Robotics and Computer-Integrated Manufacturing 27 (2011) 942–948

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Transformative CAD based industrial robot program generation

Heping Chen [a,*], Weihua Sheng [b]

[a] Ingram School of Engineering, Texas State University, San Marcos, TX 78666, USA
[b] Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA

ARTICLE INFO

Article history:
Received 27 September 2010
Received in revised form
9 February 2011
Accepted 14 March 2011

Keywords:
Industrial robot
Surface manufacturing
Robot programming
CAD model

ABSTRACT

Industrial robots are widely used in various processes of surface manufacturing, such as spray painting, spray forming, rapid tooling, spray coating, and polishing. Robot programming for these applications is still time consuming and costly. Typical teaching methods are not cost effective and efficient. There are many off-line programming methods developed to reduce the robot programming effort. However, these methods suffer many practical issues, such as cable/hose tangling, robot configuration, collision, and reachability. To solve these problems, this paper discusses a new method to generate robot programs. Since industrial robots have been used in production for decades, there are many robot programs for different parts generated by the robot programmers. These robot programs, which contain not only the robot paths, but also the programmers' knowledge and process parameters, can be transformed to generate new robot programs for similar parts. In this paper, a transformative robot program generation method is developed based on the existing ones in the database. Experiments were performed to validate the developed methodology. The results are very promising in reducing the programming efforts in surface manufacturing.

1. Introduction

Surface manufacturing is a process of adding material to or removing material from the surfaces of a part. Spray painting, spray forming, rapid tooling, spray coating, and polishing are some of the typical examples in surface manufacturing. Industrial robots are typically used for these applications. Since there are many requirements for these complex industrial processes, robot programming to satisfy these requirements is very challenging. One example is the robot program generation for the painting processes. The product quality, paint usage as well as the robot performance has to be considered. The quality of the robot programs generated by the teaching methods depends on the programmers' experience and skills. The methods also require that programmers carry out extensive tests on work cells to improve the product quality and system performance. Therefore, they are not cost effective anymore because products are subject to a shorter product life, frequent design changes, small lot sizes, and small in-process inventory restrictions. Furthermore, the programmers have to be exposed to the hazardous environments. To overcome these problems, many off-line programming methods have been developed to make the complex robot programming easier [1–10]. Even though these methods can be used to generate robot paths based on the CAD models of parts, there are many practical issues when the generated robot program is downloaded into a robot controller to control the motion of a robot. Fig. 1 shows a paint robot with hoses attached to the tool for a painting process.

Since the rotation joint can rotate many cycles with the same orientation reading from its encoder/resolver, the attached hoses may be tangled onto the robot arm. This problem is typically solved by the robot programmers manually. When they run a robot program and observe that the hoses and/or cables may cause problems, they will adjust the robot path manually by rotating the joint 360° or changing the robot configuration. The other major problem is the robot configuration. When the robot moves from one position to another, the configuration may need to be changed. However, it is difficult to define the robot configuration using off-line programming. Collision and reachability problems are also typical for industrial robots in off-line programming. Therefore, the robot path generated off-line still needs many adjustments in order to obtain a usable robot path. Furthermore, the process parameters may need to be adjusted accordingly. These adjustments will take a lot of time and effort of the robot programmers. After communicating with several experienced paint robot programmers in ABB Inc., we found that the majority of the robot programming effort is to make adjustments to deal with the above issues and tune process parameters.

Since industrial robots have been used in many manufacturing processes for decades, the robot programmers have been accumulating a lot of experience and data, including the robot programs and AD

* Corresponding author.
E-mail address: hc15@txstate.edu (H. Chen).

0736-5845/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2011.03.006

models. The generated robot programs incorporate the programmers' experience and process knowledge. These data could be transformed to generate new robot programs for new parts. In industry, most of the processed parts have similar geometry, but with some modifications, especially in automotive industry. One example is spray painting of car bodies. Even though there are some differences from one car model to another, the car shapes are quite similar from generations to generations. Therefore, the robot programs can be transformed with some adjustments according to the geometry of new parts. Since the hose/cable tangling, robot configuration, collision, and reachability issues have been solved and the process knowledge is included, reusing these robot programs could save a lot of programming effort and reduce the production cost. However, since there are many robot types, robot programs, and CAD models in the database, there are several problems that have to be solved in order to generate a robot program for a new part. The first one is to find a best matching part in the database with the new part. The second one is to overlap the existing robot path onto the new part. The last one is to make adjustments of the existing path to generate a new path.

