

This document is downloaded from DR-NTU, Nanyang Technological University Library, Singapore.

Title	Automatic Robot Taping: System Integration
Author(s)	Lembono, Teguh Santoso; Yuan, Qilong; Zou, Yuhua; Chen, I-Ming
Citation	Lembono, T. S., Yuan, Q., Zou, Y., & Chen, I.-M. (2015). Automatic Robot Taping: System Integration, 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), July 7-11, Busan, Korea.
Date	2015
URL	<a href="http://hdl.handle.net/10220/38759">http://hdl.handle.net/10220/38759</a>
Rights	© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [ <a href="http://dx.doi.org/10.1109/AIM.2015.7222633">http://dx.doi.org/10.1109/AIM.2015.7222633</a> ].

# Automatic Robot Taping: System Integration

Teguh Santoso Lembono, Qilong Yuan, Yuhua Zou, I-Ming Chen, *Fellow, IEEE*

**Abstract**— Industrial robot arm has been used in a lot of applications, but most of them require manual teaching from the operator, or careful path-planning by experienced programmer. Offline programming is one of the solution for more sophisticated applications, but the 3D model of the workpiece should be provided, especially for applications such as painting, thermal spraying, etc. With the increasing availability of 3D scanner and cheap RGB-D camera such as Kinect (with its 3D model generation algorithm, Kinect Fusion), the 3D model of the object can be generated with lesser effort. This paper discusses the integration of scanning, path planning, and control of robot to achieve a complicated task: Automated Robot Taping. Kinect and 3D scanner Artec Eva were used to generate the 3D model, and a taping tool (attached to an industrial robot arm) was designed and manufactured. Based on the 3D model, a path planning algorithm was developed to generate the robot trajectory. To validate the system, some experiments have been conducted, and the automatic taping has been successfully done on the actual object. The application that is considered here is automatic taping, where a workpiece surface is to be covered by a masking tape, to be protected during plasma spraying. The overall framework, however, can be used for other applications such as painting, thermal spraying, etc.

## I. INTRODUCTION

Industrial robot has been used in a lot of applications, ranging from the simple ones such as pick and place to the more sophisticated ones that require more involved programming such as welding[1] and thermal spraying[2]. For simple applications, the operator simply teaches the robot to move to certain positions manually. The positions will then be recorded for subsequent operation. Application such as painting, however, requires more complicated robot trajectory in which manual teaching is too cumbersome, especially for complicated geometry. Offline programming[3][4] is widely used in this case, where the trajectory is generated in a simulation environment (such as RobotStudio in ABB). Another possibility is programming by Augmented Reality, which combines the knowledge of the real world with the CAD data to ease the programming [5].

There have been a lot of works regarding path planning for applications such as painting[6]–[8], thermal spraying[7][8] and welding [10][11]. However, most of these assume that the 3D model of the workpiece is readily available. For manufacturing industries such as automotive (where the use of robots is prevalent), it does not pose any problem, since the manufacturer will definitely possess the 3D model from the

design stage. However, for others such as the overhaul industries, where parts of the engine need to undergo major repair, such 3D model might not be available. The parts that need repair are also changing, depending on the order. Even if the 3D model of the part is available, the damage on the part may cause substantial change to the geometry of the part.

One of the examples is in aeroplane engine overhaul, where damaged engine parts need to be repaired. One of the process involved is plasma-spraying, where parts of the engine is coated at high deposition rate. Before this process, the other parts of the engine that do not require coating should be covered by tape. The taping is normally conducted manually. If robot is to take over such task, it requires the 3D model of the part. However, to construct the 3D model of each part manually might need too much effort.

With the increasing availability of 3D scanner, the 3D model of the workpiece can be generated with lesser effort. No expertise in CAD system is required, and the time spent in building the 3D model is reduced by a lot. Moreover, there are now cheap RGB-D cameras such as Kinect, which is capable of building a 3D model by using the algorithm developed by Microsoft, Kinect Fusion.[12]

In this paper, we introduce a system and the corresponding method for automatic robot taping. A 6-dof industrial robot arm, equipped with the taping tool, is used to automatically tape a workpiece. The workpiece is attached to a rotating platform, consists of a stepper motor and Arduino Mega as the controller. A 3D scanner is used to scan the workpiece and generate the 3D model as point cloud data, with the meshes and the normals of the surface. These data are then used for path planning, which generates the trajectory of the robot. The trajectory is finally sent to the robot controller.

