

Robot Path Simulation: a Low Cost Solution Based on CAD

Pedro Neto, J. Norberto Pires

Department of Mechanical Engineering (CEMUC)
University of Coimbra
Coimbra, Portugal
(pedro.neto|jnp)|@robotics.dem.uc.pt

A. Paulo Moreira

Institute for Systems and Computer Engineering of Porto
University of Porto
Porto, Portugal
amoreira@fe.up.pt

Abstract—The benefits of robot simulation technology have been recognized by scientists and engineers, with applications ranging from simple robot path simulation to complete robotic cell layout simulation. Robot simulation is one of the essential elements of modern and agile manufacturing plants, as it allows to visualize and test a robotic system, even if it does not exist physically. High-growth industries and emerging manufacturing processes will increasingly depend on advanced robot technology such as robotic simulation. Robot path simulation is a very useful process to predict and pre-evaluate performance of robot programs generated off-line. This paper presents a simulation system where a relatively low cost and commercially available 3D CAD package is used as an interface to visualize/simulate pre-programmed robot paths. The developed system is intuitive and friendly, so that in a few minutes any user without knowledge of CAD and robot programming will be able to simulate robot paths and visualize it in a CAD environment. This way, this simulator can be useful for small and medium sized enterprises and for educational purposes. Three different experiments (simulation of robot motion/paths) are presented and discussed: a material handling task; a robot performing work in the footwear industry (shoes soles); and finally, the simulation of robot paths for a robot operating in a bending cell. The experimental results showed that the proposed system is flexible, easy to use and efficient. This paper also covers topics like how simulation makes robot programming easier, advantages and disadvantages of simulation in robotics, and the future trends in this field. Finally, the results of the experiments will be analyzed and discussed. The pros and cons of the system in relation to off-the-shelf robot simulation packages are analyzed.

Keywords—robot, simulation, CAD, low cost

I. INTRODUCTION

The ever increasing demand for short production cycles and high product quality requires increased manufacturing process flexibility. This flexibility can be achieved for example through the introduction of robots in the manufacturing shop floor. Generally, small and medium sized enterprises (SMEs) have more difficulty in automating its production systems because they have no skilled labor available to upgrade manufacturing facilities [1].

The greatest advantage of robots is their flexibility, i.e. their ability to be rearranged for new production tasks. Utilization of the robot's flexibility presupposes effective, intuitive and fast programming. Today, most industrial robots

are still programmed on-line using the robot teach pendant, however, this method is not a suitable approach for modern manufacturing systems. On the other hand, an off-line programming approach could be used to drastically reduce programming and set up time. The off-line programming concept allows that the robotized cell can keep production while the programmer realizes new robot programs, in an office environment [2]. Nevertheless, any robot program which is off-line written should be tested before launching the real production system run [3].

Industrial robot programming based on off-line techniques and simulation systems enables the creation and optimization of robot programs without interfering with production, and thereby reducing downtimes. Moreover, off-line programming also offers the possibility of using simulation technology to make experiments with a variety of scenarios and processes [4].

Simulation has been recognized as an important research tool. The growing interest in simulation is due, in part, to the development of computers. Today simulation is a powerful visualization, planning, and strategic tool in different areas of research and development, and robotics as a modern technological field is no exception [5]. Robotic simulation has played an important role in the development of the “digital factory” and in the last few years has witnessed a striking expansion, for example for plant layout simulation and optimization [6]. Simulation opens a wide range of options for solving problems creatively, allowing to investigate, visualize and test a robotic cell, even if it does not exist physically. The boundaries for what is and what is not feasible are pushed far away. Using simulation tools one can avoid injuries, damages, unnecessary changes in design after the production of parts has already started, and long cycle times. All these things make the manufacturing process easier and when used properly make the process cheaper. Simulation helps the designer of a robotic system to determine an appropriate choice of robot and its workplace elements, including feeders, tools, grippers, fixtures and sensors. As the complexity of the system under investigation increases the role of simulation becomes more and more important, so that this is an important tool for car manufacturers simulate their plant layouts [7]. An ideal simulation process approaches reality so closely that the robot programs developed off-line can be loaded into the real robot controllers without having to correct them at the shop floor.

Unfortunately, programming and simulation of a manufacturing system which includes industrial robots can involve far greater complexity. This is a direct consequence of the wider variety of possible application areas and potential solutions given by the enhanced flexibility offered by robotics.

