Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 779–785

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

CAD-based robot path planning and simulation using OPEN CASCADE

Amit Kumar Bedaka[a], Chyi-Yeu Lin[a]

[a]*Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei City 10607, Taiwan (R.O.C)*

Abstract

In manufacturing industries, human operators are required in many manufacturing processes, which is time-consuming and not cost-effective. In the present approach, a CAD-based off-line programming (OLP) platform is developed based on OPEN CASCADE (OCC) open source libraries. The developed platform works as a human-robot interface (HRI), offers friendly interaction in an intuitive way, so that in few minutes with basic information, any user can generate a robot path from a CAD model and visualize the simulation graphically. The platform provides important steps such as loading CAD models to define a task, extracting CAD information to generate a path, checking the reachability of the manipulator, and simulation to check and prevent the possible collision before generating a robot program. To evaluate the industrial usability of the proposed platform, a robot program for gluing application is generated from simulation and mapped to a real 6-Degree Of Freedom industrial manipulator.

Keywords: Robot path planning; offline programming; open cascade; cad model, simulation platform; glue dispensing

1. Introduction

Manufacturing industries have many challenging applications, which require the skills and craftsmanship of a human operator to perform various tasks. Certainly, some of these applications are predefined and highly repetitive, hence carried out by industrial robots. Applications such as pick and place, welding, painting and gluing are some of the basic tasks. To perform these types of applications robots are usually programmed using the conventional method (a teach pendant), which is a user-friendly approach and doesn't require a human operator with high programming skills. However, this approach is laborious and time-consuming when modifying the robot path. This type of automation requires more effort to generate robot path for complex geometry and is not cost effective in manufacturing industries.

* Corresponding author. Tel.: +(886)-978-745-471.
E-mail address: akbedaka@gmail.com

10.1016/j.procs.2018.07.119

The prospect of time and difficulties associated with reprogramming of industrial robotic systems in a manufacturing scenario provides an opportunity to replace the conventional method with a CAD-based OLP approach [1, 2]. In this respect, OLP software gives a possibility to represent graphically and model the robot and other necessary equipment, extract the robot program automatically from a CAD model and simulate a program in the virtual environment before mapping on a real system. In recent years, many products have been availed commercially and used in research fields such as KUKA Sim for Kuka, RobotStudio for ABB, MotoSim for Motoman Delmia from Dassault Systems, RobCAD from Technomatix Technologies and Robotmaster from Jabez Technologies. In particular, the OLP system in [3] uses geometric functions of CATIA (e.g., curve/surface intersection, a projection of the points onto the surface etc.) and simulation function of KUKA Sim Pro (e.g., robot kinematics, collision detection etc.). It is focused on robotic drilling applications in aerospace manufacturing to improve the position accuracy by using bilinear interpolations model and redundancy resolution. In [4], the effectiveness of automated OLP techniques for the robotics welding process is demonstrated and the evaluation of several OLP software packages, such as ABB Robotics, Delmia from Dassault Systems and a Matlab based OLP system and RinasWeld from Kranendonk Production System is presented. Therefore, a significant amount of research and development on OLP system is undergoing to reduce the programming efforts and capital investment to make them feasible with low volume production. Moreover, all these commercial OLP platforms need one to buy a license and are difficult to customize. These limitations have opened opportunities for researchers to generate innovative solutions.

Recently, relatively low cost and commercially available 3D CAD packages were used to visualize a robot path in the CAD environment. The CAD-based OLP presented in [5] focuses on the representation of robot motion in a CAD drawing and its automatic extraction of data to make the mapping between the CAD model and the real environment as well as the automatic generation of robot program. In [6], a simulation platform developed from 3D standard CAD package Autodesk Inventor [7], wherein the robot path and program were generated directly from the CAD drawings not from the CAD features. Recently [8] the IRoSim platform is developed to integrate mechanical CAD features with robotics CAD, which is based on the SolidWorks API. All these approaches rely on the standard 3D CAD software that require additional packages for simulation platform development CCB and still licenses to be bought. Now, these shortcomings are addressed in the present work that seeks to develop a low-cost platform which generates a robot path by utilizing open source OCC from the CAD features automatically.

