

# Cyber-physical-robotic System (CPRS)-based Modeling and Execution of Assembly Tasks

Yu-Lun Chang,\* Chyi-Yeu Lin, and Jen-Hao Yang

Department of Mechanical Engineering, National Taiwan University of Science and Technology,  
43 Keelung Road, Section 4, Taipei 10672, Taiwan, Republic of China

(Received October 26, 2017; accepted June 18, 2018)

**Keywords:** vision system, cyber-physical-robotic system, CAD model, robot assembly, BPR

In this study, we develop an autonomous process of conducting an assembly task by attaching systematic assembly sequence information to an object's computer-aided design (CAD) model. The information registered in the CAD model is then captured to arrange the robot trajectory, programmable logic controller commands, part three-dimensional printing, and actuator/vision system commands. The vision system can feedback information to modify the CAD model, the operating system, and the position errors in the robot space to achieve the cyber-physical-robotic system (CPRS) closed-loop control. This innovative system can reduce the fixture implementation time, the (robot) manual teaching time, the error variation, and the demand for robot teaching experts in production lines in fast-changing and easy expansion situations. The capability of automatic trajectory generation will increase the precision of the operation to extend the work range from tasks such as loading-unloading, pallet, spray, and welding, which rely on a tolerance of about  $\pm 0.5$  mm, to assembly, which relies on a tolerance of  $\pm 0.1$  mm.

## 1. Introduction

In recent years, industrial automation from the third industrial revolution (factory automation and flexible manufacturing systems) evolved into Industry 4.0,<sup>(1)</sup> of which robots are an important component, but failed to include computer numerical control (CNC) machine tools such as product development and sample preparation process, and can be transplanted to a mass production line with work coordinates set on different CNC machine tools. When an order changes, the material supply chain is abnormal, business process reengineering (BPR) or optimization of processing flow causes process change design, and different processing programs can be applied to hardware-compatible CNC machine tools without modification and easily change the production process. Because there is still an absolute space error of several millimeters (max. 2.1 mm) on the robot under the rated load, the maximum error after correction is 0.696 mm.<sup>(2)</sup> Such error limitation restricts the robot to work that requires less precision. If the production line requires work adjustment due to scheduling, all working

---

\*Corresponding author: e-mail: d9703105@mail.ntust.edu.tw  
<https://dx.doi.org/10.18494/SAM.2018.2013>

points require manual inspection and reprogramming of each robot to conform to operation requirements, thereby undermining line change efficiency, increasing changes in the manual adjustment of various types of equipment, and reducing the processing capability.

Conventional robot programming is divided into online programming and offline programming (OLP).<sup>(3)</sup> Online programming uses programming tools to program working points to conform with robot specifications and precision limitations, so the robot's production capacity decreases during programming and undermines production efficiency. Because only professional personnel can reduce programming error, the dispatch of professional personnel and online programming will become bottlenecks in production. OLP sets the work trajectory in three-dimensional (3D) software, but site operation still requires professional personnel to perform certain portions of the programming operation. This article refers to the fourth stage of the industrial revolution<sup>(1)</sup> and CNC processing programming and adds the cyber-physical-robotic system (CPRS) to the site operation to reduce the workload of professional personnel and improve process capability.

This paper is organized in five sections. In Sect. 2, we present the current programming of the robot. In Sect. 3, we use a CPRS to improve the OLP. In Sect. 4, we describe the experiment and give a discussion. A summary of the study and future research trends are presented in Sect. 5.

## **2. Programming Methods for Industrial Robots**

Conventional robot programming is divided into online programming and OLP, but both methods have shortcomings that make duplication of the production line and expansion of production or workstation process adjustment less convenient than CNC machine tools since the advent of the third industrial revolution of the robot. In exploring the tolerance of the robot and CNC machine tools, we seek an appropriate programming method for the robot so a production line for assembling the robot can duplicate production capacity as easily as a CNC production line.

### **2.1 Robot tolerance**

Most robot controllers' forward kinematics use the Denavit–Hartenberg matrix parameter as a value on a specification sheet, but geometric and dimensional errors that result from the manufacture and assembly of robots cause the coordinates and controls on the robot end effector to have an error of 0.7 to 2.1 mm<sup>(2)</sup> for the coordinates calculated from a Denavit–Hartenberg matrix and each joint encoder feedback angle. The absolute precision in space of the robot is a maximum of 2.1 mm before calibration and a maximum of 0.7 mm after calibration. Such error has limited robot applications to loading, unloading, spraying, welding, and stacking with a comparatively larger tolerance in work while making it unfavorable for assembly with a smaller tolerance in workstation task adjustment and process change. At production expansion, the robot program of the original production line cannot be duplicated to the new production line and goes into immediate production because reprogramming is required to eliminate the

error caused by different robots from the allocation of each workstation's tasks. The robot's maximum angular error of  $\pm 0.1^{\circ(2)}$  is acceptable for a general assembly process.

## 2.2 CNC error

According to ISO 10791-1/ISO10791-4.2 Travel X/Y/Z: 900/610/610 mm, the CNC machine tool space error is 0.02 mm or less, and its precision is 0.01 mm or less;<sup>(4)</sup> therefore, the scheduling of the CNC machine tool production line in the machine room can still produce workpieces that conform to general assembly error. Production expansion can be accomplished by duplicating its original program to a compatible CNC controller.

## 2.3 Online programming

Online programming is divided into two types. One is to use the robot's teaching device to command every task of the arm, and the operator visually checks whether the arm has reached the working point. Upon reaching the working point, the robot coordinates and posture are recorded in the temporary controller working point register. Because most robots can only adjust the percentage of movement to control the amount of feed, each different type of robot has a different movement rate. Another method of robot operator programming is to lead-through the robot to maintain the robot's ideal position and location.

