Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 55 (2016) 6 – 11

www.elsevier.com/locate/procedia

5th CIRP Global Web Conference Research and Innovation for Future Production

Time optimal path planning for industrial robots using STL data files

E. Abele[a], F. Haehn[a]*, M. Pischan[a], F. Herr[a]

[a]*Institute of Production Management, Technology and Machine Tools (PTW) - Technische Universität Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, Germany*

* Corresponding author. Tel.: +49 6151 16 20133; fax: +49 6151 16 20087. *E-mail address:* haehn@ptw.tu-darmstadt.de

---

Abstract

This paper presents an approach based on the A*-Algorithm to generate time optimal tool paths for a robot based deburring process of internal contours. Besides the optimization of the deburring process by using special deburring tools with proper parameters the robots trajectory between the individual points has to be optimized to minimize the deburring process. The simulation environment OpenRAVE with a Matlab interface is used to find the shortest path by solving the traveling salesman problem of a graph automatically generated using the workpiece in the Surface Tessellation Language (STL) data file.

Keywords: *Deburring, Robotics, Path Planning*

---

1. Introduction

Hydraulic control systems are used to control moveable parts like the front strut or the landing flaps in aircrafts. These parts are usually made of aluminum or titanium. Due to an increasing amount of functions, these valves show an increasing number of cross holes. The production process causes burrs at the intersection of the holes. Until today, these burrs cannot be removed reliably by an automated process. Remaining burrs can influence dimensional tolerances and reduce the efficiency and technical lifetime of the component. In some applications, cross holes are used for the lubricant and coolant supply. In this case, burrs could lead to cloggings of critical passages and cause turbulences in the fluid. This may lead to leakages or bursting of the valve. Hence, an uncontrolled removal of the burr during operation must be avoided. The consequence of these basic conditions is a time consuming manual deburring process. An automated deburring process of cross holes for non-safety related parts with industrial robots is usually performed with flexible abrasive brushes. Alternatively processes like Abrasive Flow Machining (AFM), Electro Chemical Machining (ECM) or Thermal Energy Machining (TEM) are used. These processes are very efficient but require specialized equipment and cleaning processes for the used chemicals and the remaining abrasive paste, so they are not suitable for the deburring of safety relevant parts. Due to the high flexibility of industrials robots, the robot based deburring of cross holes is investigated. The path accuracy of industrial robots is low in comparison to machine tools. Hence a special tool which was presented in [1] is used for the process. It compensates the path deviation of the robot and was selected among several other special tools for the deburring of cross holes in [2].

The focus of this paper is the time optimal path planning for industrial robots based on a STL data file. Other researchers aimed at choosing the optimal configuration of axes [3] or using neural networks and genetic algorithm [4] for path planning. In addition path planning has been optimized for machining operations and their characteristics. [5] and [6] adapted the path to compensate force induced deflection and [7] developed a method for automated polishing.

In the first sections of the paper the deburring process and the simulation environment are described followed, by methods to generate and optimize a graph which represents possible tool paths based on the workpiece and the location of the bore holes. Finally the optimal path is calculated and the process validated.

doi:10.1016/j.procir.2016.08.038

2. Deburring Process

A KUKA KR5 sixx R850 robot with an asynchronous spindle from Jäger is used for the process. The so called Orbitool is clamped in the spindle. It is a tungsten carbide cutter which has been developed for the deburring of cross holes. Due to the defined cutting conditions, a better control of the required dimension at the intersection compared to brushes and other deburring methods is possible. Furthermore, the tool can be used on machine tools and industrial robots and is applicable for a great variety of bore diameters. The tool mainly consists of a ball shaped carbide milling cutter with a protective disk, which is made of polished steel and a shaft of tool steel. The deburring procedure using the Orbitool is depicted in Figure 1. To remove the burr, the tool is moved along the bore axis into the smallest of the intersecting holes (see Figure 1 A) until the tip of the tool is close to the intersection (see Figure 1 B). Then the tool is moved in radial direction to the bore surface until the tool axis corresponds to the interpolation diameter. This causes a deflection of the tool. In this situation, only the protective disk is in contact with the bore surface. While the tool rotates, it is moved towards the intersection in a helical motion (see Figure 1 C). When the tool tip has reached the intersection the cutting edges get in contact with the intersection and the deburring process begins. After the tool has passed the whole intersection, the tool stops its rotation and is moved to the bore hole center (see Figure 1 D) and is finally moved out of the workpiece (see Figure 1 E).