In this paper, a novel CAD-based OLP platform is developed using OCC libraries with Microsoft Visual Studio. The developed HRI platform is intuitive and offers friendly interaction to generate a robot path from CAD information, to visualize the simulation graphically with basic information in few minutes. In addition, the proposed platform provides several features such as loading of the CAD model for task definition, generating a robot path, manipulator's accessibility verification, simulation to check and prevent a collision before converting into robot program. Further, in order to show the effectiveness and robustness of the proposed approach, simulation of a glue dispensing task is performed using a 6-DOF manipulator and mapped with a real industrial robot. The experimental results showed that the proposed platform is robust and easy to use with important role in robotics. This innovative platform can assist many industries to achieve a higher level of intelligent automation and offer a low-cost virtual environment for developing and analyzing robot motions.

2. Overview of the proposed CAD-based system

CAD-based offline robot simulation platforms using 3D CAD packages such as SolidWorks API [8, 9] and Autodesk Inventor [5, 6, 7] have been investigated to perform the simulation for various industrial applications [5, 10]. In the present approach, OCC libraries [11] are used to develop the application-oriented platform. Fig. 1 shows the overview of the proposed CAD-based system, that allows a user to define and generate a path using the CAD features (face, wire, edges, etc.) and simulate a given industrial task (glue dispensing) in the virtual environment. The development process of the proposed system consists of various steps briefly explained in sections below such as extraction of the CAD information using the OCC CAD kernels to generate position and orientation for the robot path. Finally, the end-effector of the 6-DOF industrial manipulator follows the generated path automatically from the CAD features to perform simulation. This new approach of generating a robot path using CAD features and OCC libraries is explained in [12].

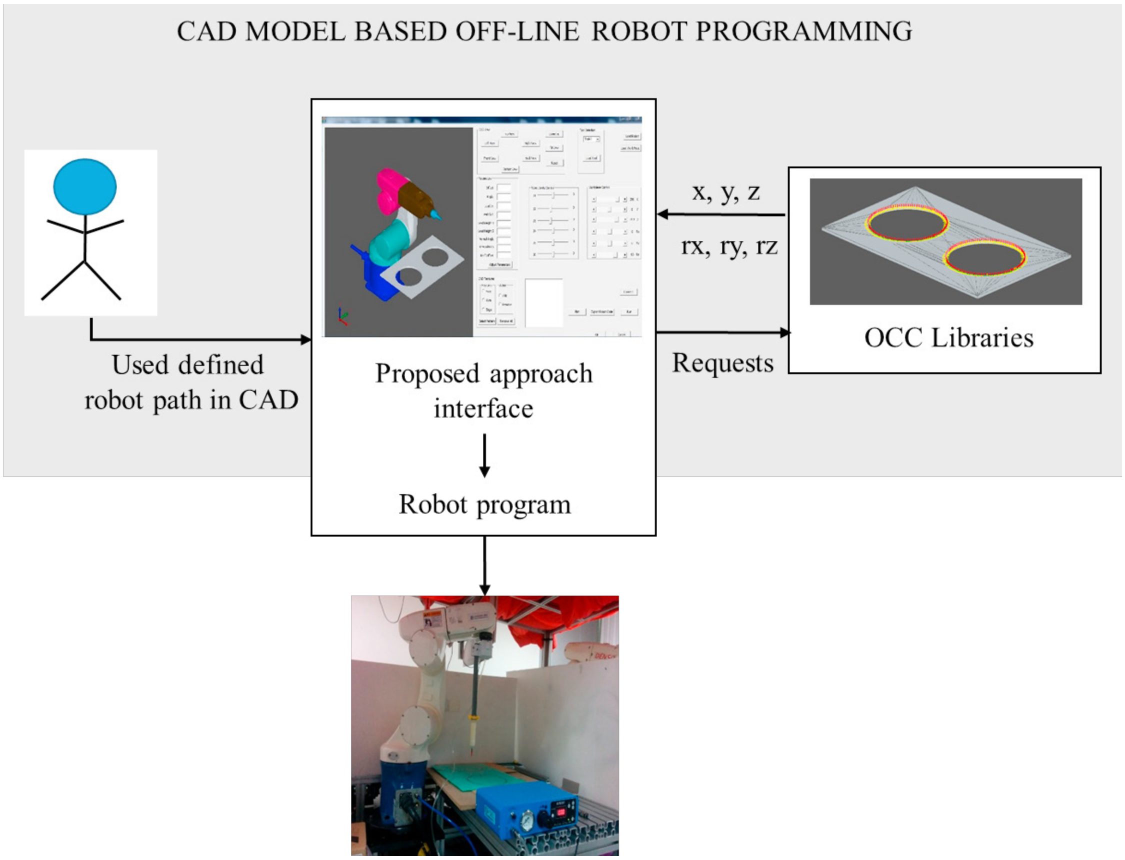


Fig. 1. Overview of the proposed CAD-based system.