Both methods require prolonged training of an operator with inconsistent error of each training, thereby decreasing the production line's processing capability, increasing the defective yield, and taking up a considerable portion of the robot's operation time. An unskilled robot operator can easily crash the machine or make the production process too time-consuming.

## 2.4 OLP

OLP cannot fully automate the process of computer-aided design (CAD) to robot programming,<sup>(3)</sup> even though the computer-aided manufacturing (CAM) system developed before the OLP must be added to the manual selection of tools, workmanship, cutting point, feed mode, feed rate, and other parameters. When a product contains more and more parts, its life cycle is short, and its design changes frequently, and engineers must complete the design changes in the fastest robot assembly trajectory planning. Online programming takes up robot production time and makes production less efficient. Therefore, most robot factories have configured OLP software, but the function is not easy to expand. Self-development is not subject to factory restrictions. Self-development can be divided into two categories: the use of CAD software application programming interface (API) functions such as Autodesk Inventor,<sup>(5)</sup> FreeCAD,<sup>(6)</sup> SolidWorks,<sup>(7)</sup> and AutoCAD,<sup>(8)</sup> and the use of a drawing library or ROS.

Compared with the teaching method of the CNC machine, the traditional teaching process of industrial robots does not know the teaching target error amount and the industrial arm space precision is insufficient; therefore, new methods must be developed to perform the assembly task.

### 3. CPRS

References 2, 9, and 10 used laser trackers to measure the absolute coordinates of the ends of robots, and it has been shown that the absence of sensors on the robots for feeding back the absolute coordinates of the end space of the robots requires an additional sensing system. Most OLPs generate industrial robot programs by the open-loop method.<sup>(3)</sup> Even if each step is optimized, it does not solve the absolute accuracy error of robots.

Most commercially available OLP systems only work with large-tolerance tasks such as loading and unloading, stacking, painting, and welding. CPRS changes the traditional OLP workflow process by business process reengineering, as shown in Fig. 1. It reduces the setting time of tags. The commands run the same tasks in different industrial robots. The amount of tag information of an assembly task is more than that of a general task. Adding tag information in the part file is shown in Fig. 2. This reduces the set time of the tag information of the assembled parts. The traditional OLP is modeled first, and the tag is then changed to the same stage execution. The vision system of the CPRS flowchart is shown in Fig. 3.

The CPRS hardware architecture is shown in Fig. 4. Users send g-codes of parts to the 3D printer in the PC's CAD software. Assembly information is added to the CAD drawing system, and CAD software API function analysis assembly information is used to generate a programmable logic controller (PLC), gripper, vision system, ROBOT, and other actuator commands transmitted via Ethernet to the PLC. The PLC follows the order control ROBOT and vision system via Ethernet to complete the position feedback control system. The PLC's I/O port controls a gripper and an actuator to finish I/O action.

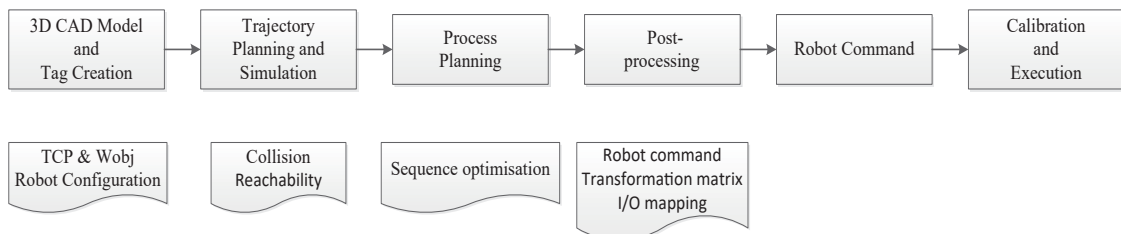


Fig. 1. Key steps of CPRS.

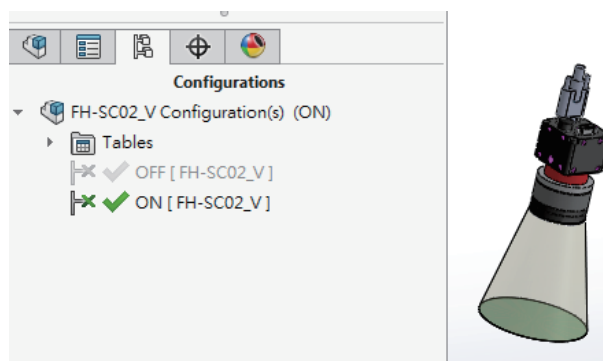


Fig. 2. (Color online) Visual component configuration.

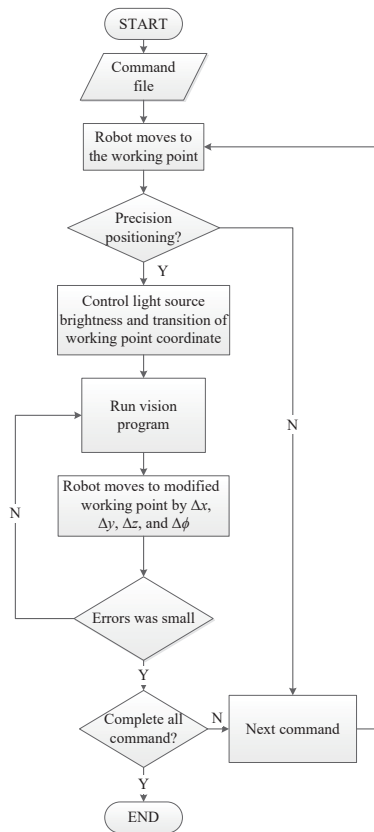


Fig. 3. Vision system of CPRS flowchart.