This paper presents a simulation system where a relatively low cost 3D CAD package, Autodesk Inventor, is used as an interface to visualize/simulate pre-programmed robot paths, allowing a quick assessment of the performance of robot programs generated off-line. Thus, the user can graphically visualize the robot paths on a computer screen before converting it into effective robot programs. The developed system is intuitive and friendly, so that in few minutes any user without knowledge of CAD and robot programming will be able to simulate robot paths in a CAD environment. This way, the simulator can be useful for SMEs. Three different experiments will be presented and discussed: first, simulation of robot motion/paths for a material handling task; second, simulation of robot motion/paths for a robot performing work in the footwear industry (shoes soles); and finally, the simulation of robot paths for a robot operating in a bending cell. The experimental results showed that the proposed system is flexible, easy to use and efficient. This paper also covers topics like how simulation makes robot programming easier, advantages and disadvantages of simulation in robotics, and the future trends in this field. Finally, the results of the experiments will be analyzed and discussed. The pros and cons of the system in relation to off-the-shelf robot simulation packages are analyzed.

II. ROBOTIC SIMULATION: AN OVERVIEW

A variety of simulation tools has been developed in the robotics field, and today the motion behavior of simulated robots can match reality with high precision. Robot simulation packages are already being used extensively, however the high price of these systems is still a problem that hinders their use.

Robotics is a very interdisciplinary and fast growing research field, as robots are introduced in various areas and used for many different operations [8]. The requirements for the simulation tools depend on the particular application. However, all new applications, methodologies and technologies in robotics share the requirement to simulate robot systems with sufficient sophistication and accuracy.

A number of robot simulators (Delmia Igrip, Robcad, RoboSim, etc.) have become available in the world marketplace. These software packages provide a set of modeling and simulation tools which can be used to represent graphically a robot manipulator and its attendant equipment, and hence simulate any manufacturing task [9]. Some robot simulation packages have capabilities to program most of robot types, from different manufacturers. On the other hand some robot manufacturers provide their own simulation package dedicated to their own robots (MotoSim from Motoman, RobotStudio from ABB, etc.). Early versions of robot simulation packages were based on simple wireframe models of the robot's kinematics. However, in recent years robot simulation techniques have seen a rise in realism and popularity, possibly coinciding with the advancement of computing and graphical animation technologies. The use of

such packages can allow the manufacturing engineer to try several solutions for robotic cells before purchasing any equipment or to investigate alternative uses of existing cells. Hence, these simulation packages can be used to improve the choice and layout of robotic systems, reduce set up costs, reduce installation times and improve system performance.

Reference [10] reports the necessity to generate robot paths from CAD data of different free forming surfaces. A simulation interface was used to parameterize the robot path and visualize the generated paths. Commercially available robot simulation systems led researchers to propose the use of a postprocessor in an off-line robot programming system [11]. Already in the 80's, it was used an off-line robot programming system to design and program the robots that weld the space shuttle main engines. Engineers programmed the operation even before the equipment arrived, based on the welding parameters database specified by welding engineers and the CAD database that describe the parts to be welded [12]. Another off-line programming approach for welding robots is presented in [13]. The approach is based on the integration of a software tool for robot simulation and automatic generation of robot programs. Reference [14] describes a method to establish a virtual environment using Java3D.

In fact, it is very difficult to list all of the possible advantages that can be obtained with a good robotic simulation tool. The most prominent issues are listed below:

- a) *Evaluation and validation of system response to abnormal events.*
- b) *Detection of collisions.*
- c) *Prediction of quality, productivity and efficiency performance.*
- d) *Robot simulation allows to adjust timings of production.*
- e) *The ability to visualize robot motion can quickly answer questions regarding path planning, workspace constraints and coordination issues with other systems.*
- f) *Reduce downtime incurred by on-line programming. Without simulation it is likely that a proper solution would require many interactions on the factory floor before a satisfactory solution is obtained.*
- g) *Simulation helps us to understand what type of modifications and improvements should be made.*
- h) *Prediction of performance without taking the robot out of production during the programming phase.*
- i) *Allow to design and optimize robotic systems in advance, ahead of production.*
- j) *Allow to test robot programs quickly, intuitively and in a safe environment.*
- k) *Evaluation of performance: path generation and trajectory planning.*

Nevertheless, all of these capabilities come at a cost. A license for a commercial robot simulation software costs thousands of dollars, a situation which is not tolerable for SMEs. It was seen that robot simulation by graphical computer methods has considerable potential advantages, but these are tempered by

some limitations of existing systems. The above programming examples are geared toward simulation of industrial facilities where the environment is known *a priori* and well modeled. However, such an approach will fail if the model is inaccurate or if the environment surrounding the robot is changing in an unpredictable fashion (is “unstructured”). In such cases the limited “intelligence” and adaptability of today’s robots requires that the operator remain in the loop.

