chen, 1997).

Currently, a large number of commercial applications oriented by graphical simulation and off-line programming of robots are available in the market. These products are produced both by companies developing products for CAD and companies that design and commercialize robots, launching their own applications on the market (Silva, 1996).

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.364-373

ROBCAD, RobotStudio, Easy-Rob, KUKA.Sim Pro and Grasp10 are some examples of commercial applications used and available on the market.

3. COLORED PETRI NETS

The second technique used to verify the proposed approach is the Petri Net. Petri Nets (PNs) are a graphical and mathematical modeling technique originally developed by C.A. Petri in the early 1960s to characterize concurrent operations in computer systems. The greatest appeal of PNs is their conceptual simplicity (Moore and Brennan, 1996). It is also good to describe static and dynamic system characteristic, and system uncertainty. The structure of PN models can be exploited to develop efficient algorithms for system control (Qiao et al., 2002).

PNs have been extended to capture many important aspects of systems, which includes attributes, timing relationships, and stochastic events (Moore and Brennan, 1996). Colored Petri Nets (CPNs) extend the classical Petri Nets with colors (to model data), time (to model durations), and hierarchy (to structure large models) (Mulyar and Van der Aalst, 2005). In real-world systems, we often find many parts that are similar. These parts must be represented by disjoint and identical sub-nets in PNs. This means that the net becomes largely and it becomes difficult to see the similarities between the individual sub-nets. CPNs provide a more compact representation where individual sub-nets are replaced by one sub-net with different kind of tokens, each token having a color and representing a different sub-net in the equivalent PN (Elkoutbi and Keller, 1998) (Qiao et al., 2002). CPNs are very appealing and appear uncomplicated because of their simple graphical representation (Arjona and Bueno, 2003). Like in classical Petri Nets, CPNs use three basic concepts: transition, place, and token (Mulyar and Van der Aalst, 2005) (Jensen, 1997a). CPN is a graphically oriented modeling language capable of expressing concurrency, non-determinism, and system concepts at different levels of abstraction. CPNs combine PN and programming languages within the same mathematical framework. Petri Nets are used to model concurrency, synchronization, and resource sharing and allocation, whereas a functional programming language is used to model data manipulation and to create compact and parameterized models (Zhang et al., 2001).

Places are used to store entities. Color sets associated with the places indicate the different types of entities that can be stored in the places. Entities are represented by color instances (copies) of the colors associated with the places and are referred to as colors or tokens. Tokens stored in the places, referred to as the marking of the CPN, define the state of the system. Transitions represent sets of related events. Transitions fire (are executed) to change the state of the system. Color sets associated with the transitions indicate the different ways (events) in which the transitions can fire (Arjona & Bueno, 2003).

Given a CPN net structure and an initial marking, the dynamics of a CPN is determined by the enabling and occurrence rules, modeled by arc inscriptions and guards of transitions (Gordon and Billington, 1998) (Zhang et al., 2001). There are two types of arcs in respect to transitions: incoming arcs and outgoing arcs. Incoming arcs connect from input places to a transition, and outgoing arcs connect from a transition to output places. Inscriptions on incoming arcs specify the numbers and colors of tokens from the input places that must be in place in order for a transition to be enabled; and, once the transition occurs, to be consumed. Inscriptions on outgoing arcs specify tokens that the occurrence of a transition produces and puts into the output places (Zhang et al., 2001).

Transition guards are optional, and if present, impose additional conditions for transitions to be enabled and to occur. Enabled transitions can occur either concurrently or sequentially, and in various orders, depending on the numbers of available tokens in places. It is also possible that the occurrence of some transitions changes the state of a net such that the conditions of other enabled transitions are no longer met (Zhang et al., 2001).

Despite its conceptual simplicity, CPNs have proved to be a very powerful modeling tool for discrete event systems, capable of modeling sequences, conflicts, concurrences, and synchronization. They provide a formal framework for the design, specification, validation, and verification of discrete systems (Arjona and Bueno, 2003).

PN has been used in a large variety of areas. Their application ranges from informal to formal systems and from software to hardware systems and from sequential to concurrent systems. As mentioned in (Jensen, 1997b) PN are used in communication protocols, distributed algorithms, computer architecture, computer organization, human-machine interaction and many others areas (Elkoutbi and Keller, 1998).