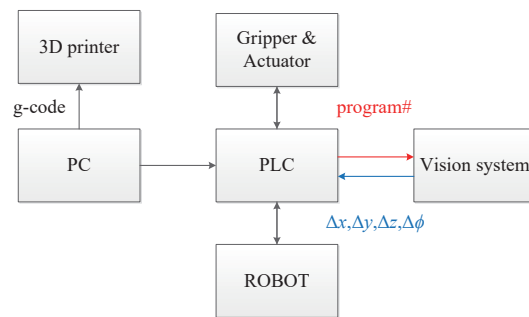


Fig. 4. (Color online) CPRS hardware architecture.

### 3.1 3D CAD model and tag creation

In recent years, the computing power of CPUs and graphics processing units (GPUs) has grown considerably. The computer hardware and software in most design departments of a modern company meet the requirements for 3D simulation of automated design. The suppliers of parts or equipment take the initiative to provide evaluation-specific 3D model appearance for users to evaluate the design and reduce sample costs and time to lease, and upon completion of the design, it naturally eliminates the chances of incompatible installation. Workpieces not available for 3D modeling can be modeled using 3D scanning<sup>(11,12)</sup> or manual measurement of assembly dimensions. Tag creation is usually a time-consuming process and can require secondary programming for automatic tag recognition.<sup>(2)</sup> The 3D CAD models provided by the robot manufacturers often lack interconnection details. Therefore, we must define the interconnection details between every connecting joint and define the zero angle position for each joint to allow the detection of all robot joint angles in the model space.

In 1988, owing to the hardware and software limitations of PCs at that time, 3D simulation could not be carried out. Robot Assembly Language Planner in Harmony (RALPH)<sup>(13,14)</sup> was built using 3D CAD models based on constructive solid geometry. RALPH is not intuitive. Traditional OLP would take a lot of time on tag creation after building 3D CAD models. CPRS allows the addition of basic tag information in the part-model building stage. The trajectory planning of an industrial robot that involves the part models can be edited in a much more intuitive manner.

Most CAD software can set up multiple configurations with a particular part drawing. On the basis of the requirements, variable values or shape features would be specified in each configuration. For example, various types of bearings can be defined by different configurations, which means that different types of bearings can be chosen simply by choosing the bearing configuration in the combination drawing. The combination section of configuration data and product database is convenient for drawing file management.

The method of extending the configuration settings of parts is to set the main assembly order in the model configuration, as shown in Fig. 5. The operator refers to Tables 1 and 2 to input the sub-assembly instruction into the description column of each main assembly configuration, as shown in Fig. 6. The actuator and the state of the visual system are set as shown in Table 3. The variable

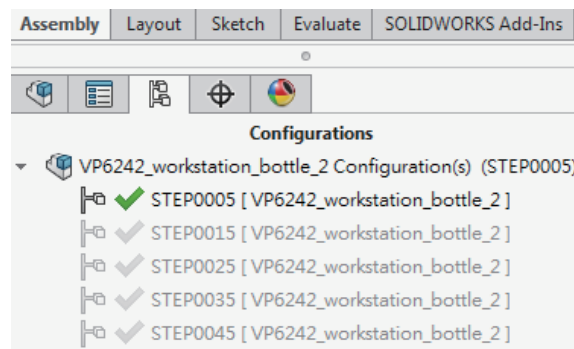


Fig. 5. (Color online) Assembly sequence configuration.

Table 1  
Command types.

| Command | Description  |
|---------|--|
| T       | ROBOT moves to work point.   |
| V       | Visual system correction $x$ , $y$ , $z$ , and $\phi$ of work point. |
| G       | Gripper controlled.  |
| F       | Fixture controlled.  |
| P       | Program paused.  |

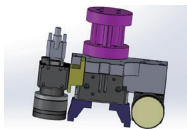
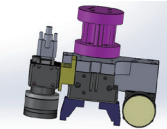
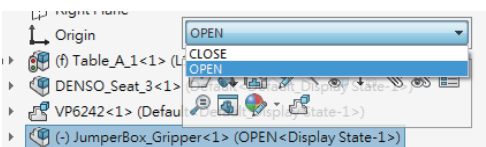
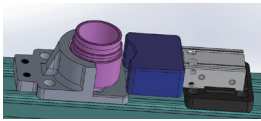
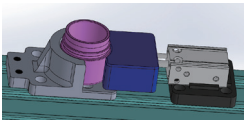
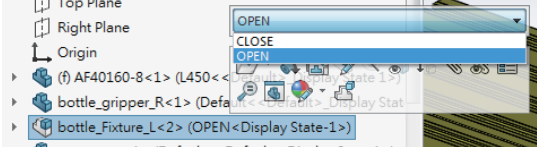
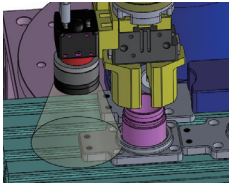
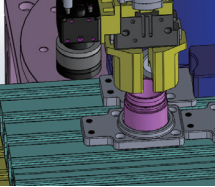
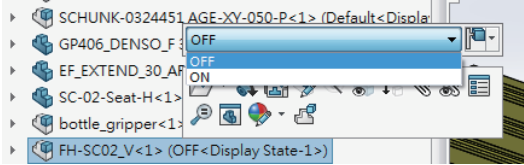
Table 2  
Industrial robot movement form.