The remaining part of the paper is organized in the following manner. Section II describes the equipments used in the application. Section III introduces the whole automated taping solution. Section IV provides some of the experimental results. Section V discusses the challenges in the implementation. Then, the paper is briefly concluded in Section VI.

## II. EQUIPMENT

In this section, all equipments that are used in the automatic taping system will be described.

### A. Rotating Platform

The platform is operated by a stepper motor (NEMA 23), equipped with a 15.3:1 planetary gear. It is capable of holding 15.5 Nm torque. The motor is driven by Leadshine M542 Stepper Motor Driver, and controlled by Arduino Mega. The platform design is pictured in Figure 1.

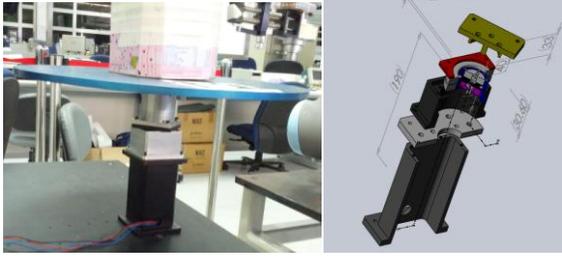


Figure 1: Rotating Platform Design

The rotating platform has two functions: one is for the scanning, where the platform is continuously rotating while the scanner scans the workpiece. The other function is to help the taping process. The taping tool, attached to the robot arm, cannot reach all points on the workpiece surface. By rotating the workpiece during the taping, the whole surface becomes accessible to the taping tool.

### B. Taping Tool Mechanism

The taping tasks require proper contact between the surface of the taping object and the tape. In human manual taping, the finger of the worker presses the tape along the taping orientation in order to guarantee the taping quality. In automatic taping, a special mechanism is needed to replace the function of the human. In this work, we designed a taping end-effector as shown in Figure 2. The rear part holds the tape roller and guides the tape. The front part serves as a compliance ‘finger’ to press the tape to the surface and compensate some of the manipulation errors from the rotation platform and the robot arm. The stiffness of the spring is about 2000N/m. During the taping, the spring mechanism is pressed with 0.5cm allowing about 10N contact forces to press on the tape. This spring mechanism can compensate the error from the scanning results and the robot manipulation.

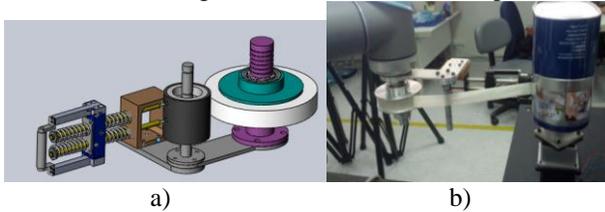


Figure 2: a) Taping Tool Design b) Taping tool attached to the robot arm

### C. Kinect & Artec Eva

Kinect and Artec Eva are used to build the 3D model of the workspace. Both uses structured light to obtain the depth data of its environment. The accuracy of Artec Eva can achieve up to 0.1mm, while Kinect has much poorer accuracy (its random error in depth measurement is around a few millimeters at 0.5m range, and increasing with the distance)[13]. However, Kinect has larger range (up to 8m) compared to Artec Eva (maximum 1m).

### D. UR10

UR10 is a low-cost industrial robot arm, developed by Universal Robots from Denmark. Its weight is only 28.9 kg, and its repeatability is 0.1mm. Its controller can communicate with other computer through TCP-IP protocol.

## III. AUTOMATED TAPING SYSTEM

The automatic taping system consists of four sub-processes: scanning, post-processing, robot path-planning, and execution. The whole program is written in C++, with the help of additional programs such as Kinect Fusion and Meshlab. Each component will be discussed in the following sections.