One issue with using PN to model a manufacturing system is that the difficulty of building and analyzing a PN increases greatly with the complexity of the system being modeled. If a system model is very complex, containing thousands of nodes and transitions, the analysis of this model will be very difficult and time consuming. An approach must be developed where correct PN models of a complex system can be developed and extended from simpler models that are easy to prove valid (Qiao et al., 2002).

The wish to model and analyze large systems by means of PN has shown the need for a modular approach. The main advantages which are expected from this approach are (Sibertin-Blanc, 1993): (a) a better command of the complexity of systems, thanks to a rigorous structure of the model and the possibility to consider different parts of the model independently of each other; and (b) a greater ease to adapt, correct, analyze or reuse a model, thanks to the localization of these tasks at the level of components.

PNs may be connected by merging of transitions, by merging of places, and also by arcs (Sibertin-Blanc, 1993) (Lakos, 2005) (Jiao et al., 2005) (Westergaard, 2004). Transitions merging have the advantage to ease the system's analysis, because many properties are preserved when nets are composed in this way (Sibertin-Blanc, 1993).

Composition of nets by places merging corresponds to communication by variable sharing. The management of the shared place requires an additional synchronization between the nets. Connecting nets by arcs ensures the smallest coupling and it corresponds to communication by "message sending" (arc from a transition to a place) and "message taking" (arc from a place to a transition) (Sibertin-Blanc, 1993).

This work uses the CPN and CPNTools application for modeling the control logic of supervisory system and robots.

4. VALIDATION OF FLEXIBLE ROBOTIC CELLS

4.1. Proposed Approach

The approach proposed in this study for validation of flexible robotic cells is organized in the following steps:

Step 1: Process Modeling in PFS

Using the PFS (Production Flow Schema) modeling technique, it should develop models of robots programs and monitoring system so they have the entities and their relationships in a high level of abstraction.

Step 2: Definition of Communication Protocols

Using information from the previous step, it should set a standard for communication between components of the cell, i.e. the commands and variables which will be exchanged and what should be the logical.

Step 3: Modeling of the Programs in Petri Net

The top-down approach is adopted in this step. Using Petri net, it should model from the models developed in PFS, the programs of robots and supervisory system in a higher level of detail. Subsequently, you should include the models by means of exchanging messages with the logic used in Step 2.

Step 4: Validation of Models

The models in Petri net, now integrated, should be validated by simulation and/or by a formal verification.

Step 5: Conversion of the Models

The model of robot programs to be converted to Petri net for the programming language of robots. The model of the supervisory system should already be converted to a structured programming language.

Step 6: Simulation

The simulator of robots and the supervisory system program must be implemented in an integrated way in order validate the procedure.

4.2. Case Study

The drilling and placement of rivets in aircraft fuselages are currently implemented in a manner predominantly manual in the Brazilian aircraft industry. Aimed at automating the process, the proposed approach is applied to the design of a robotic cell for assembly of aircraft structural.

The application considered deals with the movement of two robots in tightly coupled tasks, running the drilling and placement of rivets in a longitudinal junction of fuselage, as indicated by the arrows in Figure 1.

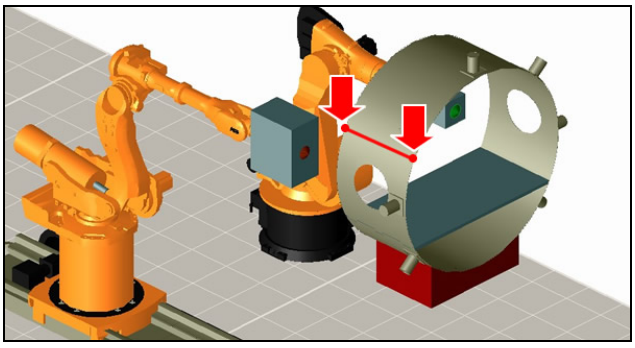

Figure 1. Robotic cell

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.364-373

To simulate the performance of two industrial robots interacting with the section of fuselage and other components, ROBCAD[1] was used as tool for the graphical simulation.