| Type | Subassembly steps   |
|------|---|
| T000 | Move to operating point <b>P</b> .  |
|      | 1. Generate a visual work point <b>P'</b> and a forward insertion point <b>P''</b> .  |
|      | 2. Robot moves to <b>P'</b> , performs visual correction:<br>$\mathbf{P}' = [P'_x + \Delta x, P'_y + \Delta y, P'_z + \Delta z]$ and $\phi' = \phi' + \Delta\phi$<br>$\mathbf{P}'' = [P''_x + \Delta x, P''_y + \Delta y, P''_z + \Delta z]$ and $\phi'' = \phi'' + \Delta\phi$<br>$\mathbf{P} = [P_x + \Delta x, P_y + \Delta y, P_z + \Delta z]$ and $\phi = \phi + \Delta\phi$ |
| T001 | 3. Move to <b>P''</b> point.<br>4. Move straight to <b>P</b> .<br>5. Do subfunction.<br>6. When subfunction is finished.<br>Move straight to <b>P''</b> .<br>7. Move to <b>P'</b> point.  |
| T002 | Same as T001 1–6 steps.   |
| T003 | Move straight to operating point <b>P</b> .   |



Fig. 6. Define command at description of assembly sequence configuration.

Table 3  
(Color online) Part configurations.

| Item    | Options   | Setting method   |
|---------|---|--|
| Gripper |       |    |
| Fixture |     |   |
| Vision  |   |  |

preinstallation height and visual height are equal to the line insertion assembly distance and the working height of the visual system. The camera’s front end taper height is the working height of the vision system. The green circle under the cone bottom is used to determine whether the landmark deviates from the visual detection range, as shown in Fig. 2. The statuses of the gripper, fixture, and visual system are set as shown in Table 3. The tag of the main assembly calibration has been completed, as shown in Fig. 7 and Table 4.

To implement the proposed CPRS system, the following coordinate systems in the CAD model must be established, as shown in Fig. 8.

- ROBOT coordinate system: the robot program depends on this coordinate system.
- Camera coordinate systems (CCD1 and CCD2): for visual correction of error.
- Flange surface coordinate system (E.F.): the origin of the coordinates is the program point **P**, the rotation matrix can find the Euler angle, and the axis angles can be obtained from the CAD model.
- Work coordinate system (workpiece): for setting up the workpiece.

SolidWorks API can obtain  ${}^{CAD}_{robot}H$ ,  ${}^{CAD}_{E.F.}H$ ,  ${}^{CAD}_{CCD1}H$ ,  ${}^{CAD}_{CCD2}H$ , and  ${}^{CAD}_{workpiece}H$ . Because robot



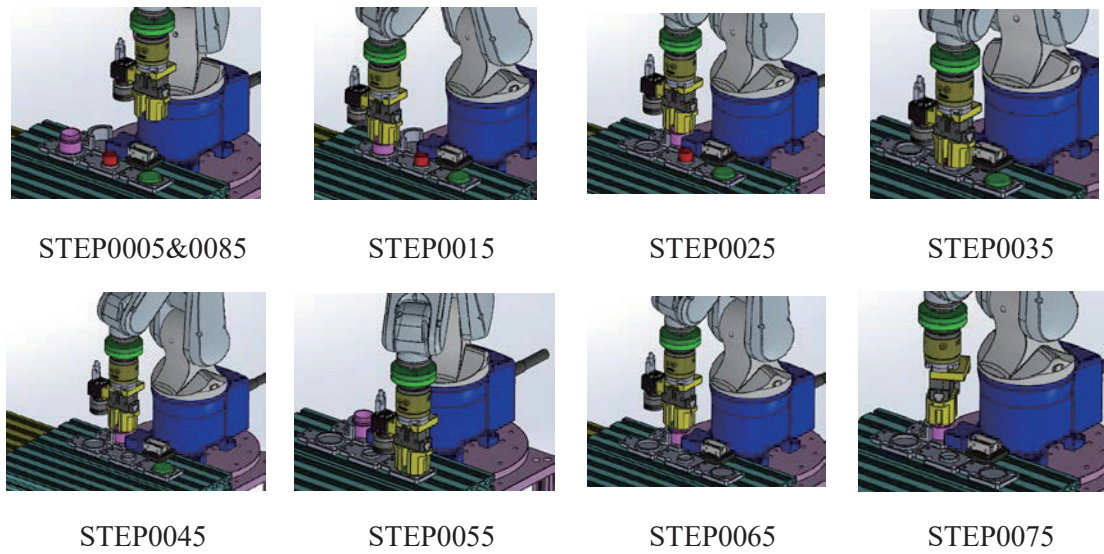


Fig. 7. (Color online) Assembly key process.

Table 4  
Configuration of bottle-filling assembly.

| Step No. | Task             | Description          |
|----------|------------------|----------------------|
| STEP0005 | Initial position | T000_G_P010          |
| STEP0015 | Pick bottle      | T001_V01_G_P0300     |
| STEP0025 | Place bottle     | T001_V01_G_P0300_F   |
| STEP0035 | Take filler      | T001_V01_G_P0300     |
| STEP0045 | Place filler     | T001_V01_G_P0300     |
| STEP0055 | Pick cap         | T001_V01_G_P0300     |
| STEP0065 | Place cap        | T002_V01             |
| STEP0075 | Lock cap         | T003_G_P0300_F_P0300 |
| STEP0085 | Initial position | T000                 |

