

The CAD drawing as a source of data for robot programming purposes – a review

Krzysztof Foit^{1,*}, and Grzegorz Ćwikła¹

¹Silesian University of Technology, Faculty of Mechanical Engineering, Institute of Engineering Processes Automation and Integrated Manufacturing Systems, Konarskiego 18A St., 44-100 Gliwice, Poland

Abstract. Industrial robots are widely used in mass production, successfully replacing the workers during repetitive tasks. The lack of direct human participation in the production process involves the necessity of programming of the machines, which is mostly the time-consuming process. Most of the operations are well described in the process documentation, but some actions require the use of complicated paths, like for example welding, laser or water cutting, edge grinding, application of liquid sealants or films of paint etc. All of the mentioned activities need the precise movements of robot's effector and this is - in many cases - connected with the necessity of entering of many points that approximate the path. One of the ways to solve this problem is the possibility of generating the complex path automatically, using the information from the technical documentation. This paper presents such approach that uses the CAD drawings as a data source. The available literature presents this problem mainly in the aspects of large software applications, but this paper was focused on presenting the problems of the proper analysis of the drawing and how to process the data to the form that can be used during robot programming.

1 Introduction

The history of industrial robotics started in the mid of 20th century, together with the first robot, developed by George Devol. At the beginning robots were used for manipulation task only: the first Unimate machines were working in GM factory and were used to stack the hot pieces of metal moved from the die-casting machine. Soon the robots were adapted to the other works, like painting, welding etc.

Performing of handling tasks is relatively simple activity that is carried out by robots. Basically, it is required only to define particular points that should be reached during the movement of the gripper carrying out the part. Of course, an important issue is to ensure collision-free movement of the individual links of the manipulator, while performing the transportation task. If there are only a small number of obstacles in the robot's workspace, the basic knowledge of the rules of motion programming is often enough to do the job properly. These tasks often use the simplified method of trajectory definition called "point-

* Corresponding author: krzysztof.foit@polsl.pl

to-point." In this method, the operator defines the key points that must be reached in exact or approximate way. The movement of the particular links of manipulator's kinematic chain is calculated by inner algorithm of robot's controller and the operator has very limited influence on the shape of resulting trajectory. The "point-to-point" method could use both on-line and the off-line programming. On the other hand, tasks related to painting, welding, cutting, etc. usually require precise definition of the path followed by a moving tool. Such approach is called "continuous point" method of trajectory definition and requires a large number of points to be generated and stored in the controller memory. In order to define the path manually, the demonstration is used: the wrist of the robot is led by the operator along the path, while the controller samples the position of kinematic chain links and store them in memory. The off-line path generation involves the use of special, dedicated software. Because the use of complex path increased during time, the modern robot controllers often offer support for advanced paths, like parametric curve definition, curve fitting and interactive curve design.

The relationship between CAD software and robot programming has not always been clear. In the early years of robotization the tasks conducted by robots used simple trajectories with a small number of points. Another milestone was the use of robots for painting or welding. It soon became apparent that programming of complex trajectory is very time-consuming process, and the quality of the paint coating or weld is largely dependent on operator skills. Thus began the search for efficient methods of designing the complex paths. Market launch of the PC-class computers and the rapid development of software for engineers, made possible to develop the methods for off-line programming of the robot, which also had a beneficial effect of short-term exclusion of the robot from the production process. Over the time, the researchers began to take into account the use of CAD software as one of the tools for preparing a program for the robot.

This paper is focused on presenting the possibility of using CAD software and the documents created by these applications as a source of data used in robot programming process. The research trends and problems are presented. Finally there are presented the selected solutions that are the results of works conducted in the Institute of Engineering Processes Automation and Integrated Manufacturing Systems.

2 The background of the idea

Off-line programming of a robot brings a number of advantages. The most important is the shortest time of machine inactivity during the realization of the production process. Others, less significant from the manufacturing process point of view, are [1]:

- Increased safety – programmer is not performing in the real environment and the developed program can be tested using simulation software,
- More comfortable conditions for programming – the work is moved from the factory floor to the office room.