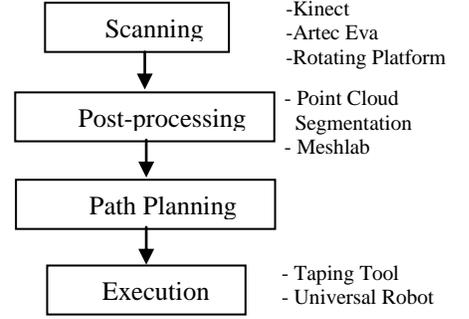


Figure 3: Process Flow for Automated Taping System

### A. System Setup

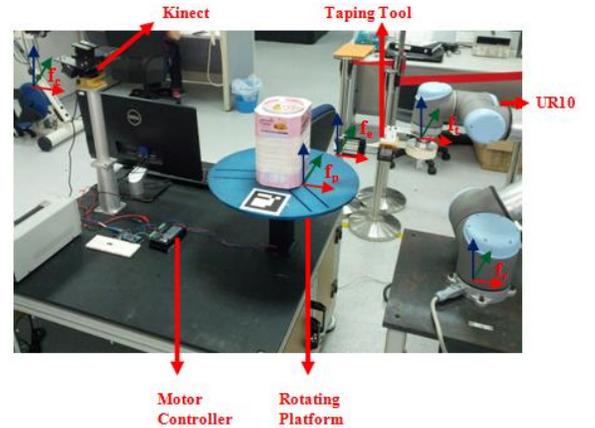


Figure 4: System Setup

First, the object is attached to the rotating platform. Kinect is located around 1m from the object, while the robot is on the opposite side. There are five coordinate frames attached at different location:

- $f_r$  = robot frame, attached at the base of the robot
- $f_c$  = camera frame, attached at Kinect
- $f_p$  = platform frame, attached at the center of the rotating platform
- $f_t$  = TCP frame, attached at the TCP of the robot
- $f_e$  = end effector frame, attached at the taping head

In the setup, there are three parameters that should be determined: the transformation from the camera frame to the platform frame ( $T_{pc}$ ), the transformation from the platform frame to the robot frame ( $T_{rp}$ ), and the transformation from the end of the taping head to the TCP of the robot ( $T_{et}$ ).

$T_{pc}$  is obtained by aligning the robot base and the rotating platform orientation manually, and measuring the translation between the two origins. To obtain  $T_{rp}$ , a checkerboard pattern is placed on the rotating platform, as seen in Figure 4. This checkerboard has a predefined frame (which has the

same orientation as the platform frame), and it can be detected by the camera. Let this transformation from the pattern to the camera frame be called  $T_{cs}$ . Its inverse,  $T_{sc}$ , is the camera frame described in the pattern frame. Since the pattern is only offset from the platform frame, by applying simple translation, we can obtain  $T_{pc}$ . Finally,  $T_{et}$  can be obtained from the CAD model of the taping tool.

### B. Scanning

To obtain the 3D model of the object, the object is attached to the rotating platform for scanning. As the object is being rotated, the scanner captures the object's geometry and generates 3D model in the form of point cloud data (in .OBJ format). The output consists of each point's x-y-z location in the camera frame, including the the mesh and the normal of each point.

Both Kinect and Artec Studio were used to generate the 3D model. The results are compared in Figure 5.

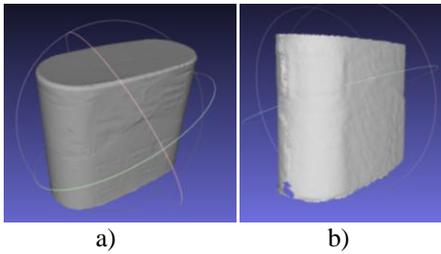


Figure 5: Comparison of Artec Eva and Kinect. a) Artec Eva scanned result  
b) Kinect scanned result

While Artec Studio can generate better result in terms of accuracy, Kinect still generates sufficiently accurate model for the application. The above result was generated using Artec Studio and Kinect Fusion (developed by Microsoft).

Another version of Kinect Fusion has been developed by PCL using PCL libraries, which is open source. This version was also tried, but the result we got is not as accurate as the Microsoft version. The comparison can be seen in Figure 6.

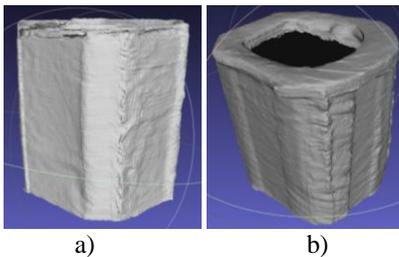


Figure 6: Comparison between Kinect Fusion, a) Windows and b) PCL version.

