Proceedings of the 2017 IEEE
International Conference on Robotics and Biomimetics
December 5-8, 2017, Macau SAR, China

Path Generation and Tuning with Touchscreen Device for Industrial Robot

Xinyu Fang, Cheng Li, Hao Gu
ABB Corporate Research Center
Shanghai, China
Email: {xinyu.fang, cheng.li, hao.gu}@cn.abb.com

***Abstract*—Traditional path generation and tuning process for industrial robot is tedious, time consuming and experience dependent. This paper presents a novel approach to generate and tune robot path using a touchscreen device. With technique of curvature guided data filtering and percentage guided data aligning, a 2D robot path can be easily created and compensated, which can highly improve the productivity for robot engineers. This paper gives a concrete demonstration to show this idea.**

Keywords—Path generation; path tuning; touchscreen device; industrial robot

I. Introduction

Industrial robot path generation is tedious, repeated, time consuming and experience dependent in robot applications. In general, there are two methods to create robot path: offline programming and online programming. Offline programming is based on virtual or ideal world while online programming is in the real world. A comprehensive review of recent research progresses on the programming methods for industrial robots is provided in [1].

In general, the advantage of online programming is 'what you see is what you get' and the program can be used directly. However, a robot path (especially an irregular path) is not easy to be programed in online mode. Particularly in some applications such as dispensing and laser cutting, the path pattern is randomly drawn by the process engineer with experience, of which the numerical description is difficult to get. What's worse, the path must be tested and changed frequently, which greatly increases the workload of robot programming.

With robot offline programming, the robot path can be generated easily by computer algorithm. But the programed path cannot be directly applied to robot because of two reasons. Firstly, the ideal tool and work piece are different from the real ones. To make up the gap, calibration of tool and work piece is necessary. The second reason is that the practical robot path at motion has deviations from the ideal one. So path tuning is usually needed in real processes.

To avoid the disadvantage and take full use of the advantage of online and offline programming is the key in industrial robot field. In other words, easy robot programming and high accuracy robot path is the critical request in current robot applications.

To improve the efficiency of robot path generation, different solutions have been proposed in recent decade. Reference [2] presents an introduction to the challenges of robot path programming, surveys and analyzes four approaches for programming robot path. Paper [3] gives several new concepts, such as parametric driven modeling, path based on application and scenario template, to simplify robot offline programming and simulation. A pattern based robot offline programming method is demonstrated in [4]. Reference [5][6] describes a method and system to generate robot path automatically with lead-through and path learning based on force control. A new vision based programming methodology is presented in [7] which combines information of the real and ideal world, especially adapted for robot grinding and deburring operations. Reference [8] introduces a path generation method in robot arc welding application with a pseudo stereovision system and a personal computer. Reference [9] presents a complex support system to do robot trajectory planning and programming online.

As the touchscreen device is more popular and powerful in recent years, some features from this device are applicable to robot programming. For example, it can record the touchpoint position, even orientation and force. At the same time, it can communicate with robot easily by wireless or cable. So the touchscreen device has been a suitable equipment to make robot programming easier. It can be used as device for tool & work piece calibration [10] and path tuning. Additionally this type of device is easy to use, even to novice operator, which will lower the level of expertise needed for robot operator.

In the following sections, a new approach with touchscreen apparatus for easy robot programming will be described. This approach is a close loop method of robot programming between offline and online. Section 2 will describe the developed technology. The third section will give a demonstration as a real example and the conclusions are drawn in Section 4.

II. Methodology

To match the mentioned request in section I, a method is proposed to automatically program a 2D path and quickly tune the practical path. The proposed system is illustrated as in Figure 1. Upon the robot system, the hardware of touch probe and touchscreen device, and several software functionalities are involved.