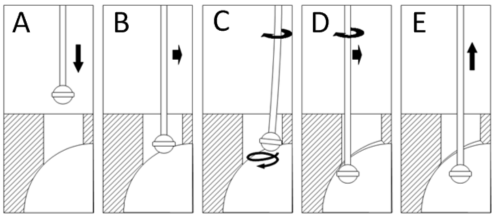

Figure 1. Sequence for the deburring of cross holes using the Orbitool

3. Path Optimization approach

Time saving can be achieved by optimization of the robot path between the bore holes. In the following section, the approach for the path optimization will be presented. The approach was implemented using the robot simulation environment OpenRAVE which was presented in [8]. As one can see in Figure 2 OpenRAVE has an interface to MATLAB. So MATLAB was used to generate and optimize the robot paths.

3.1 Simulation Environment

OpenRAVE is an open-source software for the simulation, visualization, path planning and control of robot applications. Figure 2 shows the four core elements of the OpenRAVE environment.

These elements consist of the core layer, plugin layer, scripting layer and the database layer. The core layer provides the basic functions as an Application Programming Interface (API). The functions can be expanded via the plugin layer. The scripting layer has the function to receive commands and to process them in a simulation using a script language. For reuse data is saved in the database layer. All layers ensure that also non-experts can analyze and solve their problems without a deeper understanding of software systems, path planning and physics.

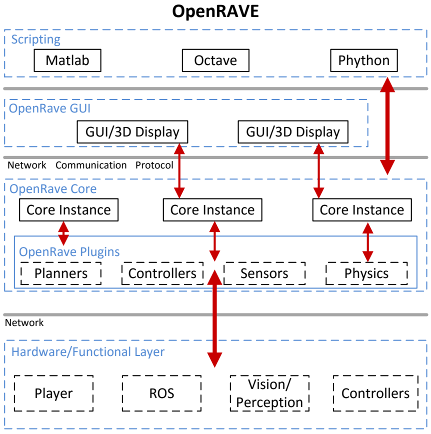

Figure 2. OpenRAVE software architecture [9]

As a first step, a simulation environment was implemented for the used KUKA robot. Since the movement speed and acceleration depend on the maximum torque, the masses, the inertias of the parts and the payload at the tool center point are provided for the dynamic model of the robot:

$$M(q)\cdot\ddot{q}+C(q,\dot{q})+G(q)=\tau+S(F_{xyz,tool},q) \tag{1}$$

where $q \in \mathbb{R}^n, \dot{q} \in \mathbb{R}^n, \ \ddot{q} \in \mathbb{R}^n$q are the position, velocity and acceleration of the joints. $C(q,\dot{q}) \in \mathbb{R}^n$ are the centrifugal and coriolis forces. $G(q) \in \mathbb{R}^n$ is the gravitational force. $M(q) \in \mathbb{R}^n$is the mass inertia matrix. The masses are provided by the robot manufacturer. $\tau \in \mathbb{R}^n$ are the acting torques in the joints. Finally $S(F,q) \in \mathbb{R}^n$ represents external torques and forces at the tool center point (TCP). The test bench for the practical tests and the simulation environment used for the investigations is depicted in Figure 3.

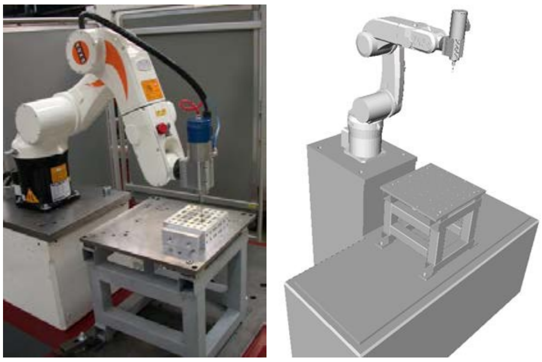

Figure 3. Test bench with the KUKA KR5 sixx R850 (left) and the simulation environment (right)