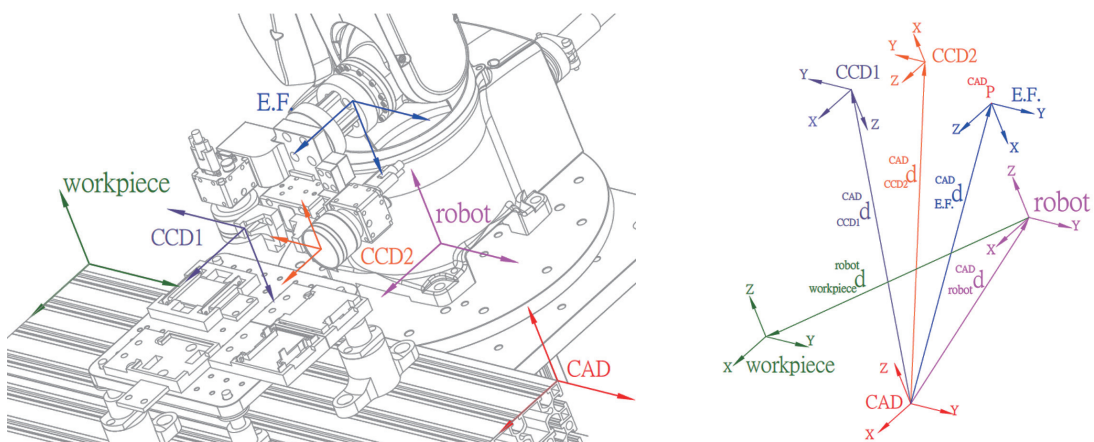


Fig. 8. (Color online) Coordinate system.



commands are based on the robot coordinate system, the coordinates of  $\mathbf{P}$  points at various steps are first converted to the robot coordinate system by Eq. (3) and then modified by Ref. 15 to ( $\beta = \frac{\pi}{2}$  or  $-\frac{\pi}{2}$ ), and finally, the Euler angle is calculated with Eq. (4).

$${}_{CAD}^{robot} \mathbf{H} = \begin{bmatrix} {}_{robot}^{CAD} \mathbf{R}^T & -{}_{robot}^{CAD} \mathbf{R}^T {}_{robot}^{CAD} \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}_{CAD}^{robot} \mathbf{d} = -{}_{robot}^{CAD} \mathbf{R}^T {}_{robot}^{CAD} \mathbf{d} \quad (2)$$

$${}_{robot}^{CAD} \mathbf{P} = {}_{robot}^{CAD} \mathbf{R} {}_{CAD} \mathbf{P} + {}_{robot}^{CAD} \mathbf{d} = {}_{CAD}^{robot} \mathbf{H} {}_{CAD} \mathbf{P} \quad (3)$$

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}(z, \alpha) \mathbf{R}(y, \beta) \mathbf{R}(x, \gamma) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c\alpha c\beta & cas\beta s\gamma - s\alpha c\gamma & cas\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & sas\beta s\gamma + c\alpha c\gamma & sas\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix},$$

where  $\cos\theta = c\theta$ ,  $\sin\theta = s\theta$ . When  $r_{11} < 0.001$  and  $r_{21} < 0.001$ , then

$$\begin{cases} \alpha = 0, \\ \beta = \text{atan2}(-r_{31}, r_{11}), \quad \beta = \frac{\pi}{2} \text{ or } -\frac{\pi}{2}, \\ \gamma = \text{atan2}(r_{11}, r_{22}). \end{cases}$$

Otherwise,

$$\begin{cases} \alpha = \text{atan2}(r_{12}, r_{11}), \\ \beta = \text{atan2}\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right), \\ \gamma = \text{atan2}(r_{32}, r_{33}). \end{cases} \quad (4)$$

### 3.2 Trajectory planning and simulation

The assembly order and instructions can be obtained by API functions after reading the assembly model's configuration data. Table 1 lists the assembly instructions, T, V, G, F, and P. Table 2 lists the conversion from industrial robot movement to subassembly configuration. Traverse the tree of feature manager of each assembly to find the name of the part, which includes “\_G” as a gripper, “\_F” as a fixture, and “\_V” as the vision system. The configuration of the part acquired determines whether the part is involved in the action. The 3D model is modified according to the preinstall height and visual height. The robot motion is then checked in terms of reachability and collisions. Each configuration of the gripper, the fixture, and the visual system status constitutes the assembly command. The STEP0055-generated configuration of the subassembly is shown in Table 5.

Table 5  
STEP0055 of assembly process expansion.

| Sub step No. | Main Step        | Related command | Description                                |
|--------------|------------------|-----------------|--|
| STEP0055.1   |                  | T001            | See Table 3 T001.1                         |
| STEP0055.2   |                  | V01             | Run vision program 01 to do Table 3 T001.2 |
| STEP0055.3   |                  | T001            | See T001.3 of Table 3                      |
| STEP0055.4   | T001_V01_G_P0300 |                 | See T001.4 of Table 3                      |
| STEP0055.5   |                  | G               | Change status of gripper                   |
| STEP0055.6   |                  | P0300           | Delay 0.3s                                 |
| STEP0055.7   |                  |                 | See T001.6 of Table 3                      |
| STEP0055.8   |                  | T001            | See T001.7 of Table 3                      |

### 3.3 Process planning

The movement sequence of the industrial robot has been set in the assembly model configuration. The visual assistant positioning system can find the optimal trajectory of the robot by solving a traveling salesman problem. References 16–18 provide examples to solve the traveling salesman problem using genetic algorithms. The vision system is switched off to increase the assembly speed after recording the corrected position in every work point that requires vision correction. The vision system is turned on when the sampling test of statistical process control is executed.