978-1-5386-3742-5/17/$31.00

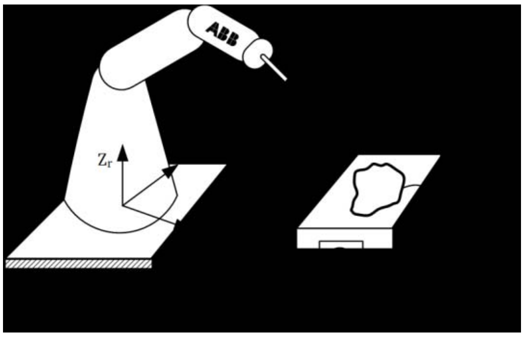


Fig. 1. The concept of robot path generation and tuning

The model of the part to be processed can be easily imported into the touchscreen device. As illustrated in Figure 2(a), the software functionality of touchscreen device enables the process engineer to draw a 2D path as wish on a 1:1 scaled part model. The touchscreen device can convert this drawn path into numerical description by recording all the drawing points in the touch panel frame and pre-process them. Since the work object of the commissioning apparatus can be identified [10], the numerical path can be described with respect to the robot frame. Therefore, robot targets along the path can be generated automatically.

With the desired path generated, robot can automatically go through the path with a touch probe. After a physical path running on the touchscreen device, deviation between the actual and desired path can be measured and identified with sub-millimeter accuracy (depending on the resolution of the touch screen, typically it could be smaller than 0.1mm), as illustrated in Figure 2(b). By feeding back the path error to the programming environment, a close-looped path tuning can be conducted automatically by adjusting the parameters such as robot targets, speed, zone, and tool orientation.

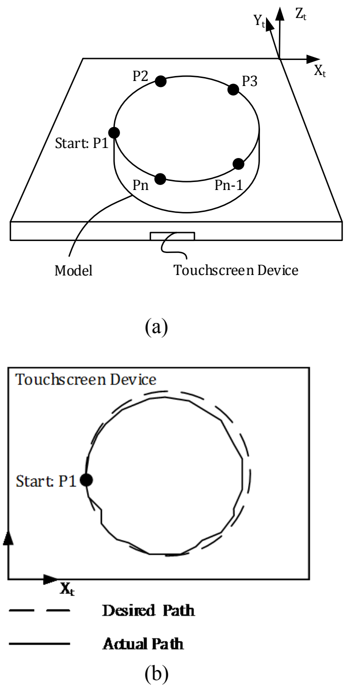


Fig. 2. Path data generation and capture with touchscreen device

The above methodology can be divided into several technical packages, which are described in detail as following.

A. Stroke Data Capturing and Separation

The touch probe can be capacitive pen or electromagnetic pen. It will be induced with the touchscreen device in an effective range.

Once the touch probe is entering the effective range, the event for touch pen captured is invoked by the touchscreen device. In the event, the data of pen can be read. It includes pen's position or orientation relative to the screen. As for some advanced pen, it can include more information, such as touched pressure in normal and tangent direction, and so on. In each invoked event, the pen's data will be stored into a data structure.

In general, one drawing consists of several strokes. The strokes' sequence will define the drawing sequence. Figure 3 give s simple example to explain the data structure. In this drawing, there are 2 strokes. The first stroke is A, the second one is B. The stroke A's data will be stored into data list A while the stroke B's data will be stored into data list B.

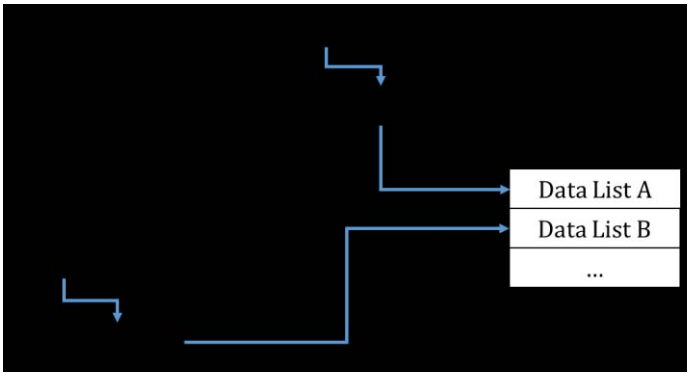


Fig. 3. Data structure of stroke data capturing

The problem of how to distinguish different strokes rises naturally. The following method according to the type of touch pen is proposed.