### C. Post-processing

The 3D model is then edited in Meshlab to be simplified, to reduce the computational effort for further path planning. The vertices which are closed to each other at a certain distance (0.001m is chosen, close to the accuracy of the camera) are merged together. The number of points can be reduced from around 500,000 points to around 70,000 points after simplification.

The resulting 3D model from the scanner also contains the rotating platform, which is not necessary for the path planning. The object model can be separated from the rotating platform model by manual edit in Meshlab. To increase the automation, however, a point cloud segmentation program was developed. In the program, the user selects three points on the rotating platform, and the object model on top of the platform will be automatically separated. The output is again in .OBJ format.

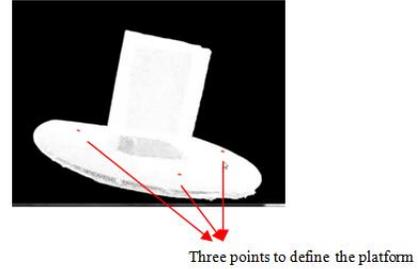


Figure 7: Selection of three points for platform detection in point cloud segmentation

### D. Path Planning

The .OBJ file from scanning process is then fed into another program to do the path planning. The point cloud in the .OBJ file is described in camera frame, so it needs to be transformed to the platform frame (otherwise, when rotation is considered in the path planning, it becomes more difficult).

The path planning can be divided into two parts: global and local path planning.

#### i. Global Path Planning

In global path planning, the desired points along the workpiece surface are identified, as the point-of-contact with the end of the taping tool. The taping starts from bottom of the object, all the way until the top. The algorithm is as follow:

1. Get the initial point for taping from the digital 3D model with initial height ( $h=h_0$ ), and angle ( $\theta=0$ ) with respect to the reference frame of the platform.
2. Numerically calculate the perimeter ( $S$ ) of the closed curve on section  $z=h_0$ . The detailed method is as follows
  - a.  $Int\ i=0; \theta=0$
  - b. When  $\theta < 360$ ,  $\theta = \theta + d\theta; i = i + 1$ ; Search the surface point  $v_i$  in neighbourhood best satisfying  $(h_0, \theta). S = S + |v_{i-1}v_i|$
  - c.  $i = 0; \theta = 0$
3. Determine the "pitch angle" (Equivalent to screw pitch if the part is cylinder) of the tape ( $p$ ) based on the perimeter:

Pitching angle (in rad):  $p = (1 - r)d_t / S$   
where  $r$  is the overlap rate of the tape. Here  $r=20\%$ .

**Then start the path planning.**

The main idea is to update the pitching angle and robot end-effector pose to make sure that the tape is properly taped to cover the surface with the given overlapping parameter.

4.  $\theta = \theta + d\theta; i = i + 1$ .
5. Search the surface point  $v_i$  with normal vector  $v_{n,i}$  in neighbourhood best satisfying  $(h, \theta)$ .  
Then:  $h = h + p/|v_{i-1}v_i|$  (This is the pressing point in the taping path planning).
6. If  $\theta > 360$ , then  $i = 0; \theta = 0$  and update  $S$  and  $p$  in the same way as in 2 and 3.
7. Calculate the robot pose based on  $(v_i, v_{n,i}, p)$ .
8. If  $h < h_{max}$ , go to 4.

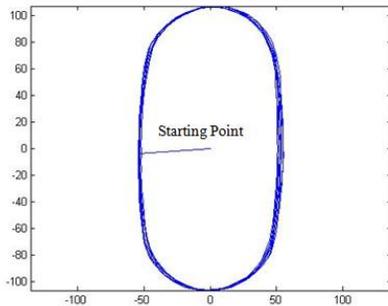
**ii. Local Path Planning**

The global path planning yields the points on the workpiece surface that are to be traversed by the taping tool. However, for most object, the taping tool will not be able to reach all 360° positions on the workpiece surface. Therefore, in local path planning, the idea is to divide the taping process into two phase:

- Phase 1. The taping tool traverses 90° of the workpiece surface in counter-clockwise (CCW) direction
- Phase 2. The platform rotates the object 90° in clockwise (CW) direction.