There are many tessellated model matching methods developed, especially in image processing. Osada et al. [11] developed a shape distribution method to compare the similarity of different parts. They processed over one thousand parts and categorized them according to their shape distribution based on distance histogram. Ohbuchi et al. [12] improved the shape distribution method using a 2D histogram method, which used the shape orientation distribution. Since the orientation information of a part is important for robot program generation in surface manufacturing, the 2D histogram method is adopted in this paper to find the best matching part. After the best matching part of a new part is found, the new part is transformed to the matching part coordinate system and the existing robot path is overlapped onto the new part. The robot path adjustments method is also developed to generate a new robot program. The programmers' knowledge and parameters related to the process are automatically transferred into the new program. The experimental results show the potential of the developed method to be used for new robot program generation to reduce the robot programming effort and cost. Section 2 describes the developed transformative robot program generation method. Section 3 discusses the experimental results to validate the developed method. Section 4 presents the conclusions.

Fig. 1. Paint robot with hoses. Hoses may be tangled on the robot arm.

2. Robot program generation algorithm

The robot programmers have been accumulating a lot of data in programming robots over decades. These data can be transformed to generate new robot programs to reduce the robot programming effort. When programming a robot to process a part, the robot model and the CAD model of the part are typically given. Therefore, to generate a robot program using the database containing the robot types, CAD models, and robot programs, there are several steps:

- To find a matching robot in the database;
- To find a matching CAD model in the database;
- To overlap the existing robot path in the database onto the new part and generate a new robot path;
- To adjust the robot path according to the geometry of the new part;
- To generate a new robot program by copying all existing process parameters.

The process is summarized in Fig. 2. When the new data (robot type and the CAD model of a part) is input into a computing system containing the database, the robot type is compared with those in the database first. Once the robot types matching, the CAD model of the new part is processed and compared with the existing ones in the database. The process continues until all CAD models are processed. The highest matching score is then found

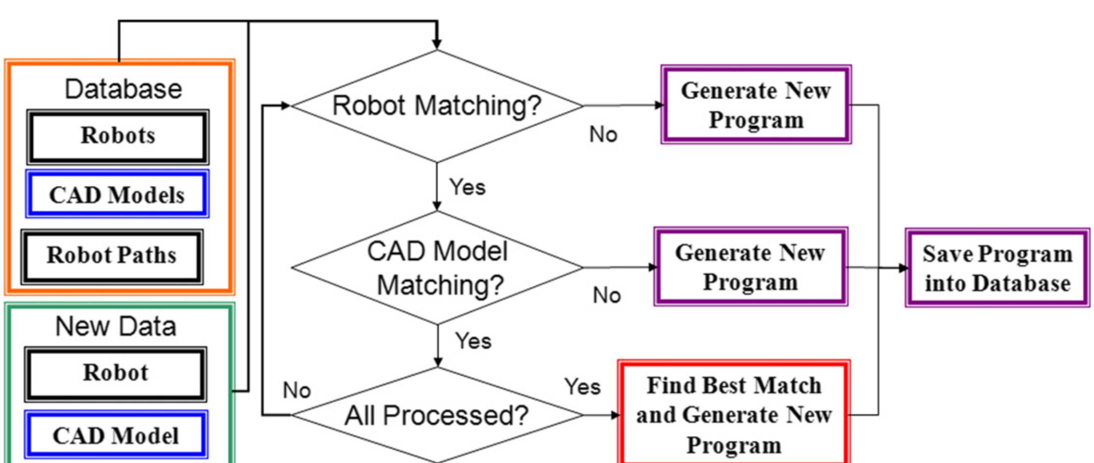

Fig. 2. Robot program generation process based on the database containing the robot types, CAD models, and robot programs.