### 3.4 Postprocessing and robot command

The I/O control of the gripper, the I/O control of the fixture, and the name of the visual inspection program must correspond to the name of the external controller. CPRS generates industrial robot position movement instructions. The controller of the industrial robot performs a written explanation CPRS instruction program. Various industrial arms can perform the same instruction without postprocessing. Instructions include the transfer matrix for the vision system to fix the robot space error.

### 3.5 Calibration and execution

In this paper, the maximum error of the pixel size correction of the visual system is 0.0268 mm, using two solid dots of 6 mm diameter marked at a distance of 20 mm with the tolerance of the workpiece placed at a  $\pm 20$  mm error range. Foci of the CCD and light source have been properly adjusted in advance, and the center of two circles, the position of the visual system, and the camera work height are registered, as shown in Fig. 9.

The traditional teaching process requires the fixture to be fixed to a working surface. The correction process is time-consuming and is subject to potential collision. In CPRS, the same landmarks described in the previous paragraph are generated on the fixture. During the calibration process, the fixture is placed on the desk, and the industrial robot clamps the workpiece to be on top of the fixture. The fixture position is manually adjusted so that the workpiece is tightly fit for a quality insertion to the fixture. The fixture is then fixed, and

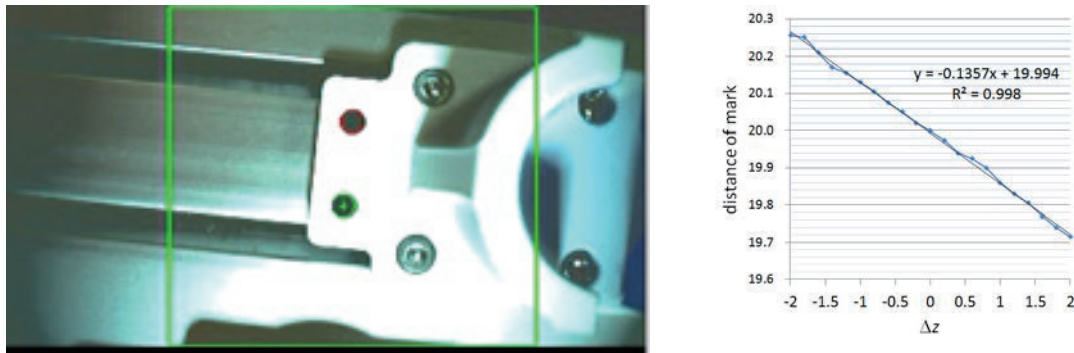


Fig. 9. (Color online) Height-modified experimental result (unit: mm).

the industrial arm moves up to the default height of the visual system (camera). The camera system now registers the landmark coordinates as the standard values of this fixture, which will be efficiently used when a new robot is used to perform the insertion in a new production line. This proposed process, which can be used with other robots, will save considerable time compared with the traditional teaching process, which is only valid with the current robots.

After the registration process is completed, the program can be used in any new production line with a new robot. For a new robot, the visual system is moved to a preset position and allowed to detect the distance of the two landmarks (solid dots) and the angle between the horizontal axis and the line connecting two dots, and then return error values of  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$ , and  $\Delta\phi$  of the  $Z$ -axis. These error values are substituted into Eq. (5) to obtain the displacement correction and into Eq. (4) to obtain the angle correction for the new industrial robot. The visual servoing process will use these error values to minimize the camera position errors to reach the same relative position for successful insertion. When the error is less than 1.8 mm or  $0.2^\circ$ , the required movement of the industrial robot in the visual servoing needs to be cut in half to ensure rapid convergence.

$${}^{CCD1}\mathbf{E} = \begin{bmatrix} \cos\Delta\phi & -\sin\Delta\phi & 0 & \Delta x \\ \sin\Delta\phi & \cos\Delta\phi & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ where } \Delta x, \Delta y, \Delta z, \text{ and } \Delta\phi \text{ form visual system} \quad (5)$$

$${}^{robot}\mathbf{E} = {}^{robot}_{CAD}\mathbf{H} {}^{CAD}_{CCD1}\mathbf{H} {}^{CCD1}\mathbf{E}$$

#### 4. Experiment

We will conduct two assembly tasks in the experiment. For each assembly task, the operation of the robot teaching process is first performed so that the landmark positions in the camera system for each single insertion are registered. This robot assembly task can then be implemented in a new production line with a new robot with satisfactory efficiency. The CPRS system will move the robot with the camera system to the default position based on

the positions of the fixture and the robot position defined in the CAD model. The robot with the camera system will perform a new calibration operation to move it to a precise position at which the landmarks appear at identical positions to the camera. The end-effector position is registered as the new assembly position of this robot for the particular task. When the end effector position is registered, the six joint angles are recorded. The robot will then repeatedly perform the assembly task at the same location with high repeatability and accuracy. No further vision calibration will be needed at the routine operation stage. The two common assembly tasks selected include the installment of the printed circuit board assembly (PCBA) into the case of a jumper box and the screw-tightening operation of a bottle cap onto a bottle. In this experiment, the fixture of two operations was randomly placed in the work area to show its robust applicability.

#### 4.1 Stack assembly

The first experimental target is a jumper box that consists of three objects—the upper case, the middle PCBA, and the lower case—stacked together. The assembly accuracy required in this task is  $\pm 0.1$  mm. The end effector of the robot comprises the camera system, which consists of two industrial cameras and one gripper with a pair of specially designed L-shaped clamps for picking up two objects of different sizes, as shown in Fig. 10(a). Before the calibration, the assembly fixture is placed with a positioning error of  $\pm 20$  mm compared with the precise location defined in the teaching process. The calibration process will move the robot to the correct position by using visual servoing techniques to minimize the landmark location errors. After calibration, the robot will use the newly obtained trajectories to perform the assembly sequence. The entire assembly includes the insertion of the lower case into the fixture, the placement of the PCBA into the fixing recesses of the lower case, and finally the insertion of the upper case into the integrated set of the lower case and the PCBA. The CPRS-guided assembly procedure was proven successfully in the assembly task, with an assembly accuracy requirement of less than  $\pm 0.1$  mm. A video that shows the entire assembly process can be seen in Ref. 18.