During phase 2, the taping tool is to maintain a contact with the same point on the workpiece surface. Therefore, the taping tool should also rotate back 90° CW (the center of this rotation is also the center of the rotating platform). This pattern is repeated until the whole surface is covered with tape.

The example of global path planning can be seen in Figure 8a, and the local path planning in Figure 8b. Both are viewed from the top. As explained above, in local path planning, first the taping tool traverses the workpiece surface for 90° CCW (1). And then it rotates 90° CW (2). It traverses another 90° CCW (3), and then rotates back 90° CW (4).



a)

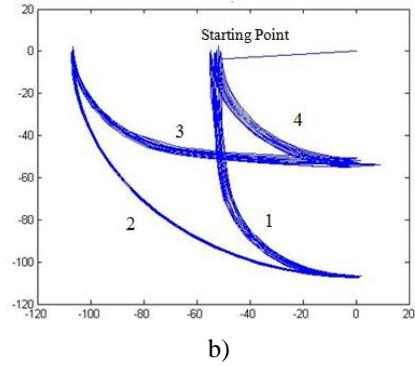


Figure 8: Path Planning result: a) Global b) Local

**E. Execution**

From the result of the path planning, the trajectory of the taping head is obtained. The trajectory is described in the platform frame, and it needs to be transformed to the robot frame.

Moreover, the robot controller controls the TCP of the robot instead of the taping head. Hence a transformation should be done to each point in the trajectory, to transform those points to the TCP of the robot.

As explained in section D, there are two phases in the taping: in phase one the robot moves to tape the object 90°, and in the second phase the robot and the platform move together. To control both the robot and the tape, one computer is used to interface to the robot controller and Arduino Mega (for the rotating platform). First the path planning algorithm is run in the computer, to generate the robot trajectory and the platform angles. Then the computer communicates with the robot controller through TCP-IP, sending each pose (x,y,z point and rotating vector for orientation) at certain interval. The computer communicates with Arduino Mega by serial communication, sending the amount of angles for the motor and the duration of the rotation.

To ensure that the robot motion will be smooth, the motion command should be sent in precise timing (accurate within 50 ms). If the command is late, the motion will be jerky, but if the command is too fast, the synchronization with the rotating platform will be affected. To realize the smooth motion, the algorithm in Figure 9 is applied in the robot execution program (The word inside the bracket “( )” is a variable name).

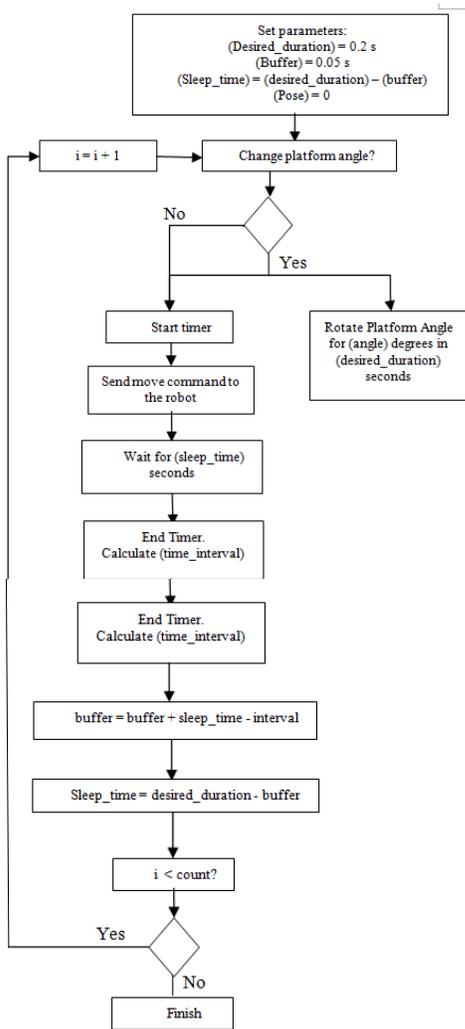


Figure 9: Algorithm for robot execution

#### IV. EXPERIMENT

The whole process from scanning, post processing, path planning, until execution has been conducted, and the result of each process can be seen in Figures 10-12.

The scanned result obtained from Kinect was sufficiently accurate for the current application purpose, as the taping head could follow the surface of the workpiece closely for the whole trajectory. The only exception was at the sharp edges, where bumps are formed near the edges, affecting the normals at those locations greatly. Some care should be exercised at those locations, especially in the path planning. Post-processing of the data will help to rectify this problem, for example by smoothening the normals.