In particular, a Denso 6556 robot has been selected to perform the gluing task as shown in Fig. 1 with the glue dispensing setup. After the simulation, results are sent automatically to the Denso robot to perform the task and to check the effectiveness of the results without any external aids.

2.1. Method

The process begins with the extraction of information from the CAD model. It is important to know the kind of information necessary for generating a robot path and how to extract the same. In the present approach, OCC CAD kernels are used to extract the position information from CAD. Based on the robot end-effector frame, the orientation of the workpiece is defined by a transformation matrix. The complete process of CAD data extraction and orientation required for a robot path simulation and generation of a robot program is explained in the sections below.

2.2. Position

In the present approach, CAD features are employed to extract position information using OCC library. Different classes are used to extract the position information from a CAD model. Prior to this a user needs to select a face/wire/edge to start this extraction process as explained in the Fig. 2. Upon selection, the class BRepBuilderAPI_MakeWire is used to convert the shape topology into geometry. During this stage, points are created on the curve as shown in Fig. 2 using the class GCPnts_AbscissaPoint that require a wire to be converted into a curve using the BRepAdaptor_CompCurve class. This converts the shape topology information into geometry. The Geom_surface Class converts the shape to a surface. The normal vectors obtained from the cross product of U and V parameters on

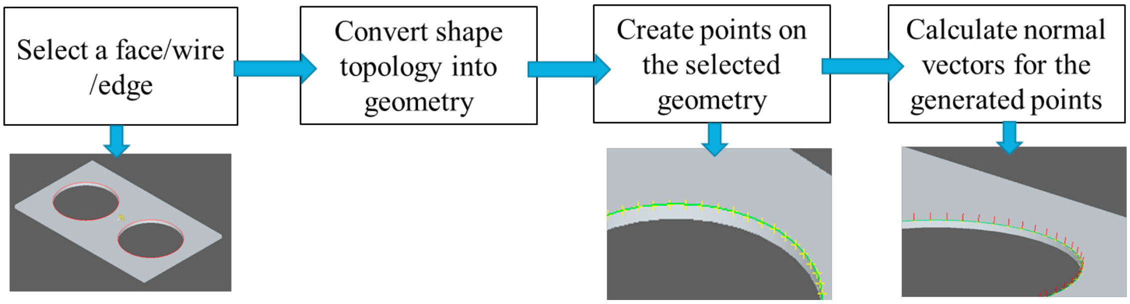


Fig. 2. Steps to generate points on a user defined path.

the surface, as shown in Fig. 2, are represented by red bars. Similarly, the **x**-direction for the defined points on the curve is represented by green bars. The normal vector serves as a reference for the robot pose perpendicular, whereas the path direction is employed to calculate the x-direction to fix the orientation of the robot.

2.3. Orientation

To perform robot simulation, both the position and the orientation information are needed. The position information of a workpiece is obtained using the OCC CAD kernels. Similarly, the orientation of the workpiece *w.r.t.* a given frame from the end-effector is defined by a transformation matrix. The orientation of the end-effector *w.r.t.* the workpiece is obtained in the form of α, β and γ as shown in eqs. (2), (3) and (4). The special case of singularity [13] is solved by assuming $\alpha = 0$ and $\beta = +90°$ or $-90°$ in eq. (7).

$$T = \left[\begin{array}{c|c} R & P \\ \hline 0\ \ 0\ \ 0 & 1 \end{array}\right] \tag{1}$$

where, R is the direction cosine matrix as shown in eq. (6) to describe the robot orientation; $P = (P_x, P_y, P_z)^T$ the vector to describe the reference point of the robot position.