If the pen is an electromagnetic pen, its normal pressure will indicate whether the pen touches the screen. One variable will record the pen's normal pressure of last time. If last normal pressure is zero and current normal pressure is not zero, it means that a new stroke is created. If last normal pressure is not zero and current normal pressure is zero, it implies that this stroke is completed.

If the pen is a capacitive pen, the touched point will be valid when the pen is really touched with the touchscreen device. One variable will store the data of touched point of last time. If last touched point is invalid and current touched point is valid, it means that this is a new stroke. If last touched point is valid and current touched point is invalid, it implies that the last touched point is the ending point of the current stroke.

B. Stroke Data Filtering

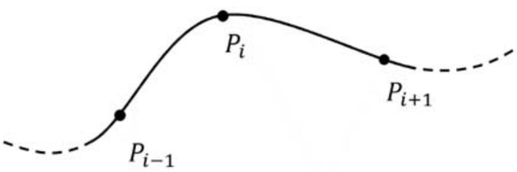
During the process of capturing a stroke's data, the amount of sampled data is huge. With the help of curvature guided data filtering, all the stroke data is filtered to a reasonable amount which still can properly fit the original stroke. Furthermore, based on such a way to process the data, the robot can provide only linear movements, which will simplify the robot programming.

The core problem here is whether each individual point P_i of all the position data $P_0, \ldots, P_N$ should be kept or discarded. In order to clearly state this method, the following two concepts are defined:

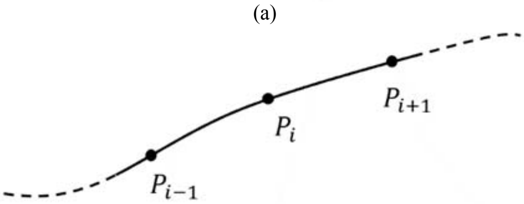
- $\mathrm{D_{line}}(P_i, P_{i+1}) = \|P_{i+1} - P_i\|$, the Cartesian distance between any adjoining two points;
- $\mathrm{D_{arc}}(P_i, P_j) = \mathrm{D_{line}}(P_i, P_{i+1}) + \cdots + \mathrm{D_{line}}(P_{j-1}, P_j)$, the arc length between any two points along the path, which is the sum of all the Cartesian distances of adjoining two points in between.

For a point P_i, the following is its regulations:

- If it is the first point P_0 or the last point P_N, the point should be retained.
- If $\mathrm{D_{line}}(P_{i-1}, P_{i+1}) < c \cdot \mathrm{D_{arc}}(P_{i-1}, P_{i+1})$, which suggests the local curvature is large, then the point P_i will be retained, as illustrated in Figure 4 (a); otherwise, the point P_i should be removed since the straight line from P_{i-1} to P_{i+1} is a good representation of the original path in the small local curvature region, as shown in Figure 4 (b). Here parameter c represents the local curvature and its range is from 0 to 1. The closer this parameter is to 1, the better chance that point P_i will be retained;
- If $\mathrm{D_{line}}(P_{i-1}, P_i)$ is smaller than the robot target resolution, the point P_i will be removed.



(a)



(b)

Fig. 4. Stroke data filtering (a) P_i in a curved segment should be retained; (b) P_i in a relatively straight segment could be removed.

C. Robot Path Generation

After the sampled stroke's data is refined, they will be converted into robot targets. And the stroke will be mapping into a path section.

To connect two path sections, two additional targets are created in each path section. One is for approach to the path section, the other is for departing from the path section.

Fig. 5. Additional targets creation in one path section

Figure 5 illustrates the method to create two additional targets in one path section. The start target is the first target on the screen and the end target is the last target on the screen in a path section. The approach target is created above 5 millimeters of start target and the departing target is above 5 millimeters of end target.

Because all strokes' data are relative to the corner of the touchscreen, the generated robot targets will be based on the corner of screen too. This corner will be defined as work-object. After the calibration of the work-object with the same system (reference [10] presents detailed methodology, experiments and error analysis), the robot path can be executed by the real robot.