Following this approach, the supervisory system and the programs of the robots were modeled in a Coloured Petri Net and later converted to a programming language of ROBCAD, the TDL. Information on the TDL can be found at (Siemens PLM Software, 2009).

The C programming language was adopted to program the supervisory system.

The implementation of the proposed procedure in this case study is shown below.

Step 1: Process Modeling in PFS

The process considered in this application is the implementation of a longitudinal junction between two sections of fuselage. The robot 1 uses the tool for drilling and insertion of rivets, while the robot 2 operates as counterpoint of the operation. The PFS of the process is illustrated in Figure 2.

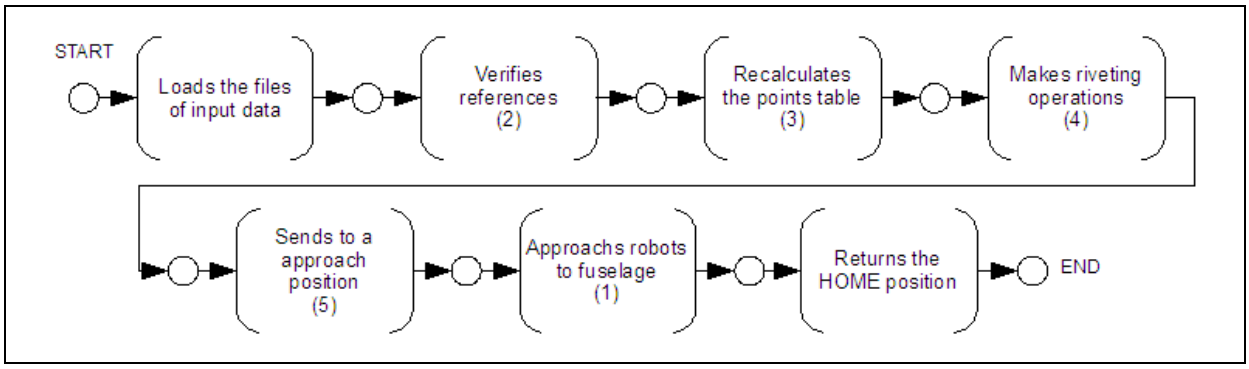

Figure 2. General PFS of the process

The process begins with loading the file of input data. This file consists of the positions of the references installed in the fuselage and by the ratio of operations for insertion of fasteners to be executed.

Following, the robots are closed to fuselage (activity 1). This movement is carried out in fast speed.

The actual position of each reference should be verified by Robot 1 (activity 2), which has a system of vision. In the process used as an example there are only two references.

Depending on the difference between the actual position and the position defined in the data file of the two references, the position of the rivets is corrected (activity 3). Both robots are then used to make the installation of the rivets (activity 4). Activities 3 and 4 are performed on low speed. Following these activities the robots are sent to the HOME position with command of fast speed.

Step 2: Definition of Communication Protocols

The commands to be exchanged between robots and supervisory system for accomplishing this process are:

- Command "move $joint" to move the robot in fast speed;
- Command "move $linear" to make incremental movements in low speed.

The data to be exchanged between each robot and the supervisory system are organized in the following variables:

- Position and orientation of destination: set of 6 real variables;
- Flag_interno: integer variable that indicates the robot that it should execute a new command in accordance with data of other variables. This variable also identifies the command (joint or linear) to be executed by the robot;
- Flag_externo: binary variable that informs to the system of supervision that the robot completed the last command requested.

Step 3: Modeling of the Programs in Petri Net

The PFS of Figure 3 is detailed in Colored Petri nets. As example, shows the net equivalent to activity 1. The other activities are detailed in a similar way.

[1] The environment ROBCAD software, version 8, supplied by Siemens PLM Software to the Centro de Competência em Manufatura (CCM/ITA).