Of course there are also disadvantages, like for example [1]:

- the necessity to buy a proper software and costs of personnel training,
- differences between robots that result in differences in accuracy and positioning errors,
- the necessity to perform the tests on real machine in the isolated environment.

These disadvantages are the background for the other problem, which is the strict connection between robot and off-line programming software – robots manufacturers often deliver dedicated application for the particular type of robot. Of course, there are some generic off-line programming and simulation systems on the market, like RobCAD or Robotmaster, but they are rather expensive.

It is worth to notice that CAD software has become more popular over the years. Today the electronic documentation replaced the paper documents and the same is in the field of

engineering. Neto and Mendes say that «it is very common to see a human being explaining something to another human being with base on a CAD drawing» [1]. The increased interest in CAD techniques has started the search for the relationship between electronic drawings and creation of programs for the machines involved in the production process – the very popular manifestation of such activity is the elaboration of the G-code, which has become a standard for programming of the CNC machines. This is the clear example of how the data from the CAD drawings could be used in the manufacturing process. However, doing the same in the field of robotic is almost impossible, because there is no standard language that could be used to program any robot – almost every manufacturer uses its own programming language. Such fact makes the use of CAD data for robot programming purposes quite hard but not impossible. The need for such research is justified, among other things, by high prevalence of CAD systems that offer ample opportunities to extend the functionality of application by using APIs and scripting programming languages. This topic is also the subject of works conducted by scientist from all around the world.

3 The examples of CAD – robot interfaces implementation

The main purpose of the CAD – Robot Interface is to use the data entered into the CAD program during creation of technical drawing as a source of coordinates used in the robot program. It should be noted that the drawing of the item is not used directly, but as a base to determine the characteristic points of the robot path (data for the trajectory planner software) or to designate the path directly in the CAD program. The most commonly used methods of obtaining coordinates from CAD drawings include:

- processing the drawing data by running scripts within the CAD application,
- passing on the data from the CAD application to the external program by using the CAD application API,
- exporting a drawing to the one of universal formats (the most frequently used is the DXF standard) and processing the resulting file in an external application.

The mentioned methods could be mixed in order to achieve optimal results, depending on the type of a CAD program and its capabilities.

An interesting example of the use of data from CAD system for programming the robot has been proposed by Kim [2], who has described the system for application of the glue during the process of connection of soles to the shoes. The author has elaborated the script for AutoCAD, written in AutoLisp language. The trajectory was generated on the basis of four polylines that describes the points, separately for the XY and YZ plane. The polylines has been extracted from the 2D drawing that consists of three views of the sole. Moreover, the author has also proposed the second solution, where the path was generated on the basis of data acquired from the 3D scan of the shoe. The data from the scanner has been imported to the CATIA software, where the trajectory line has been drawn point by point, using the mouse. After saving the path to the IGES file, the data for the robot program have been extracted using author's "Iges to Point" program. Of course, there is the need to do some additional work, like calibration and scaling, however author has claimed that conducted experiments confirmed the usefulness of the methods.

Another approach has been presented by Neto et al. in the works [1,3-4]. Authors have recounted the process of acquiring the data from the CAD system using scripts and API. As a source application they have used Autodesk Inventor. The software has been supported by external programs, written by authors of the paper. From the inside, Inventor application has been extended by scripts, written in Visual Basic for Application language. On the other hand, the definition of the trajectory was realized by drawing a line (a virtual path), like in the previously mentioned example. Authors also have pointed out the main

limitation of their interface that is the collaboration only with the particular type of robot, because of lack of the universal language that could program any robot.

The third example of CAD-based Human-Robot Interface has been developed for a welding robot. Pires et al. [5] have presented the system based on AutoCAD. Again the desired trajectory was represented by line placed on the drawing, however there is a possibility to insert all the necessary parameters (like voltage, velocity, torch distance from plate etc.) directly – they were introduced as labels with the corresponding value. After exporting to the DXF format, the drawing was processed by two applications: the first one for extracting the point data from DXF, while the second one for adjusting the data for performing the simulation of welding.