III. SYSTEM ARCHITECTURE AND EXPERIMENTS

A. System Overview

A robot path simulation system based on a 3D CAD package (Autodesk Inventor) has been developed. This simulation system allows to display the virtual robotic cell and robot paths on a computer screen, providing a highly detailed 3D view. The aim was to create a user friendly and low cost robot path simulation system to quickly simulate robot motion. The system input can be a robot program or a data file containing information about robot motion. An important area of concern is the ease of use of the simulation systems. Our goal was to create an extremely simple interface, not only functional but simple as well.

After uploading a data file or a robot program to the simulation system, the user should define the necessary reference frames. This is a very simple process where the user only needs to indicate the necessary reference frames in the CAD model. This is achieved by selecting the origin and orientation for the frame into the CAD environment. Finally, the virtual robot paths are generated and displayed into the CAD environment. After analyzing the virtual paths, and if the result is not satisfactory, the user can change the robot program and quickly repeat all the process (Fig. 1).

The way the simulator interprets robot programs and/or the data files is described in the following steps:

- 1) Read the input file (robot program or data file).
- 2) The system detects the robot tag points and collect it.
- 3) Search instructions that make the robot moves. These instructions begin with the characters “MOV”. This phase is only necessary when the system input is a robot program.
- 4) After detecting a robot movement instruction it is necessary to specify what type of instruction it is (MOVS, MOVL, MOVJ or MOVC) and what path point is associated with this movement instruction. The instruction MOVS moves the robot to the specified point with spline motion type, MOVJ with joint motion type, MOVL with linear motion type and MOVC with circular motion type. In our simulation software the MOVJ instruction is treated as the MOVL instruction. This phase is only necessary when the system input is a robot program.
- 5) Finally, in order to “draw” the virtual robot paths, the system starts to create an invisible part and then use its “sketch” functionalities to “draw” the robot paths in a 3D CAD model.

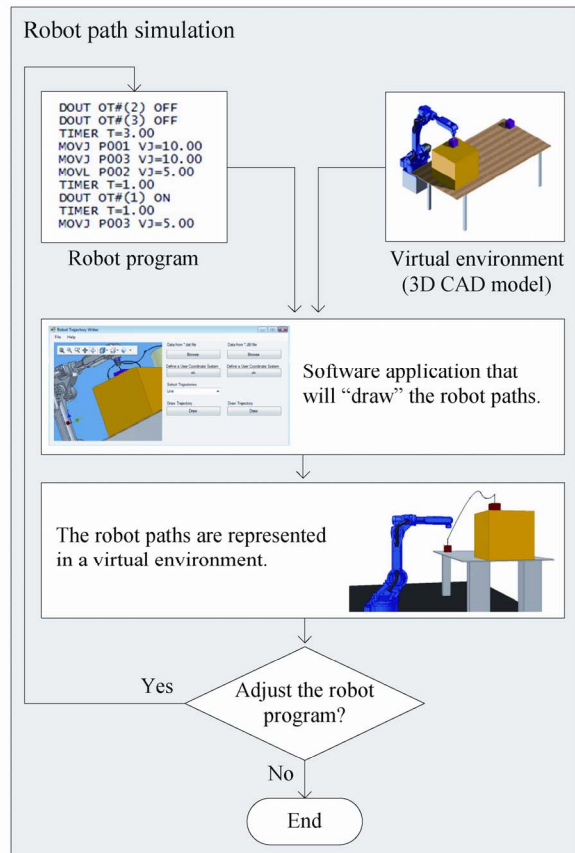


Figure 1. Workflow of the robot path simulation system.

The virtual robot paths are created through the help of an application programming interface (API) provided by Autodesk.