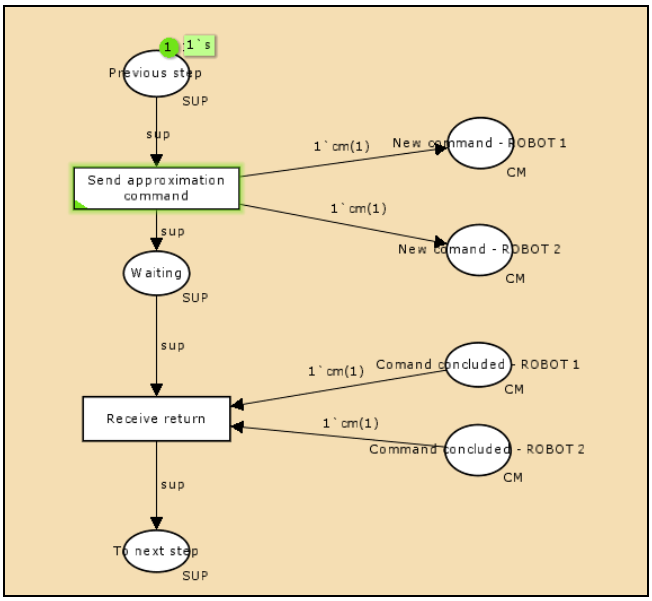

Figure 3. Details of the activity 1

The nets corresponding to implementation of programs of the robots are built too. In this case, both networks of Robot 1 and Robot 2 are similar (Figure 3).

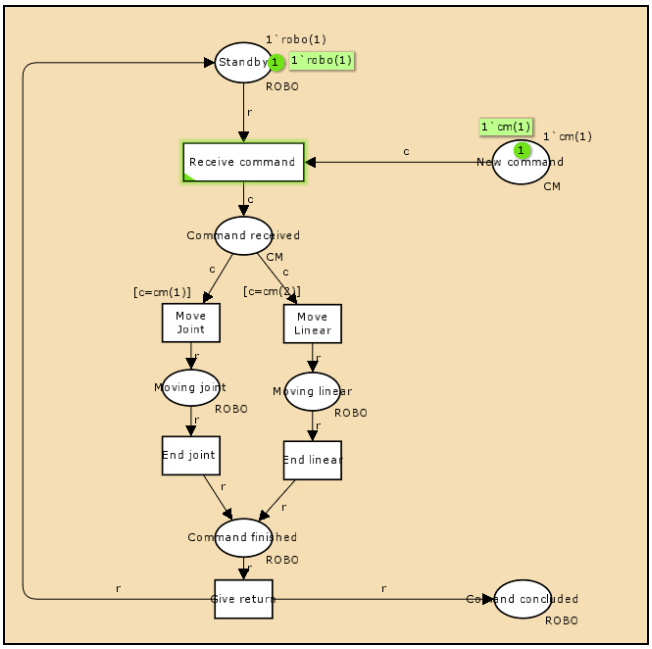

Figure 4. Net of Robot 1

Step 4: Validation of Model

The models in the Petri nets are validated by simulation. In this case the CPN Tools application was used (CPN Group, 2009).

Step 5: Conversion of the models

The Petri net resultant of the supervisory system was converted to C language. The Petri net corresponding to the robot programming was converted TDL language of the ROBCAD. The communication between the ROBCAD and the supervisory system program was performed by means of writing and reading of files containing shared variables exchanged between the two applications. Due to the limitation of ROBCAD communication, this was the chosen way to exchange messages with external programs.

The input data are composed by:

- Six real numbers that indicate the position of the first reference with regard to the global coordinate system. The first three numbers indicate the coordinates X, Y and Z and the last three indicate the angles of rotation rX, rY and rZ;
- Six real numbers that indicate the position of the last reference with regard to the global coordinate system. The first three numbers indicate the coordinates X, Y and Z and the last three indicate the angles of rotation rX, rY and rZ;
- One integer number that indicates how many rivets should be inserted between the two references;
- A table with the original position of installation of the rivets.

Step 6: Simulation

In order to verify the process of modeling and creation of the programs, a simulation was implemented by using the models developed in the ROBCAD integrated to supervisory system written in C language.

Some files were generated in this simulation. They made the communication between the two programs by writing and reading, which are listed below:

- "flag1_interno.robcad" and "flag2_interno.robcad": files responsible for receiving the signal of the type of movement from the supervisory system and to provide to robots for reading. Each robot has a file for reading;
- "flag1_externo.robcad" and "flag2_externo.robcad": files responsible for receiving the signal indicating that robots made certain task and to available for reading the supervisory system. These files are only needed when there is the task of synchronization. Each robot has a file for writing;
- "dadosrobo1.robcad" and "dadosrobo2.robcad": files responsible for storing information of coordinate necessary to the movement of robots. Each robot has a file for reading;

To initialize the simulation, it needed to create a file called "coordenadas.robcad" containing the input data for the simulation. Its content is provided in Table 1.