D. Robot Path Tuning

To tune the robot path, the first step is how to align robot path data from two paths.

Figure 6 illustrates the method to align the robot path data. Reference path is the desired path that the robot should follow, which consists of points $P_0^R, \ldots, P_N^R$. After the reference path is imported into the robot controller and executed, the touchscreen device would tracks the robot motion and obtain the actual path, which consists of actual points $P_0^A, \ldots, P_M^A$. Please note that the number of points in the reference and actual path may not be the same. Therefore, the problem here is to find a method to synchronize points $P_0^I, \ldots, P_M^I$ on actual path with points $P_0^R, \ldots, P_N^R$ on reference path.

Along the actual path, points P_i^I corresponding to the i^{th} reference points P_i^R will be interpolated in the following way.

- The first point P_0^A and the last point P_M^A of the actual path are the first and the last point of our interpolation point P_0^I and P_N^I, respectively;
- For all the points in the middle, interpolated point P_i^I will be calculated from (1)

$$\frac{\mathrm{D_{arc}}(P_0^A, P_i^I)}{\mathrm{D_{arc}}(P_0^A, P_M^A)} = \frac{\mathrm{D_{arc}}(P_0^R, P_i^R)}{\mathrm{D_{arc}}(P_0^R, P_N^R)} \quad (1)$$

In short, the ratio of the path length before point P_i^R relative to the whole reference path would be the same as that before point P_i^I relative to the whole actual path. So this method is called as the percentage guided data aligning.

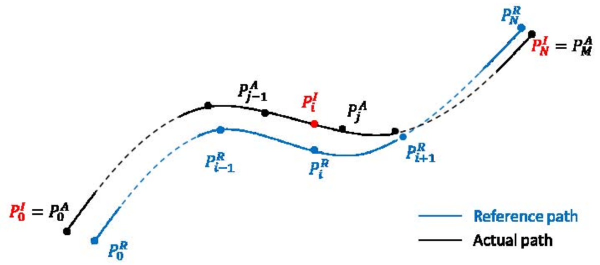


Fig. 6. Robot Path Data Alignment

By the above rules all the interpolated points can be found corresponding to the reference points.

Offset is the difference between the interpolation point P_i^I and the corresponding reference point P_i^R in two dimension, which will be compensated to P_i^I. Equation (2) and (3) explain the compensation formula.

$$Offset = P_i^I - P_i^R \quad (2)$$

$$New\ P_i^I = P_i^R - Offset \quad (3)$$

This measurement-compensation process may be iterated until the average difference between the latest actual path and reference path is small enough. Average difference between the two paths is define as following:

$$\text{Average difference} = \frac{1}{N+1}\sum_{i=0}^{N}\left\|P_i^I - P_i^R\right\|$$

III. Implementation

To evaluate the proposed method, a demonstration is created.

In this demonstration, the main hardware components include 1 ABB robot (IRB120), 1 touchscreen device (DTH-W1300 from WACOM), 2 touch probes (one is for user's drawing and the other is installed on robot for drawing) with 2048 pressure levels, 1 slider and 1 gripper. The touch probe is type of electromagnetic pen. It can be induced with the touchscreen device. The effective range, which is the distance between pen's tip and tablet's screen, is about 10 millimeters.

The main software consists of two parts: one is running on DTH-W1300 to record user drawings, generate path from the drawings and tunes the generated path; the other is running on IRB120 controller to accept and execute the generated path.

The DTH-W1300 communicates with IRB120 controller through WIFI or Ethernet cable.

Fig. 7. Demonstration of path generation from graphics tablet

Figure 7 illustrates the demonstration cell. IRB120 is in a robot cell to isolate from user. A slider is designed to carry the DTH-1300 into or out of the robot cell.

The major feature of this demonstration is that IRB120 can repeat what the user draw on the DTH-W1300. The basic work flow for this demonstration is shown in Figure 8.