The presented examples have some common properties. First of all, in all cases the trajectory should be directly mapped in drawing as a segmented polyline or a curve. Moreover, all of the mentioned examples are dedicated to a particular type of robot. Finally, it should be noted that the authors put emphasis on data adjusting process – particularly on accuracy and calibration.

4 Research activities

Research on the use of CAD systems in the process of programming the robots have been also carried out at the Institute of Engineering Processes Automation and Integrated Manufacturing Systems. In parallel, the study related to the milling with the use of robot arm [6] and related to the planning of collision-free path of robot's effector [7-8] have been conducted.

In regard to the use of CAD systems for robot programming, three concepts have been developed. All tests were conducted on a Mitsubishi Movemaster RV-M1 robot.

The Mitsubishi Movemaster RV-M1 robot has a small, five axis manipulator with about 1 KG of lifting capability. The controller has the ability to communicate with the computer using Centronics port for one-way data flow or RS-232 port for two-way data flow. It means that using the RS-232 port, the operator can send the commands to the controller, but – in addition – can also download the data. The controller's memory is divided into the two partition: one is used for storing the points of trajectory, while the second stores the program. The robot is using its own programming language, so any software must be adapted to support it. For this reason, all the software developed during the experiments work only with Movemaster RV-M1 robot so far.

4.1 The “CAD TO RV-M1 DXF” translator for AutoCAD

The first CAD – robot Interface for the Movemaster robot has been developed as a result of the simple experiment. The interface named CAD TO RV-M1 is a simple, external processor of the DXF file. The main idea of the approach was to draw the trajectory in the AutoCAD, using lines for approximating the point-to point trajectory and complement it with the point entities, which represent the state of the gripper. The processor uses a « dictionary » file for translating the DXF file entities into the simple program, which could be realized by the robot. The concept of the dictionary was also intended to allow the adaptation of the processor program for using it with other types of robots. The block diagram of the developed CAD TO RV-M1 processor is shown in Figure 1.

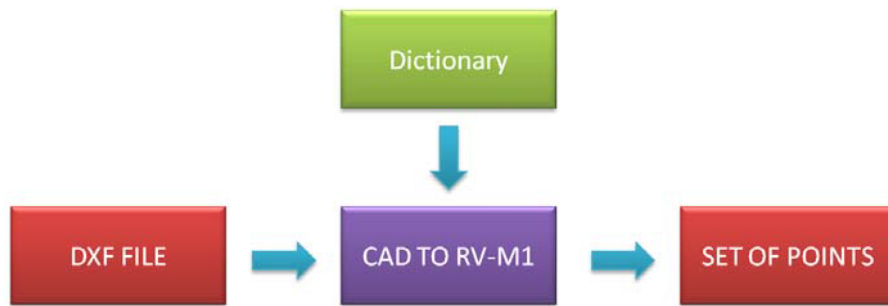


Fig. 1. Block model of CAD TO RV-M1 program data flow

Later, the CAD TO RV-M1 program has become the part of ROBBO package, which is a Mitsubishi Movemaster programming and simulation system, developed mainly for didactic purposes [9].

4.2 Retrieving the trajectory data from the SVG file

The method of retrieving the trajectory data from the SVG file has been created as the alternative to the method that use DXF file, which was described in section 4.1. The SVG acronym stands for Scalable Vector Graphics and it is the standard created mainly for World Wide Web, but it is open, free to use and it continuously evolves. The file contains a drawing coded in the text form, using the XML (Extended Markup Language) standard. In fact, many graphics programs can import and export the SVG format. The XML notation introduces the opportunity to use XML parsers to extract the trajectory data from the drawing. The SVG standard has been also used as the part of Human-Robot Interfaces [10-12].

The idea to use SVG graphics for robot programming has been presented in [13-14]. The standard is supported by many application, including the ones distributed on free licence (GNU GPL) – this definitely lowers the costs of implementation of the method. The drawing coded in the SVG standard could be easily converted to the other format – vector or raster one – using one of widely available converters. The significant limitation of the SVG manifests in its 2D character – the points are defined on the plane – but it is not a problem as long as the effector also moves on the plane. The second limitation is that the SVG cannot be easily converted to the set of points, which could be included in the robot's program. In [14] the author proposed the solution of this problem by doing prior conversion of SVG file to the raster graphics (bitmap) file or HPGL file. The first solution assumes that the plane, where the robot operates, should be divided into quadratic cells that will be mapped to corresponding pixels of raster file.