Table 1. Content provided to the supervisory system

1	5
2	5750 5898.8 2135.05 -105 0 0
3	6750 5898.8 2135.05 -105 0 0

The first line of Table 1 shows how many references should be calculated between the first and last. The second and third rows show the first and last reference relating to the global coordinate system, being six real numbers. On line 1, the X, Y and Z columns indicate respectively the coordinates X, Y and Z and rX, rY and rZ columns indicate the angles of rotation relating to the global coordinate system. In other rows, the X, Y and Z columns indicate respectively the displacements in X, Y and Z and rX, rY and rZ columns indicate the rotations relating to the tool coordinate system.

Table 2. Data calculated by the supervisory system

Line	Data of the robot 1						Data of the robot 2					
	X	Y	Z	rX	rY	rZ	X	Y	Z	rX	rY	rZ
1	5750.0000	5802.2072	2160.9320	-105.0000	0.0000	0.0000	5750.0000	5995.3924	2109.1681	75.0000	0.0000	0.0000
2	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000
4	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000
7	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000
10	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000
12	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000
13	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000
16	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000
18	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000
19	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000	166.6667	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.0000	0.0000	0.0000
21	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-100.0000	0.0000	0.0000	0.0000
22	-833.3334	0.0000	0.0000	0.0000	0.0000	0.0000	-833.3334	0.0000	0.0000	0.0000	0.0000	0.0000

The coordinates calculated for the movement of robots by the supervisory system are presented in Table 2, being the coordinates of the robot 1 of the second until the seventh column and the coordinates of the robot 2 of the eighth until the thirteenth column. The first row is the coordinates of the first reference relating to the global coordinate system. This reference has a distance of approach / departure surface perpendicular to the fuselage, as shown in point 2 of

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.364-373

Figure 5. Thus, the first line provides data to the robots move from the HOME position to the approaching position on high speed and on joint movement, because not require accuracy. The second line of Table 2 shows the coordinates relating to tool coordinate system for the movement of robots on low speed and on linear movement with the goal of bringing the tools to the fuselage of manner secure and can be seen in Figure 5 the origin (point 2) and destination (point 3) of the movement of the robot 1. Lines 3 and 4 of Table 2 also show the coordinates relating to the tools coordinate system for departure and displacement to a next approximation. The steps of approximation, departure and displacement are repeated in line 5 to line 22, except the last line because it provides data to movement the robot from last departure point to the first point of approximation, as exemplified the movement of the robot 1 from point 5 to point 2 in Figure 5, thereby ensuring the safety of the cell, because the robots will perform movements known and insurance.

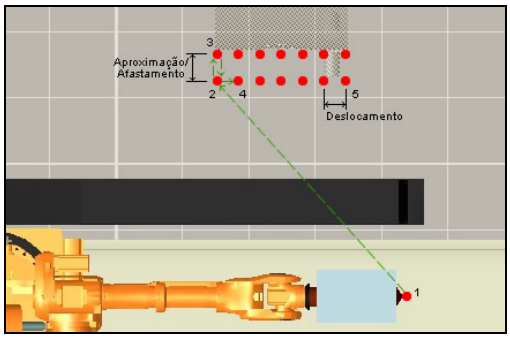

Figure 5. Detail of movement of the robot 1

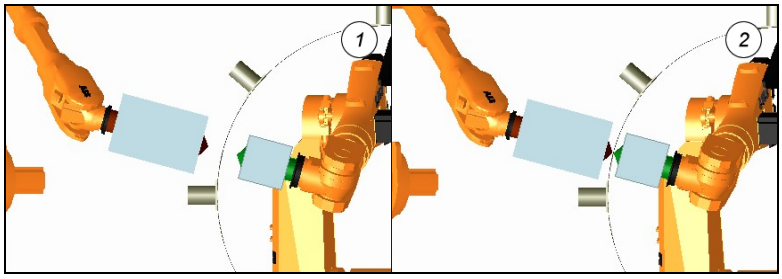

Figure 6. Perpendicularity of the tools relating to the surface of the fuselage

The last movement from the departure point of the first reference to the HOME position does not require data to the robots.