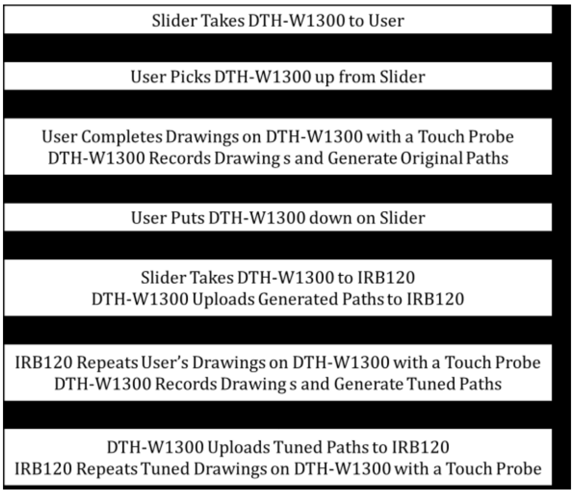


Fig. 8. Work flow of path generation and tuning from touchscreen device

The word 'repeat' defines two rules:

- What IRB120 draws should be the same as what user draws within tolerance of robot path accuracy;
- IRB120's drawing sequence should be the same as the user's drawing sequence.

To complete the major function, following sub functions should be realized:

- Record trajectory data from user's drawing on DTH-W1300;
- Generate path from sampled drawing data on DTH-W1300;
- Transfer generated path from DTH-W1300 to IRB120;
- Display drawing when user and robot draw on DTH-W1300;
- Synchronize state between DTH-W1300 and IRB120.

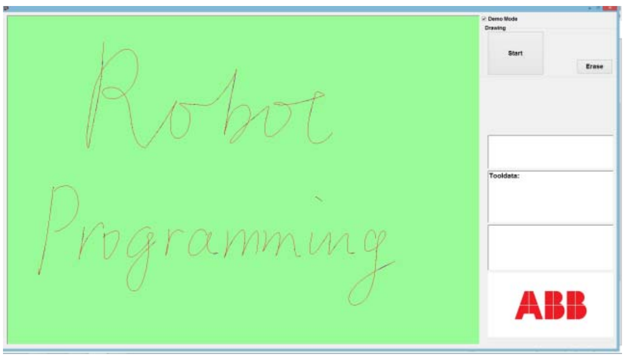


Fig. 9. User interface of path generation and tuning from touchscreen device

According the list of sub functions, one software is developed on DTH-W1300. Its name is 'Path Generation and Tuning from Touchscreen Device (PGTTD)'.Figure 9 presents the user interface of PGTTD.

PGTTD is developed by Visual Studio 2012 with program language C#. Its reference library includes PC SDK and WintabDN. The first library is used to communicate with IRB120 while the second one is used to interact with DTH-W1300. PGTTD is running on DTH-W1300, whose operation system is Windows 8.1.

Based on the recorded or tuned drawing data, a standard RAPID module file will be created on DTH-W1300. Then PGTTD will transfer the created file from DTH-W1300 to IRB120 with PC SDK.

The drawing display area is a picture box in PGTTD. As mentioned in section 'Stroke Data Capturing and Separation', the picture box will refresh and redraw the drawings in each touch probe captured event.

To realize the work flow presented in Figure 8 and to synchronize the state between DTH-W1300 and IRB120, a state machine is designed in PGTTD. This state machine is running in a timer event and the cycle period is 0.5 second. Figure 10 explains the state machine.

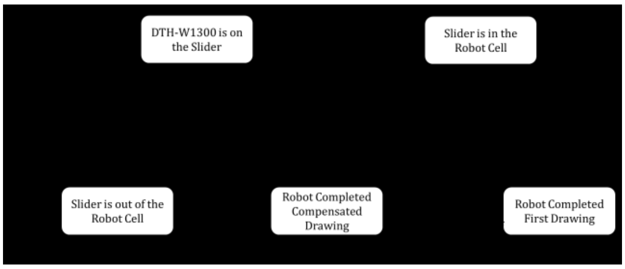


Fig. 10. State machine in PGTTD

In the demonstration, plenty of drawings have been verified. After one round of tuning, the robot path accuracy can be improved about 80%-90%. Figure 11 shows a comparison of path tuning for a real drawing. The red path is the reference path and the black path is the actual path executed by the robot. After path tuning, the actual path is better coincidence with the reference one.