B. Application Programming Interface (API)

The Autodesk Inventor API exposes the Inventor’s functionalities in an object-oriented manner, allowing developers to interact with Autodesk Inventor using current programming languages (Visual Basic, Visual C#, Visual C++), for example, access to CAD data and create CAD models. There are different ways to access the API and it is important to choose the appropriate way to access it. In our simulation system, the API was used to “draw” robot paths in a CAD environment. A snippet of the code to “draw” the robot paths into the CAD environment is shown below (coded in Visual Basic).

```
' Create a collection to store the points in.
Dim oCurvePoints As ObjectCollection
oCurvePoints =
ThisApplication.TransientObjects.CreateObjectCollect
ion

' Set a reference to the transient geometry object.
Dim oTransGeom As TransientGeometry
oTransGeom = ThisApplication.TransientGeometry
```

```

' Set the points extracted from the robot program or
the data file.
Dim i As Integer
For i = 0 To number_of_points - 1
    oCurvePoints.Add(oTransGeom.CreatePoint(wpl.Point
        .X + n_matrix_P(i, 0), wpl.Point.Y +
        n_matrix_P(i, 1), wpl.Point.Z + n_matrix_P(i,
        2)))
Next

'Create a new part model.
Dim oTracePart As PartDocument
oTracePart = ThisApplication.Documents.Add(
    DocumentTypeEnum.kPartDocumentObject, , False)

' Insert the part into the assembly model.
Dim oTraceOcc As ComponentOccurrence
oTraceOcc =
m_oDoc.ComponentDefinition.Occurrences.AddByComponen
tDefinition( oTracePart.ComponentDefinition,
ThisApplication.TransientGeometry.CreateMatrix)

' Create a 3D sketch.
oSketch =
oTracePart.ComponentDefinition.Sketches3D.Add

' Draw lines, splines or circles.
oSketch.SketchSplines3D.Add(oCurvePoints,
SplineFitMethodEnum.kSmoothSplineFit)

```

C. Material Handling

The robotic system presented in this section was designed to perform the task of robot motion/path simulation for a material handling process, more specifically the movement of a piece from one location to another and the paths of approach and escape from the piece (Fig. 3). The robot program was generated off-line using data extracted from CAD models, without considering the kinematic limitations of the robot and collisions with the objects in the workspace of the robot.

The first thing when creating a simulation model of a manufacturing system, as a robotized cell, is the construction of each component model (in CAD). Depending on the simulation software platform the user can take advantage from pre-constituted manufacturing component libraries. However in most cases the desired component has to be designed by the user. When created, the simulated robot paths and workplace models may be viewed and manipulated in standard ways. Using our simulation system, the robot paths can be verified and optimized (errors can be identified and avoided) before the robot program be made to run into the real robot controller. The path visualization helps us to understand where we must make changes. Then, we can quickly edit the robot program (change tag points, movement type, reference frames, etc.) and “re-design” the robot paths. The example in figure 2 expresses the above mentioned. The robot paths generated off-line are simulated in a CAD environment (situation (a)) and visually analyzing the simulation, any collision is detected. However, if an obstacle is introduced into the environment (situation (b)) we can visually detect that the piece to be moved will collide with the obstacle. Thus, the robot program should be manually edited and the robot paths re-simulated until to get an acceptable result (situation (c)).

Accuracy is an important issue in any robot simulation process. Our simulator is governed by the accuracy with which