$$\alpha = a\tan 2(r_{32}, r_{33}) \tag{2}$$

$$\beta = a\tan 2(-r_{31}, \sqrt{r_{11}{}^2 + r_{21}{}^2}) \tag{3}$$

$$\gamma = \tan^{-1}(r_{21}, r_{11}) \tag{4}$$

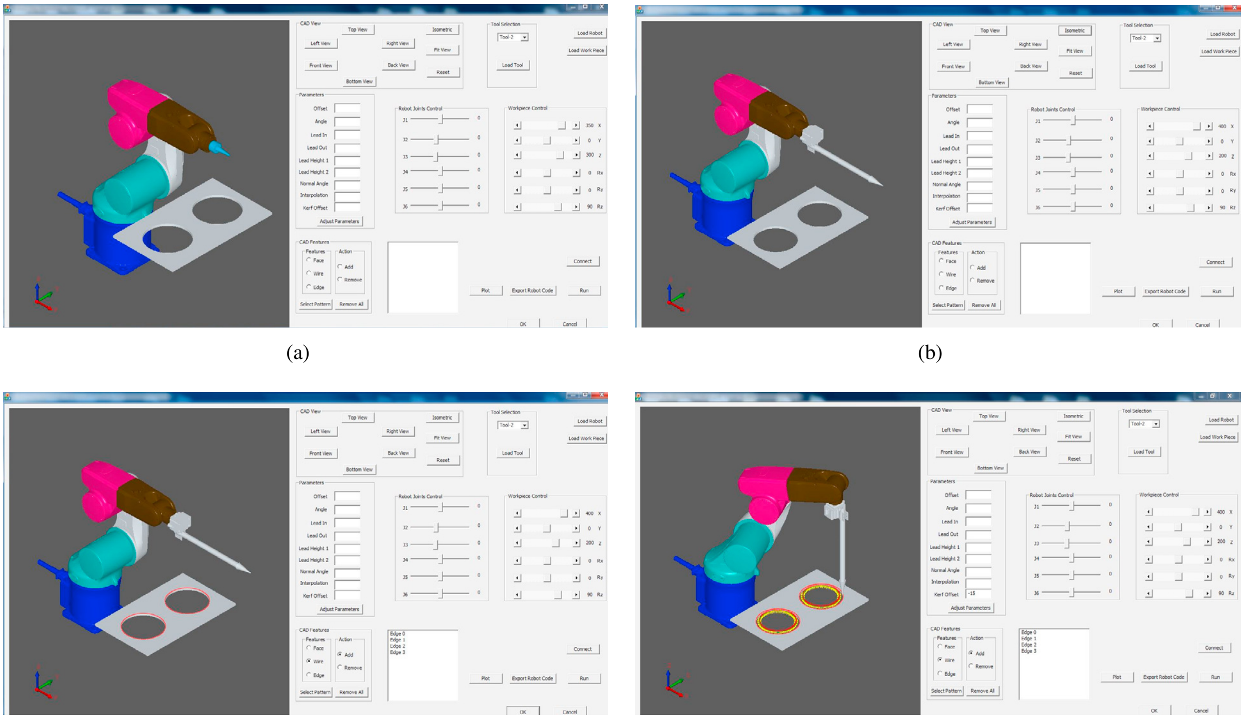
Here, r_{ab} represent the elements of the rotation matrix and $a\tan 2(y, x)$ a two-argument arc-tangent function that yields one unique solution for the angle. In addition, α represents the rotation angle about the x-axis, β, the y-axis and γ, the z-axis

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{5}$$

$$R = \begin{pmatrix} C_zC_y & C_zS_yS_x - S_zC_x & C_zS_yC_x + S_zS_x \\ S_zC_y & S_zS_yS_x + C_zC_x & S_zS_yC_x - C_zS_x \\ -S_y & C_yS_x & C_yC_x \end{pmatrix} \tag{6}$$

where, "C" and "S" refer to Cosine and Sine functions, respectively.

$$\alpha \pm \beta = a\tan 2(r_{23}, r_{13}) \tag{7}$$



(a) (b) (c) (d)