In Figure 11a, the global planning can be seen to follow the outline of the object very closely. The local path planning in Figure 11b shows similar pattern as in Figure 8b. It is divided into two phase, phase 1 (the taping tool traverses the workpiece surface 90° CCW, step 1 and 3 in the figure) and phase 2 (the robot and the platform rotate back 90° CW, step

2 and 4). Figure 11b is not as “clean” as in Figure 8b, though, where only 4 distinct lines can be observed. This is due to the deformation in the object used for this experiment (caused by the pressure applied by the taping head on the object), resulting in slight changes of the horizontal cross section of the object.

The actual execution in the robot can be seen in Figure 12 where phase 1 is captured in a – c, and phase 2 is captured in d – f. The points on the object surface were successfully traversed closely by the taping tool. The whole object can be taped in two to four minutes.

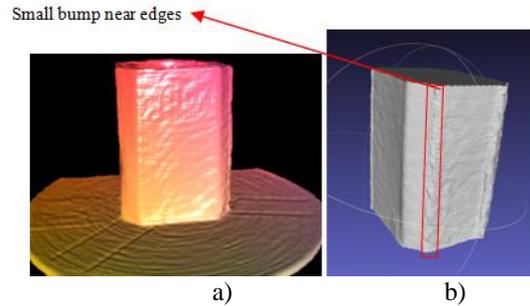


Figure 10: a) Scanned result from Kinect Fusion b) The 3D model of the workpiece after post processing

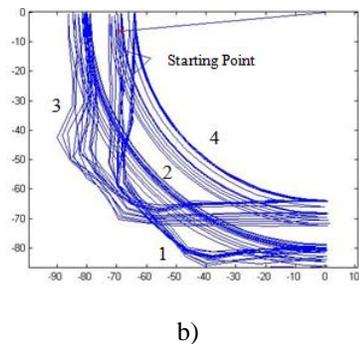
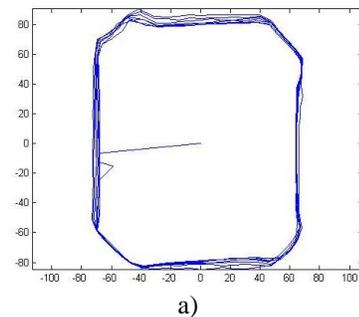


Figure 11: Path Planning result a) Global b) Local

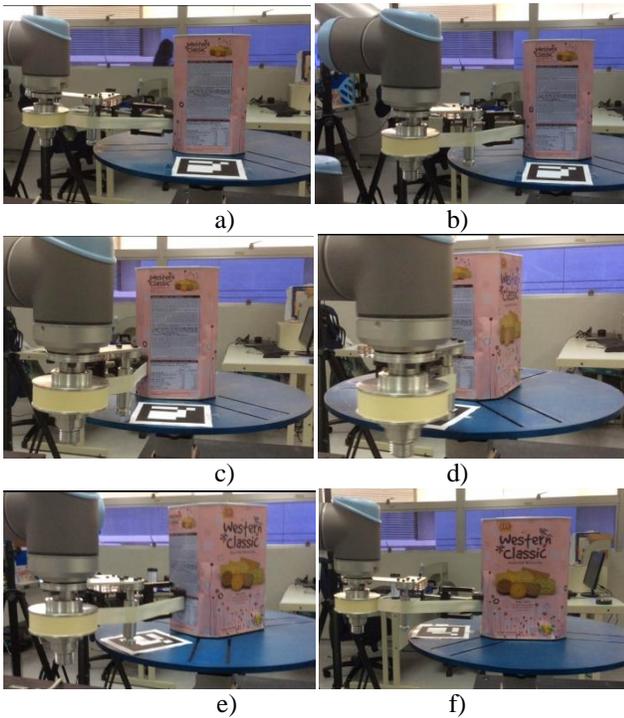


Figure 12: The actual execution on the robot: a-c) Phase 1, the taping tool traverses the workpiece surface  $90^\circ$ CCW. d-f) Phase 2, the taping tool and the platform rotate back  $90^\circ$ CW

## V. CHALLENGES

There are three main challenges in automatic taping application:

1. **The taping mechanism.** For taping complex surfaces, even manual taping requires careful and time-consuming effort to tape the object nicely. The taping cannot be done continuously, but should be done in small patches. Changes on the mechanism should be done to allow automatic attachment of the tapes onto the surface, and automatic cutting of the tape.