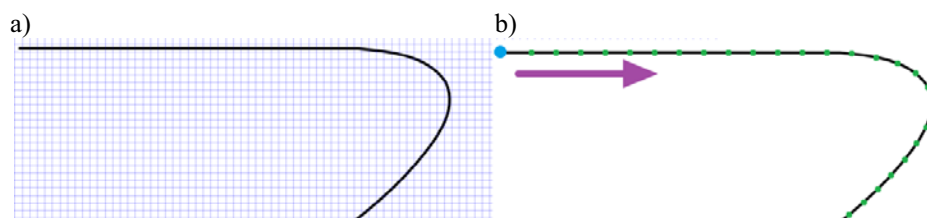


Fig. 2. Extracting path data from SVG file: rasterization (a) and sampling (b) [14]

Using the conversion to HPGL could give better results because the sampling and optimization of the path is done by the application that converts SVG to HPGL. The elaborated methods are in the experimental stage. The both ideas are shown in Figure 2.

4.3 Definition of the path using mixed reality

The method described in this section uses the photo image of the part to help the operator to define the path of robot's effector during the operation carried out on the plane, e.g. applying the glue or sealants, deburring, drilling. The photo is taken by the camera mounted above the plane on which the robot operates. Acquired image is initially processed in order to be clear and properly readable. Then it is imported to the CAD program, where is scaled and orientated according to the robot's coordinate system [15]. In order to define the path, the operator inserts the drawing entities to the CAD program, using the upper layer, above the photo of the part. The Figure 3 shows the example of the path defined with the help of mixed reality[15]. Further details connected with obtaining of the coordinates for the robot are described in [15]. The presented method has a little in common with the one presented in [2], but it is using the photo instead of 3D scan and operates in 2D space. The main advantage of the presented method is that it uses only the free and open source software.

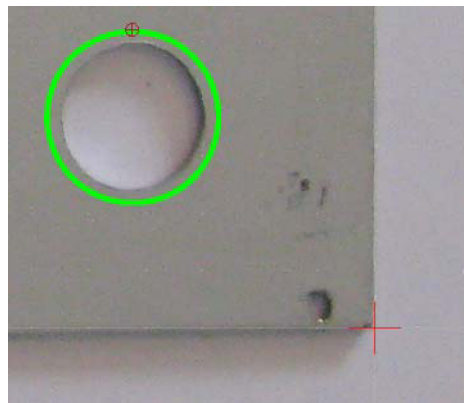


Fig. 3. Example path of effector. In the right-bottom corner of plate the calibration cross is visible

4.4 Current research activities

Each of three methods, developed in the Institute of Engineering Processes Automation and Integrated Manufacturing Systems have the common aim, which is to help the user to design a complicated path that will be recreated by the robot's effector. The previous researches have led to the development of method that uses the mixed reality along with the CAD system in order to generate the data for the robot's controller. The use of free and open source software has significantly reduced the costs and has made the method readily available for the target user. This is the significant factor, which encourages to do the further study of this topic using such sort of the software.

The important topic of the current research is the planning of trajectory in the 3D virtual environment. This issue is only partially solved because the current state of knowledge allows to design the 3D trajectory in the CAD application and export it to the DXF file, so the coordinates could be interpreted by translator. On the other hand, this task could be simplified, when the 3D model of the object could be used in order to adjust the path along the object's edges or surfaces. So far, the 3D model of the object should be drawn in CAD

application or scanned using 3D scanner. Taking into account that the 3D scanner is not always available, it is advisable to search for other effective and low-cost solutions. It is considered to use the mixed reality in the process of three-dimensional trajectory planning. The aim of the current research is also the evaluation of the usefulness of different methods of converting the series of object's photos to the 3D model.

As the third path of research, the problems described in the next section and connected with the robot-based machining will be taken into consideration. The measurements that will be done during machining may lead to valuable conclusions, which may be useful during the process of optimization of the generated trajectory.