It is observed in Figure 6 the previous step (1) and posterior (2) to the stage of approximation. There are two moments in which the tools are perpendicular to the surface of the fuselage and aligned, moving up slowly and in a linear manner until they are in the fuselage. This provides greater accuracy in the motion.

Results obtained by the graphical simulation with the program of supervision proved to be satisfactory because the access to the fuselage were made without collision and perpendicular, guaranteeing successful of the work. It was observed that the robot 2 came close to the singularity in the last stages of simulation due to a lower range.

5. CONCLUSION AND FUTURE WORKS

Some facilities and benefits that the graphical simulation incorporates the design of a robotic cell are observed in this work. Several techniques of access and communication between robots and external systems can be simulated satisfactorily without the need of using real components, reducing costs and increasing safety. Furthermore, research in the area of modeling and simulation is necessary, especially in developing new techniques for modeling.

The modeling procedure proposed in this work using Petri net makes it possible to better comprehension and detailing of the tasks of each component of the system, characterizing the elements and their relationships. Moreover, the centralization of decision-making powers in an external element makes the system more flexible and intelligent, leaving the robots just waiting for procedures and running them, creating a range of possibilities for activities, unlike the traditional off-line programming, which restricts to activities fixed and not very flexible.

As suggestions for future work:

- Use exchange of messages with the robot's tools simulating corrections in the trajectories at each stage;
- Include other steps to the procedure, as the conversion of the TDL program to a programming language native to robots, substitution the exchange of messages via simulator by the standard communication used by robots and the execution of tests in real environment;
- Develop procedures for drilling and placement of rivets of manner orbital, as the arrows shown in Figure 7. Therefore, the use of track motion as the seventh axis becomes more effective, because it would facilitate the access more complex by redundancy (Aguiar *et al*., 2007a) (Aguiar *et al*., 2007b).

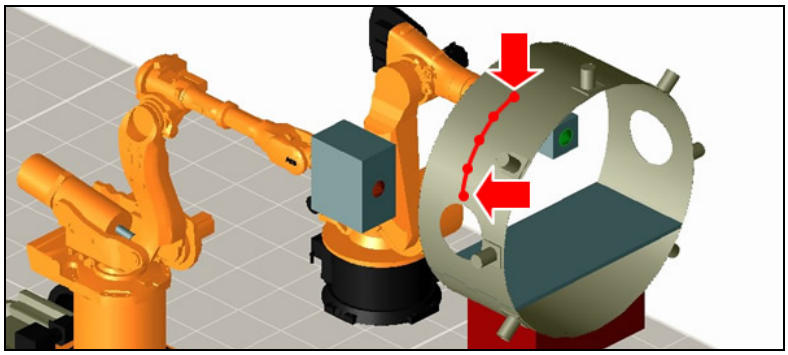

Figure 7. Drilling and placement of rivets of manner orbital

6. REFERENCES

Aguiar, A.J.C., Silva, A.S.A., Villani, E., 2007a. “Graphic Robot Simulation for the Design of Work Cells in the Aeronautic Industry”, 19th COBEM – International Congress of Mechanical Engineering, Brasília/DF, Brazil.

Aguiar, A.J.C., Silva, A.S.A., Villani, E., 2007b. “Simulação Gráfica de Robôs Aplicada à Indústria Aeronáutica”, VIII SBAI – Simpósio Brasileiro de Automação Inteligente, Florianópolis/SC, Brazil.

Arjona, E., Bueno, G., 2003, “Colored Petri Net Modules of Complex Selecting Criteria”, Simulation, Vol. 79, Issue 7, pp. 410-419, July.

Banerjee, P. and Zetu, D., 2001. “Virtual manufacturing”, New York: John Wiley & Sons. 320p.

Chaimowicz, L., 2002. “Coordenação Dinâmica de Robôs Coopernativos: Uma Abordagem Utilizando Sistemas Híbridos”, Tese de Doutorado, Universidade Federal de Minas Gerais. 126 p., Belo Horizonte, Brazil.