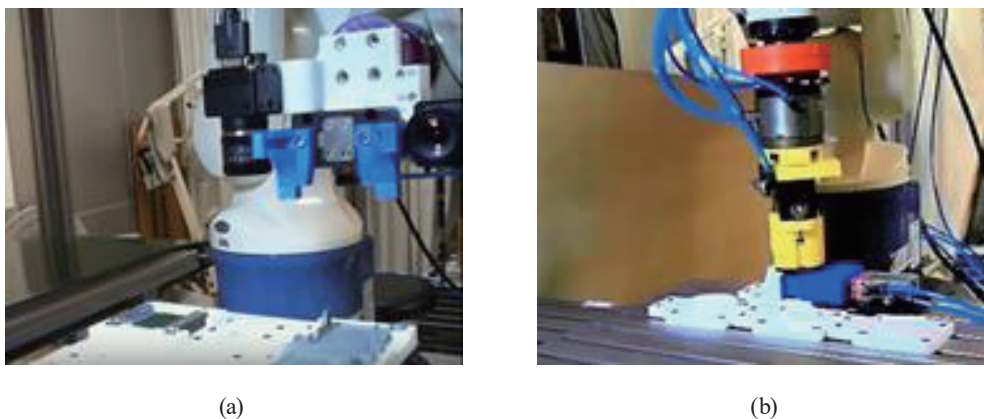


Fig. 10. (Color online) (a) Jumper box assembly and (b) bottle-filling assembly.

## 4.2 Bottle-filling assembly

The second experimental target is the glass bottle, a threaded bottle cap, and a cylindrical object to be put inside the bottle, as shown in Fig. 10(b). This bottle-filling assembly requires an assembly accuracy of  $\pm 0.1$  mm. The end effector set contains industrial cameras, grippers, and a pneumatic jig to fix the glass bottle. Before the calibration, the assembly of the jig is set to a location within the range of  $\pm 20$  mm. During the calibration process, the positioning errors are minimized with the visual servoing techniques based on the observed landmark errors captured by the camera. The corrected robot positions are recorded. The robot assembly procedure is performed by all registered robot positions and matching trajectories.

The entire assembly includes the pick and placement of the glass bottle into the pneumatic jig, the pick and placement of the cylindrical object into the glass bottle, and finally, the pick, placement, and screw-fastening of the bottle cap into the bottle. Again, the proposed CPRS-guided assembly procedure was proven successfully in the assembly task with an assembly accuracy requirement of less than  $\pm 0.1$  mm. A video that shows the entire assembly process can be seen in Ref. 19.

## 4.3 Results and discussion

During the calibration stage when a new production or a new robot is used, the first requirement is to fix the fixture within a zone that has a maximum position error of  $\pm 20$  mm compared with the position defined in the CAD model. The robot will be automatically guided to be atop the fixture based on the CAD information. The calibration will then be fully autonomously performed at each assembly step and generate the precise robot assembly positions, which are registered in joint angles. After this, the setup on the new robot or new production line is completed. The robot can perform all required assembly tasks with high repeatability accuracy without the need of any visual adjustment.

The tolerance of the stacked assembly was  $\pm 0.1$  mm. To ensure high reliability supported by sufficient position accuracy, the convergence threshold value of the position error during the visual servoing operation was set as  $\pm 0.04$  mm and  $\pm 0.1^\circ$ . With such vision landmark calibration level, 20 repeated stack assembly operations were 100% successful. The statistics of the visual servoing operation for calibration can be seen in Table 6.

The acceptable tolerance of the bottle-filling assembly was also  $\pm 0.1$  mm. Because no angular adjustment is involved in the thread-screwing operation, the convergence threshold value of the position error during the visual servoing operation was set to a larger value of  $\pm 0.08$

Table 6  
Statistics of stack and helical assembly after 20 repetitions.

| Type             | Visual servoing set value | Error    |          |         |          |
|------------------|---------------------------|----------|----------|---------|----------|
|                  |                           | Avg.     | Min.     | Max.    | Std.     |
| Stack assembly   | 0.04 mm                   | 0.021 mm | 0.001 mm | 0.04 mm | 0.014 mm |
|                  | 0.1°                      | 0.001°   | 0.001°   | 0.001°  | 0°       |
| Helical assembly | 0.08 mm                   | 0.041 mm | 0.01 mm  | 0.08 mm | 0.017 mm |

mm. Again, with this convergence condition, 20 bottle-filling assembly operations were all successful.

## 5. Conclusions

From the experiments, the CAD-based CPRS system significantly reduces the requirement of a high positioning accuracy on the robot and allows the fixture to have greater positioning tolerance. These advantages then reduce the setup time on a new production line or with a new robot. For example, in the stack assembly case shown in Sect. 4.1, the teaching time used for defining the new task will take 2 to 3 h. By using the proposed CPRS system, it takes only 5 min to teach a new landmark. When part trays are used in the assembly line, the traditional method will even double its teaching time while the time needed in CPRS remains the same with identical landmarks. When the manufacturing process is changed or the production line is extended, the setup time is shortened, the number of variation factors is decreased, the productivity rises, and the strict demand on an experienced engineer for the new setup operation is relaxed.