2. **Path planning for complex surfaces.** The method that is discussed here for path planning is still a fundamental one, without any optimization related to the shape of the object. One main difficulty in path planning for taping is the taping process itself: the movement of the tape, unlike thermal spraying or painting tool, is constrained to move within a small range of angle with respect to its orientation. Further study on the taping process and its constraints should be conducted to improve the path planning method.

3. **Accuracy.** There are a lot of sources of inaccuracy in the application: the scanning error, the coordinate transformation error, the robot positioning error, and the motor backlash. Although the compliance in the taping tool helps to compensate for some of these errors, further effort should be done to increase the accuracy.

## VI. CONCLUSION

This paper introduced an automated taping system as an example of integration between scanning, path planning and robot execution. The system consists of a taping tool attached to an industrial robot arm, a rotating platform, and a 3D scanner. Some experimental results have been presented, and several challenges (taping mechanism, more sophisticated path planning for complex surfaces, and decreasing inaccuracies) have been identified. Further research should be done on the taping process mechanism, so that the method can be applied to more complex geometry. The whole framework presented here can also be applied to other robotic application such as painting, thermal spraying, and other application requiring complex trajectory which depends on the workpiece geometry.

## ACKNOWLEDGMENT

This work was supported in part by the Agency for Science, Technology and Research, Singapore, under SERC Grant 12251 00005.

## REFERENCES

- [1] T. Hong, M. Ghobakhloo, and W. Khaksar, "Robotic Welding Technology TS," *Compr. Mater. Process.*, vol. 6, pp. 77–99, 2014.
- [2] Aydan Kutay and L. Weiss, "Economic Analysis of Robotic Operations: A Case Study of a Thermal Spraying Robot," *Robot. Comput. Manuf.*, vol. 9, no. 3, pp. 279–287, 1992.
- [3] G. Wittenberg, "Developments in offline programming: an overview," *Ind. Robot An Int. J.*, vol. 22, no. 3, pp. 21–23, 1995.
- [4] S. Deng, Z. Cai, D. Fang, H. Liao, and G. Montavon, "Application of robot offline programming in thermal spraying," *Surf. Coat. Technol.*, vol. 206, pp. 3875–3882, 2012.
- [5] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robot. Comput. Integr. Manuf.*, vol. 28, no. 2, pp. 87–94, Apr. 2012.
- [6] W. Chen and D. Zhao, "Path Planning for Spray Painting Robot of Workpiece Surfaces," vol. 2013, 2013.
- [7] D. C. Conner, S. Member, A. Greenfield, P. N. Atkar, A. A. Rizzi, and H. Choset, "Paint Deposition Modeling for Trajectory Planning on Automotive Surfaces," vol. 2, no. 4, pp. 381–392, 2005.
- [8] P. N. Atkar, "Uniform Coverage of Automotive Surface Patches," *Int. J. Rob. Res.*, vol. 24, no. 11, pp. 883–898, Nov. 2005.
- [9] R. Gadow, A. Candel, and M. Floristan, "Optimized robot trajectory generation for thermal spraying operations and high quality coatings on free-form surfaces," *Surf. Coat. Technol.*, vol. 205, pp. 1074–1079, 2010.
- [10] D. Chang, D. Son, J. Lee, D. L. N, T. Kim, K.-Y. Lee, and J. Kim, "A new seam-tracking algorithm through characteristic-point detection for a portable welding robot," *Robot. Comput. Integr. Manuf.*, vol. 28, pp. 1–13, 2012.
- [11] H. Zhang, H. Lu, C. Cai, and S. Chen, "Robot Path Planning in Multi-pass Weaving," vol. 88, pp. 351–359, 2011.
- [12] R. A. Newcombe, D. Molyneaux, D. Kim, A. J. Davison, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking \*," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, 2011, pp. 127–136.
- [13] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of Kinect depth data for indoor mapping applications.," *Sensors (Basel)*, vol. 12, no. 2, pp. 1437–54, Jan. 2012.