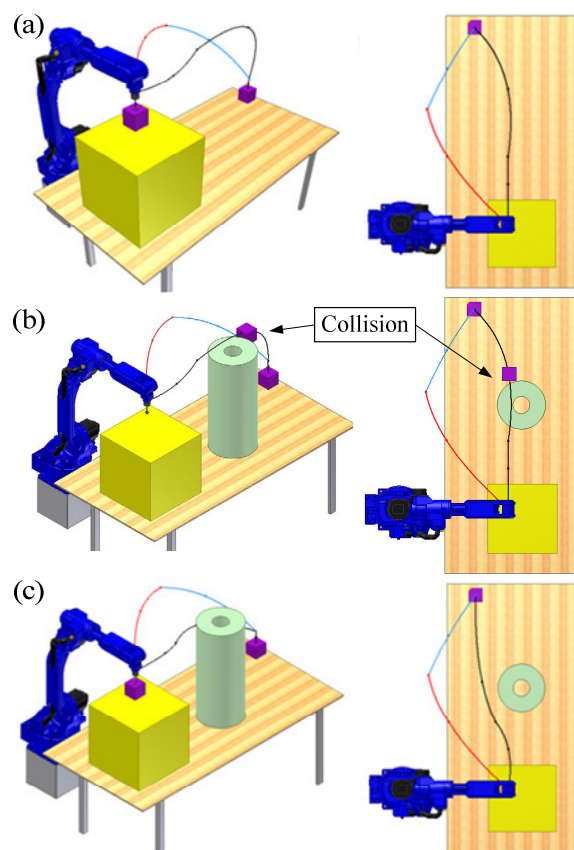


Figure 2. Simulated robot paths for a material handling task. The approach robot path (blue), the work path (black), and the escape path (red). All robot paths are “made” in “spline” motion, except when the robot makes the approach to pick up the piece or to release the piece; in this case it was used linear motion. In order to avoid the obstacle (situation b) the robot work path was changed (situation c).

the robotic cell is modeled. In other words, for example, if the CAD models have dimensional errors, the robot paths will appear in a CAD environment that do not reproduce the real environment, and in this situation the value of simulation is reduced. From the above experiment, we can conclude that the simulation system reproduce exactly the pre-programmed robot paths, off-line, and in a short time.

D. Footwear Industry (Shoe Sole Contour)

The robotic system presented in this section was designed to perform the task of robot motion/path simulation for a robotic operation in the footwear industry (shoe sole contour) (Fig. 3). The robot work involves thinning of unfinished shoe soles.

The robot program was obtained through a vision based system used to capture the points that represent the shoe sole contour. So, we have a set of data points collected off-line by the vision system and we need to test it. It was collected a big number of points and we need to select what points should be send to the robot, with the paradigm that if the number of points is increased the contour will be more perfect but at the same time the robot will take more time to execute the task. Thus, replacement accuracy depends on the distance between

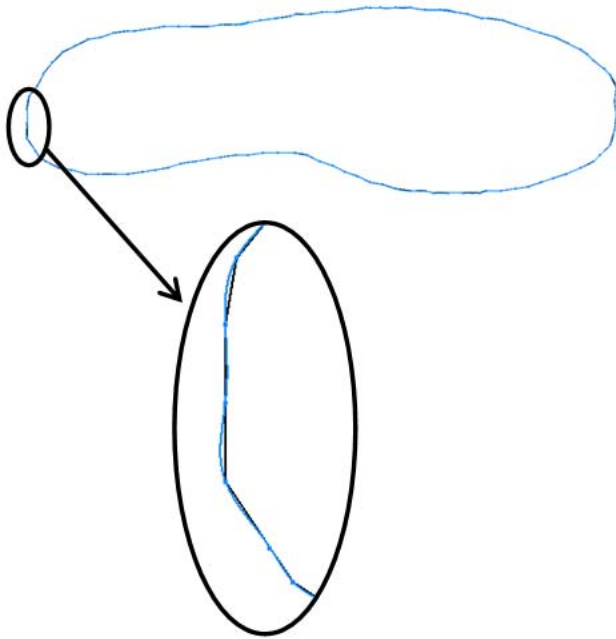


Figure 3. Simulation of the robot paths (shoe sole contour). The simulation allows to compare different types of robot motion, using straight lines (black) or splines (blue).

points, or better depends on the number of points we select. This way, in order to optimize the process and select the appropriate number of points becomes necessary to off-line simulate the robot paths in an easy way and in few minutes. Our simulation system also allows the junction of the points using different robot motion types, linear motion, circular motion, spline motion, and joint motion; allowing us to decide what is the best solution for the application in study. Sometimes the data extracted from the vision system have incurrance's and needs to be corrected; this can be achieved after make the path simulation. Given this example, starting from the robot program the complete simulation process takes about one minute.

E. Robotic Bending Cell

This experiment is in line with the above presented. The objective is to show another example of application, in this case it was simulated the paths for a robot operating in a bending cell (Fig. 4). One more time, the input to the simulation system is a robot program generated from a CAD model of the bending cell. In this specific application it is particularly important to generate and simulate robot programs off-line, since we know that tests in the shop-floor are time consuming. This way the system can be tested and improved off-line, allowing to reduce the time spent in the programming process.