5 Open problems

Because the CAD – Robot Interfaces are often used in experimental stages, there are still some open problems.

One of the problem is the stability of movement during the following of the path by the effector. This problem is connected with the number of points that are generated by the application when the arc, circle or curve should be mapped. The points are spread over the path in unequal distances – less densely on the straight sections and more densely on curves. This leads to the changes of effector's speed, which may be critical parameter during applying sealants, glues or operation of cutting. Moreover, it manifests in the changes of tool's load during milling or deburring. If the points are located too densely, the effector may vibrate if it should achieve the points accurately, because there is no enough time to accelerate and decelerate the whole kinematic chain.

The second issue is connected with the previously described one and it is the result of « zero-tolerance », CAD-generated path that must be mapped by the tool, while the different robots of the same type may have different accuracy and also the parts may differ one from another because of manufacturing tolerances. In some cases this may lead to the overloading of the tool or even to its destruction. These types of problems could be solved by using the force-sensing technique described in [16], but it requires the special system that will introduce the correction during the realization of the robot's program.

6 Conclusions

The CAD-Robot Interfaces are still the subject of interest of many scientists. As a justification for conducting such work, a large popularity of the CAD systems is indicated. On the other hand, the lack of standardized programming language for robots makes hard to elaborate the methods that will have the real application in the industry. Without developing the standards, similar to the one used in e.g. PLC programming, there will be no possibility to engage a wider range of potential recipients in the research process.

The solutions presented in the paper are still in development phase, but the scientists had shown the potential possibilities of reducing the gap between the process of designing and the production. There are still open problems that come from the differences between the accurate world of the CAD documentation and the real limitations, like construction differences between robots or tolerances in manufacturing. Beside taking into account these issues, further research should focus on the decreasing of the operator's participation in the process of aquisition of the coordinates from the CAD documentation.

References

1. P. Neto and N. Mendes, *Rob. Auton. Syst.* **61**, 896 (2013)

2. J. Y. Kim, *J. Robot. Syst.* **21**, 625 (2004).
3. P. Neto, N. Mendes, R. Araújo, J. Norberto Pires, and A. Paulo Moreira, *Ind. Robot An Int. J.* **39**, 294 (2012)
4. P. Neto, J. N. Pires, and A. P. Moreira, in *2010 IEEE Conf. Robot. Autom. Mechatronics* (IEEE, 2010), pp. 516
5. J. N. Pires, T. Godinho, and P. Ferreira, *Ind. Robot An Int. J.* **31**, 71 (2004)
6. G. Gołda and A. Kampa, *Przegląd Mech.* **3**, 35 (2015).
7. D. Reclik and G. G. Kost, *J. Autom. Mob. Robot. Intell. Syst.* **2**, 38 (2008).
8. K. Foit, G. G. Kost, and D. Reclik, *J. Achiev. Mater. Manuf. Eng.* **47**, 57 (2011).
9. J. Świder, K. Foit, G. Wszolek, and D. Mastrowski, *J. Achiev. Mater. Manuf. Eng.* **25**, 7 (2007).
10. K.-W. Jeon, J. Ki, Y.-M. Kwon, and H. Ko, in *Front. Converg. Biosci. Inf. Technol. 2007. FBIT 2007* (IEEE, 2007), pp. 622
11. S. Jo, J. Ki, K.-W. Jeon, Y.-M. Kwon, and S. C. Ahn, in *RO-MAN 2007-The 16th IEEE Int. Symp. Robot Hum. Interact. Commun.* (IEEE, 2007), pp. 493
12. C. Reinicke, M. Buss, in *The Internet Challenge: Technology and Applications*, (Springer Netherlands, 2002), pp. 55
13. K. Foit, *Adv. Mater. Res.* **837**, 577 (2014)
14. K. Foit, *Sel. Eng. Probl.* **5**, 27 (2014)
15. K. Foit, *Adv. Mater. Res.* **1036**, 737 (2014).
16. K. M. Murphy, R. J. Norcross, and F. M. Proctor, in *Proc. Second Int. Symp. Robot. Manuf. Res. Educ. Appl. Albuquerque, NM* (Citeseer, 1988).