The operating environment of the CPRS system uses commercial CAD software available in the design department, which will lead to extra software or personnel-training costs. Since the CAD-based CPRS system uses CAD model files as its source content, the relative positions between the robot gripper and the parts can be defined in a fully automated manner. Furthermore, the alternative-part function in the CAD model can be used in the CPRS system when an alternative part is used and then all required new robot assembly commands will also be automatically redefined. The camera-system feedback-based CPRS system will increase the robot's capacity by increasing its positioning accuracy. With these advantages, the CPRS system can extend the work spectrum of the OLP from loading and unloading, stacking, painting, and welding tasks that permit a large tolerance to tasks such as precision assemblies that demand great positioning accuracy.

In the product design stage, the positioning features can be added to proper positions of the parts, the landmarks on the fixture used in the CPRS system will no longer be necessary, and the time spent for creating the landmarks on the fixtures can be saved. These landmarks printed on the products can be further used as features for positioning and measurement for quality control.

## References

- 1 H. Kagermann, W. Wahlster, and J. Helbig: Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0—Securing the Future of German Manufacturing Industry, Final Report of the Industrie 4.0 Working Group, acatech—(National Academy of Science and Engineering, Munich, 2013) p. 13.
- 2 A. Nubiola and I. A. Bonev: *Rob. Comput. Integr. Manuf.* **29** (2013) 236.
- 3 Z. Pan, J. Polden, N. Larkin, S. V. Duin, and J. Norrish: *Rob. Comput. Integr. Manuf.* **28** (2012) 87.
- 4 CNC Milling Machine Specification Sheet: [http://www.quaser.com/files/proimages/1487293121\\_pdf.pdf](http://www.quaser.com/files/proimages/1487293121_pdf.pdf) (accessed October 2017) p. 15.
- 5 P. Neto and N. Mendes: *Robot Auton. Syst.* **61** (2013) 896.
- 6 L. A. Ferreira, Y. L. Figueira, I. F. Iglesias, and M. Á. Souto: *Procedia Manuf.* **11** (2017) 215.
- 7 S. Mitsi, K.-D. Bouzakis, G. Mansour, D. Sigris, and G. Maliaris: *Adv. Mater. Res. Switz* **26** (2005) 262.



- 8 J. N. Pires, T. Godinho, and P. Ferreira: *Ind. Robot* **31** (2004) 71.
- 9 A. Fillion, A. Joubair, A. S. Tahan, and I. A. Bonev: *Rob. Comput. Integr. Manuf.* **49** (2018) 77.
- 10 T. Wei, M. Dongqi, L. Pengcheng, Z. Yuanfan, H. Eng, and Z. Wei: *Chin. J. Aeronaut.* **28** (2015) 946.
- 11 Z. M. Bi and S. Y. T. Lang: *IEEE Trans. Ind. Inf.* **3** (2007) 84.
- 12 C. Y. Lin, L. T. Son, Y. L. Chang, and Y. S. Shiue: *Sens. Mater.* **29** (2017) 935.
- 13 B. O. Nnaji, J. Y. Chu, and M. Akrep: *J. Manuf. Syst.* **7** (1988) 131.
- 14 B. O. Nnaji: *Int. J. Prod. Res.* **26** (1988) 735.
- 15 O. Chocron: Handout No. 2, Course 2.05 (2000), <http://web.mit.edu/2.05/www/Handout/HO2.PDF> (accessed October 2017) p. 1.
- 16 K. Y. Kim, D. W. Kim, and B. O. Nnaji: *IIE Trans.* **34** (2002) 865.
- 17 H. J. Kang and J. Y. Park: In: *Proc. OCEANS'04. MTTs/IEEE TECHNO-OCEAN'04* **1** (2004) 218–222.
- 18 Stack Assembly Video: [https://www.youtube.com/watch?v=CF2yuc8\\_H4k](https://www.youtube.com/watch?v=CF2yuc8_H4k) (accessed October 2017).
- 19 Helical Assembly Video: <https://www.youtube.com/watch?v=Drkw6hwD-iE> (accessed October 2017).

## About the Authors



**Yu-Lun Chang** received his BS and MS degrees from Kun Shan University, Taiwan in 1990 and 2008, respectively. He began his Ph.D. studies at National Taiwan University of Science and Technology, Taiwan, in 2008. He holds many Class-B National Professional Certificates, including Computer Aided Engineering Drawing, Turning-CNC Turning, Machining, Computer Maintenance, Mechatronics, and Digital Electronics. His research interests are robot assembly, mechatronics, and CAD/CAM.



**Chyi-Yeu Lin** is a distinguished professor in the Department of Mechanical Engineering and the director of the Center for Intelligent Robotics in National Taiwan University of Science and Technology (NTUST). At NTUST, he served as both the director of the Center of Technology Transfer and the director of Innovation and Creativity between 2007 and 2009, and the vice-chairman and chairman of the Department of Mechanical Engineering in 2006–2008 and 2009–2012, respectively. He received his Ph.D. degree from the University of Florida in 1991. From 1991 and 2001, his major research interests were structural optimization and evolutionary methods. After 2001, he switched his interests to intelligent robotics. He and his research team created many intelligent robots including two androids and two dual-wheeled robots that performed in the world's first robot theater involving male and female bipedal humanoid robots in December 27, 2008.



**Jen-Hao Yang** received his BS degree from Chang Gung University, Taiwan, in 2014 and his MS degree from National Taiwan University of Science and Technology, Taiwan, in 2017. His research interests are robot assembly, mechatronics, and CAD/CAM.