Autodesk Inventor provides us the opportunity to fell immersed in the virtual environment, allowing to visualize models from any point in space, make zoom, etc. This is a valuable help when we are analyzing the virtual paths. The system does not simulate the orientation of the robot end-effector; however, the user can insert virtual robot end-effectors into the CAD environment with the desired orientation and by this way visualize its pose in conjugation with the robot paths.

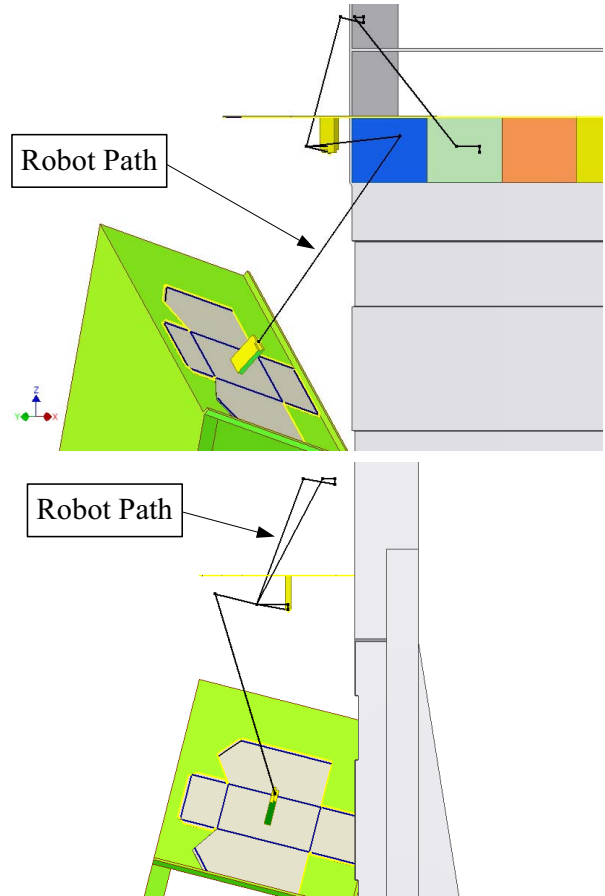


Figure 4. Simulated robot paths. The robot picks up a workpiece, move it to the "home position" and then places the workpiece in the position to be bended.

IV. RESULTS AND DISCUSSION

The experiments showed that the low cost robot path simulation system works very well, without producing errors and reducing the robot cell start-up time. The error that may exist in the simulation comes from inaccuracies in the calibration process and from situations where the CAD models do not reproduce properly the real scenario.

Since it was referred that the system is easy to use, it is also important to quantify/specify the time spent in each simulation process. It was found that the time required to generate robot programs through simulation and off-line robot programming is significantly lower that when is used on-line programming methods (using the teach pendant). This reduction in programming time depends on each specific application. In this way, the production has to stop for a time clearly lower than when used on-line programming methods. In our simulation process, the construction of the CAD model consumes a great deal of total time. Nevertheless, after the cell model is built, it can be quickly reconfigured to simulate a different task in few minutes. The simulation software proved to be efficient, fast, easy to use, and a good tool for analyzing robot paths in a virtual environment before performing tests with the real robot. The developed low cost robot path simulation system shows to be valuable and an interesting tool for SMEs.

Off-the-shelf robot simulators are excellent simulation tools with capacity to simulate not only robot paths but also the entire robot arm movement, with modules dedicated to specific tasks such as painting and welding and automatic detection of collisions. Despite this, they also have some disadvantages that hinder its use in companies, especially in SMEs. By comparing off-the-shelf robot simulators with the robot path simulator presented in this paper, we found that our system presents some advantages when compared with off-the-shelf simulators; its relatively low cost, short learning curve and ease of use. Moreover, the most time consuming task (the construction of the CAD model) is present in both systems, but since in our system the CAD models and simulation are performed in the same environment, Autodesk Inventor, this task becomes easier. In both systems, dimensional errors that CAD models may contain will lower the quality of simulation.

Developments in robot simulation systems offer great benefits in robot programming, robot simulation and human training, but some of these systems still have some limitations in terms of effectiveness, technology accessibility and reliability. Nevertheless, robotic simulation shows great promise.

V. FUTURE WORK

In future, the way the orientation of the robot end-effector is simulated should be simplified and automatic, in other words, the system should be able to represent the virtual robot end-effector with programmed orientation along the simulated robot path. Automatic detection of collisions is another issue to be studied.