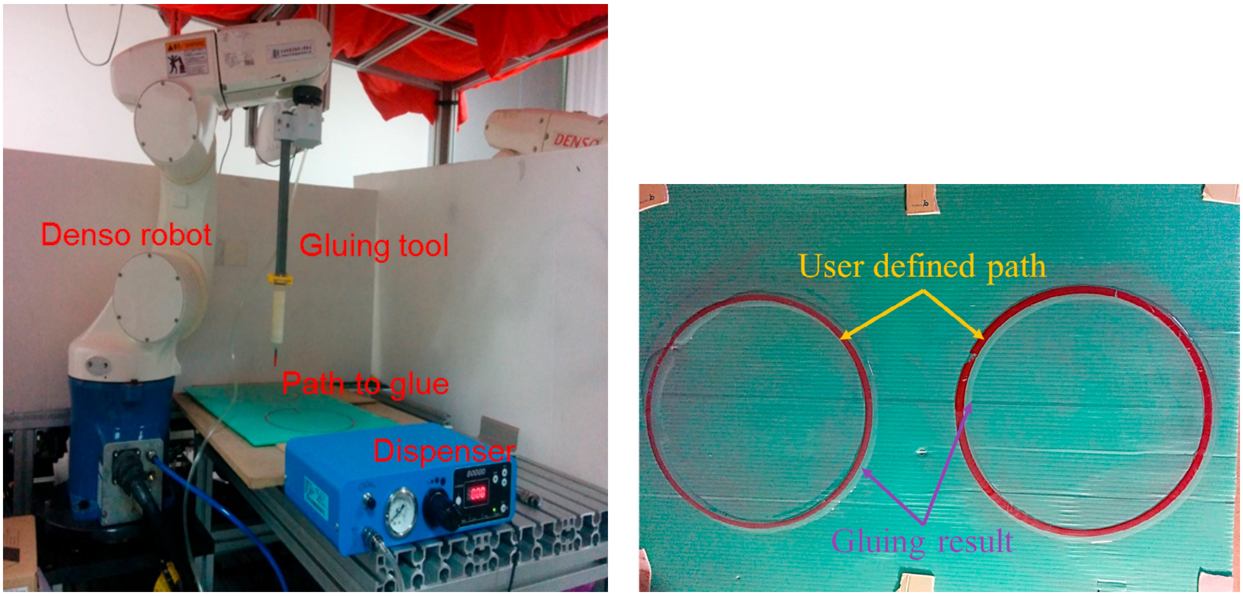
Fig. 3. (a) developed offline platform with a robot and a workpiece; (b) glue dispensing tool; (c) user selected path to generate robot path; (d) generated robot path and simulation result.

2.4. Simulation

Simulation is one of the major steps for visualizing and checking the effectiveness of a generated robot path in the CAD environment. Position information is obtained from the CAD model using the OCC kernels, while orientation from the euler angles *w.r.t.* the end-effector frame. These positions and orientations are input into the robot inverse kinematics for transforming the joint coordinates utilizing eq. (1). As a result, individual angles for each joint of the manipulator are obtained to perform the simulation. Before performing the simulation, the position of the workpiece is defined depending on the working environment. Finally, the simulation is executed for the user-defined path. This step in the OLP platform gives a clear view of the process, before generating a robot program for a real manipulator. The benefit of this tool is to verify the process without the need of any physical robot and reduces the risk associated with the wrong implementation in the final stage.

3. Simulation results for the glue dispensing

The proposed platform incorporates all the steps mentioned, such as position and orientation, to graphically visualize the simulation. Then the accessibility of a path is checked to ensure its reachability. Where the generated path is inaccessible, the workpiece position has to be redefined. In this case, the position of the workpiece is defined as per the workspace available in the laboratory as shown in Fig. 3(a). Fig. 3 illustrates the definition of the task path and the simulation for the defined path. Fig. 3(a) shows a snapshot of the platform with the various options to input for different industrial applications. The proposed platform gives an option for a user to change the tool *w.r.t.* the industrial tasks In this case, the glue dispensing tool is used to perform the given task as shown in Fig. 3(b). Fig. 3(c) shows the user selected path to perform the proposed task and path is generated with 15mm kerf offset (because of the tool



(a) (b)

Fig. 4. (a) complete experimental setup; (b) gluing result.

offset) as shown in Fig. 3(c) to perform the simulation. Finally, the simulation is shown in Fig. 3(d) to achieve the goal.

4. Experimental setup and results

The experimental setup was designed as shown in Fig. 4 to perform the manipulation task generated from the CAD-based simulation platform. The complete setup consists of 6-DOF Denso 6556 robot, controller, PC (developed platform) to perform the simulation and to control the robot, glue dispensing set, and coordinate measuring machine. To perform the experiment, the input (script) is sent directly from the platform to the Denso robot using the RS232 connector through the controller to perform the defined task. A program for the controller was built in the WINCAP-III using boost libraries in advance. The coordinate measuring machine is used to place the workpiece precisely in the workspace as shown in Fig. 4(a). The output data from the simulation platform is used as the input for the controller. As a result, the robot follows the red color line as shown in Fig. 4(b), *i.e.*, user defined path for the gluing task. The result of the gluing application in Fig. 4(b) shows the effectiveness and the robustness of the proposed platform to perform any kind of manipulation task in near future. The proposed approach developed an offline platform using OCC libraries which is capable of loading CAD models, generating a path for a defined task using CAD features (face, wire, edge, etc.), perform simulation to check the preformation of the robot in the virtual environment. Also, the platform sends the robot script directly to a robot without using any external aids. Finally, the experimental results show that the generated path is efficient to compute and easy to execute using the proposed platform with basic information in few minutes. The aforementioned results promise to perform a variety of applications without significant errors. Most of the errors occur due to the CAD model geometry and robot calibration. This shows the present approach is suitable to develop according to the application requirement from scratch with low investment and lesser development time.