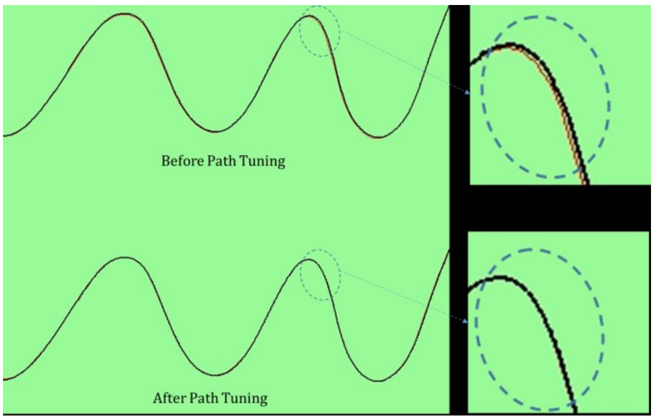


Fig. 11. Path Tuning Comparision

IV. CONCLUSIONS

In this paper, a new approach to generate and tune robot path with touchscreen device is presented. It is a generic method without big computation effort for path processing. Touchscreen device is a perfect equipment to record the 2D trajectory data. Based on the curvature guided algorithm, the sampled data can easily be filtered. It helps to create robot path in high quality. Based on the percentage guided algorithm, two robot paths can easily be mapping. With the mapping, the path deviation is easy to calculate and the path can be compensated easily. Since the touchscreen device is used widely and it is easy to use as well, this approach lows down the threshold of robot programming and tuning as well. Through the implemented demonstration, the approach has proved to have more productivity and higher engineering efficiency.

REFERENCES

[1] Zengxi Pan, Joseph Polden, Nathan Larkin,Stephen Van Duin,John Norrish, "Recent progress on programming methods for industrial robots," Robotics and Computer Integrated Manufacturing, 28.2, pp. 87-94, 2012.

[2] Gregory F. Rossano, Carlos Martinez, Mikael Hedelind, Steve Murphy, Thomas A. Fuhlbrigge, "Easy robot path programming concepts: An industrial perspective on path creation," pp. 1-6, 2013.

[3] Liwei Qi, Xingguo Yin, Haipeng Wang, Li Tao "Virtual engineering: Challenges and solutions for intuitive offline programming for industrial robot," Robotics, Automation and Mechatronics, 2008 IEEE Conference on IEEE, pp. 12-17, 2008.

[4] Xinguo Yin, Li Tao, "Virtual engineering: Model based off-line programming method for industrial robot," Robotics, Automation and Mechatronics, 2008 IEEE Conference on IEEE, pp. 18-23, 2008.

[5] Jianjun Wang, Hui Zhang and George Zhang, "A force control assisted robot path generation system," IEEE International Conference on Automation Science and Engineering IEEE, pp. 528-533, 2008.

[6] George Zhang, JianJun Wang and JingGuo Ge, "Robotic path learning with graphical user interface," International Symposium on Robotics IEEE, pp. 1-4, 2014.

[7] Bjorn Solvang, Gabor Sziebig, and Peter Korondi, "Vision based robot programming," IEEE International Conference on Networking, Sensing and Control IEEE, pp. 949-954, 2008.

[8] Pachidis, Theodore P., and J. N. Lygouras. "Vision-based path generation method for a robot-based arc welding system," Journal of Intelligent & Robotic Systems, 48.3, pp. 307-331, 2007.

[9] C.Kohrt, R.Stamp, A.G.Pipe, J.Kiely, G.Schiedermeier, "An online robot trajectory planning and programming support system for industrial use," Robotics and Computer-Integrated Manufacturing, 29.1, pp. 71-79, 2013.

[10] Hao Gu, Qingwei Li, Jinsong Li, Jihuan Tian, Liwei Qi, Jianjiang Wang, "Method for industrial robot commissioning, industrial robot system and control system using the same," Patent WO 2016/154995 A1, 2016.