VI. CONCLUSION

A new low cost CAD-based system dedicated to robot path simulation was developed and presented. This system allows users to simulate robot paths with a high-level of abstraction from the robot language. In our simulation software there's a lack of sophistication of the commercial and more mature packages, but is cheaper, easy to use and with a shorter learning curve. All of these three characteristics are fundamental when the objective is to spread the utilization of this software in SMEs and for educational purposes. The advantage of this approach is twofold. First, the robotic cell design and robot path simulation is embedded in the same interface and work through the same platform (Autodesk Inventor), without compatibility issues. Second, the same commercial software package that the engineer uses to design components is also used to simulate robot programs. Thus, robot simulation and product design can be integrated seamlessly. The system accuracy is governed by the accuracy of the CAD models that represent the robotic cell.

The effectiveness of the proposed system was proved through the experiments. Experiments showed that the CAD-

based system is intuitive and has a short learning curve, allowing non-experts in robot programming to simulate robot programs in just few minutes. Simulation in robotics has reached an important role, and we believe that in future simulation will make robotic systems more accessible.

ACKNOWLEDGMENT

This work was supported in part by the Portuguese Foundation for Science and Technology (FCT), grant no. SFRH/BD/39218/2007.

REFERENCES

- [1] J. N. Pires, K. Nilsson, and H. G. Petersen, "Industrial robotics applications and industry-academia cooperation in Europe," *IEEE Robotics & Automation Magazine*, vol. 12, no. 3, pp. 5–6, 2005.
- [2] Y. Yong, and M. Bonney, "Offline programming," *Handbook of Industrial robotics*, John Wiley & Sons, New York, Chapter 19, 1999.
- [3] L. J. Love, "Robot Simulation," *Robotics and Automation Handbook*, CRC Press, pp. 21-1–21-7, 2005.
- [4] G. C. Burdea, "Invited Review: The Synergy between Virtual Reality and Robotics," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, 1999.
- [5] L. Zlajpah, "Simulation in robotics," *Mathematics and Computers in Simulation*, vol. 79, no. 4, pp. 879–897, 2008.
- [6] R. Bernhardt, G. Schreck, and C. Willnow, "Development of virtual robot controllers and future trends," in *6th IFAC Symposium on Cost Oriented Automation*, pp. 209–214, Berlin, Germany, 2002.
- [7] T. Yagi, "Industrial Robots in the Japanese Automobile Industry-Robot equipment trends in a time of steady growth," *Industrial Robot: an International Journal*, vol. 33, no. 5, pp. 359–364, 2006.
- [8] R. Bernhardt, G. Schreck, and C. Willnow, "Realistic robot simulation," *Computing & Control Engineering Journal*, vol. 6, no. 4, pp. 174–176, 1995.
- [9] X. F. Zha, and H. Du, "Generation and simulation of robot trajectories in a virtual CAD- based off-line programming environment," *International Journal of Advanced Manufacturing Technology*, vol. 17, no. 8, pp. 610–624, 2001.
- [10] U. Berger, R. Lepratti, and M. T. May, "An approach for the automatic generation of robot paths from CAD-data," in *Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2005)*, pp. 291–297, Catania, Italy, 2005.
- [11] S. F. Chan, and R. Kwan, "Post-processing methodologies for off-line robot programming within computer integrated manufacture," *Journal of Materials Processing Technology*, vol. 139, no. 1–3, pp. 8–14, 2003.
- [12] C. C. Ruokangas, W. A. Cuthmiller, B. L. Pierson, K. E. Sliwinski, and J. M. F. Lee, "Off-line programming motion and process commands for robotic welding of space shuttle main engines," *Journal of Robotic Systems*, vol. 4, no. 3, pp. 355–375, 1987.
- [13] M. Bruccoleri, C. D'Onofrio, and U. La Commare, "Off-line programming and simulation for automatic robot control software generation," in *Proceedings of the 5th IEEE International Conference on Industrial Informatics*, pp. 491–496, Vienna, Austria, 2007.
- [14] D. Zhijiang, Y. Donghai, S. Lining, and F. Lixin, "Virtual Robot Simulation and Monitoring System Based on Java3D," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO 2004)*, pp. 313–317, Shenyang, China, 2004.