5. Conclusion

In this paper, a novel CAD-based robot path planning and simulation platform was developed independently to simplify and consolidate the development by using OCC libraries with MFC dialog based Visual Studio. The proposed HRI platform fully utilizes the CAD information to generate a defined path, embed with the CAD models of a machine tool and transform them into the robot path to perform the assigned tasks. The developed platform has its own limitation in terms of sophistication and ability to generalize for other applications. On the other hand, it is low-cost, intuitive, and offers friendly interaction. The effectiveness and robustness of the proposed simulation platform were proved through the gluing experiment. The Experiment showed that a user is able to generate a path and perform a given task with basic information, and then map it to the industrial manipulator within few minutes. This low-cost platform accords small manufacturing industries the opportunity to develop an OLP platform according to their application requirements with low investment together irrespective of the training level of their operators.

In the near future, the development of the platform to enhance its functionality in different applications with various robot systems will be investigated.

Acknowledgements

The work of this paper is supported by Ministry of Science and Technology, Taiwan (ROC) under the grant MOST 104-2221-E-011 -059 -MY3. .

References

[1] Zengxi Pan, Joseph Polden, Nathan Larkin, Stephen Van Duin, and John Norrish. (2011) "Recent Progress on Programming Methods for Industrial Robots." *Robotics and Computer Integrated Manufacturing* **28** (2): 87–94.

[2] Bjorn Hein, Martin Hensel, and Heinz Worn. (2008) "Intuitive and model based on-line programming of industrial robots: a modular on-line programming environment." *IEEE International Conference on Robotics and Automation*: 3952–3957.

[3] Weidong Zhu, Weiwei Qu, Lianghong Cao, Di Yang, and Yinglin Ke. (2013) "An off-line programming system for robotic drilling in aerospace manufacturing."*Int. J. Adv. Manuf. Technol* **68** (9-12): 2535–2545.

[4] Nathan Larkin, Aleksandar Milojevic, Zengxi Pan, Joseph Polden, and John Norrish. (2011) "Offline programming for short batch robotic welding." 16^{th} *Joining of Materials (JOM) conference*: 1-6.

[5] Pedro Neto, and Nuno Mendes. (2013) "Direct off-line robot programming via a common CAD package."*Robot. Auton. Syst* **61** (8): 896-910.

[6] Pedro Neto, Nuno Mendes, Ricardo Araújo, Norberto Pires J., and Paulo Moreira A.. (2012) "High-level robot programming based on CAD: dealing with unpredictable environments."*Industrial Robot: An International Journal* **39** (3): 294-303.

[7] Pedro Neto, Norberto Pires J., and Paulo Moreira A.. (2010) "Robot path simulation: a low cost solution based on CAD."*IEEE Conference on Robotics Automation and Mechatronics (RAM)*.

[8] Baizid K., Ćuković S., Iqbal J., Yousnadj A., Chellali R., Meddahi A., Devedžić G., and Ghionea I.. (2016) "IRoSim: Industrial Robotics Simulation Design Planning and Optimization platform based on CAD and knowledgeware technologies."*Robotics and Computer-Integrated Manufacturing* **42** : 121-134.

[9] Hongmin Wu, Huajian Deng, Cao Yang, Yisheng Guan, Hong Zhang, and Hao Li.(2015) "A robotic off-line programming system based on SolidWorks."*IEEE International Conference on Robotics and Biomimetics (ROBIO)*.

[10] Heping Chen, Weihua Sheng, Ning Xi, Mumin Song, and Yifen Chen.(2002) "CAD-based automated robot trajectory planning for spray painting of free-form surfaces."*Industrial Robot: An International Journal* **29**(5) : 426-–433.

[11] Open Cascade Technology. www.opencascade.com.

[12] Amit Kumar Bedaka and Chyi-Yeu Lin.(2015) "Autonomous path generation platform for robot simulation."*International conference on Advanced Robotics and Intelligent Systems (ARIS)*.

[13] Jose-Luis Blanco (2015) "A tutorial on SE(3) transformation parameterizations and on-manifold optimization", *Technical report*, University of Malaga.