CPN Group, 2009, “Computer Tool for Coloured Petri Nets”. 13 May 2009, <http://wiki.daimi.au.dk/cpntools/cpntools.wiki>.

Elkoutbi, M., Keller, R.K., 1998, “Modeling Interactive Systems with Hierarchical Colored Petri Nets”, Proceedings of the 1998 Advanced Simulation Technologies Conference, p.432-437, Apr.

Gordon, S., Billington, J., 1998, “Applying Coloured Petri Nets and Design/CPN to an Air-to-Air Missile Simulator”, Workshop on Practical Use of Coloured Petri Nets and Design, Aarhus University, Denmark, pp. 1-14, 1998.

Jensen, K., 1997a, “A Brief Introduction to Coloured Petri Nets”, Proceedings of the Third International Workshop on Tools and Algorithms for Construction and Analysis of Systems, pp. 203-208.

Jensen, K., 1997b, “Coloured Petri Nets, Basic concepts, Analysis methods and Pratical Use”. Springer.

Jiao, L., Cheung, T.Y, Lu, W., 2005, “Handling Synchronization Problem in Petri Net-Based System Design by Property-Preserving Transition-Reduction”, The Computer Journal, Vol. 48 No. 6.

Lakos, C.A., 2005, “A Petri Net View of Mobility”, In: IFIP International Federation for Information Processing, pp. 174–188.

Leng, D.Y. and Chen, M., 1997. “Robot trajectory planning using simulation, Robotics & Computer-Integrated Manufacturing”, 13(2): 121-129.

Moore, K.E., Brennan, J.E., 1996, ‘Alpha/SIM Simulation Software Tutorial”, Proceedings of the 1996 Winter Simulation Conference.

Mulyar, N., Van der Aalst, W.M.P., 2005, “Towards a Pattern Language for Colored Petri Nets”, Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005), volume 576 of DAIMI, pages 39-48, Aarhus, Denmark, October.

Qiao, G., Mclean, C., Riddick, F., 2002, “Simulation System Modeling for Mass Customization Manufacturing”, In: Proceedings of the 2002 Winter Simulation Conference, pp.2031-2036.

Sibertin-Blanc, C., 1993, “A Client-Server Protocol for the Composition of Petri Nets”, Proceedings 14th International Conference on Application and Theory of Petri Nets. Chicago, Illinois, USA, p.377-396.

Siemens PLM Software, 2009. “Robcad Rerefence Manual: Robcad TDL Rerefence Manual”.

Silva, M.F.S., 1996. “Simulação e Programação Off-line de Robôs de Montagem. Porto”, 219 f.. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) - Faculdade de Engenharia, Universidade do Porto.

Silva, M.S., 2004, Aplicação da Simulação à Programação Off-Line de Robôs Industriais, Ingenium, No. 81, pp. 72-77.

Souza, M.C.F., Porto, A. J. V., R., C. A. and Batocchio, A., 2002. “Manufatura Virtual: Conceituação e Desafios”. GESTÃO & PRODUÇÃO, v.9, n.3, p.297-312.

Stobart, R.K. and Dailly, C., 1985. “The use of simulation in the off-line programming of robots”, IEE Control Engineering - Robots and automated manufacture, Series 28, Stevenage, United Kingdom, pp. 11-28.

Wave Report, 2002. “3D – Points to Ponder”. 2 jul. 2007, <http://www.wave-report.com>.

Westergaard, M., 2004, “Towards A High-Level Petri Net Type Definition”, Proceedings of Workshop on the Definition, Implementation and Application of a Standard Interchange Format for Petri Nets. Satellite event at the 25th International Conference on Application and Theory of Petri Net, Bologna. Italy, June.

Zhang, L., Mitchell, B., Falzon, L., Davies, M., 2001, “Model-based Operational Planning Using Coloured Petri Nets”, 6th International Command and Control Research and Technology Symposium, pp. 1-15. 19-21, Annapolis, MD, USA, June.

Zimmermann, S.C., 2007. “Fábrica Digital”, II Seminário de Manufatura Automotiva, São José dos Campos, Brazil.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.