Development of a robotic manipulator under ROS system

Jin-Siang Shaw and Yen-Ju Lin

Abstract**—**The rise of Industry 4.0 has enabled the robots to flourish both in industry and academia in recent years; for robot users, the robot operating system employed is mainly developed by various robot manufacturers for private use. The main goal of this study is to build a five-axis robotic manipulator using 3D printing technology and then apply the open source ROS (Robot Operating System) platform and MoveIt to control the robotic arm. Consequently, the robot arm URDF model, GUI interface control, motion path planning, real-time actuation and data feedback can be obtained. Furthermore, an eye-to-hand camera and deep learning using Tensorflow are integrated to the developed robot arm. Experimental results validate the ROS-based robot arm for object recognition and sorting task.

Keywords: 3D printing, MoveIt, Robotic manipulator, ROS, RViz,

I. Introduction

With the progress of the time, robots exist in various parts of the world, and the number of sales of robots is increasing year by year. This phenomenon affects the transformation of many companies. The use of robots can effectively improve work efficiency and reduce costs [1][2]. The global industrial robot sales increase in recent years in part due to expensive labor costs in the industry.

At the early stage of robot, the main type of robots is the robotic manipulator. The automobile manufacturing industry was one of the earliest areas the robot arm applied. The first robotic arm was applied to the casting delivery and welding of the production line. After successfully automating the production line, many companies have developed special robot arms for different purposes. The most common "mechanical structure" of industrial robots can be divided into series and parallel structures [3].

The early technology and system construction of 3D printing was quite expensive, so most of them were concentrated in industries with high added value, high precision and engineering. Although there are still some 3D printing manufacturers trying to make smaller and convenient models, there was not much breakthrough until 2007, when the RepRap prototype appeared, which was an open source and self-manufacturing system and opened up new topics and research directions in industrial manufacturing [4].

In recent years, the demand for robotic arms has increased. Because the cost of industrial arms is still high, it is difficult to popularize them in small and medium-sized enterprises, offices, or ordinary families. In this study, by 3D printer technology [5], low-cost robotic arm is to be designed and manufactured. The type of 3D printer adopted is the Delta-type D-Force V3. Through SolidWorks, Cura design robotic arm and printer cutting layer, and the open source ROS (Robot Operating System) platform, we intend to build a robotic arm simulation model, planned arm path as well as real-time arm path to realize the path implementation and received route feedback data. In addition, Tensorflow deep learning with eye-to-hand camera architecture is to be integrated into the system. This whole system is an open source robot architecture, which can be embedded with more open source packages and algorithms to make the robot system more flexible to use.

J. Shaw is with the Mechanical Engineering Department, National Taipei University of Technology, Taipei, Taiwan, (e-mail: jshaw@ntut.edu.tw).
Y. J. Lin is with the Institute of Mechatronic Engineering, National Taipei University of Technology, Taipei, Taiwan, (e-mail: jimmy15121544@gmail.com).

II. Design and Construction

A. Five-axis Robotic Manipulator

In this study, a five-axis robotic arm is designed including design of each axis, driving method of the motor, and the gripper design, etc., which will be divided into six parts.

1) First Axis

In this paper, each axis of the robot arm uses a T5 wire drive belt to drive the joint movement. The first axis can be rotated at an angle of ±170°. The diameter of the bearing is 16mm in outer diameter and 5mm in inner diameter. The bearing is fixed to the lower part of the belt driving surface with 5mm screws so that the rotation is smooth, and the center of the driving surface is placed in the bearing with an outer diameter of 22mm and an inner diameter of 8mm and is fixed with 8mm screws to ensure that the first axis will not move eccentrically, as shown in Figure 1.

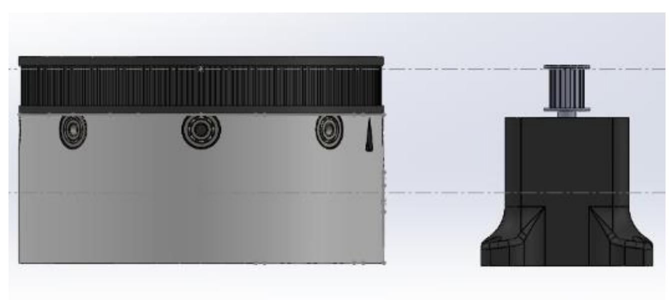


Figure 1. Design of the first axis.

2) Second Axis

The rotation range of the second axis joint is ±90° with two stepping motors of the same model as the driving source. In Figure 2 it can be seen that the belt tensioner is fixed obliquely above the synchronous wheel and the screws are used to tighten the belt to ensure that the arm does not fall when the rotation angle is too large. Meanwhile, the belt tensioner is placed with the bearing sized 13mm outer diameter and the inner diameter 4mm to retract the belt slide. The slider axis is the second axis and the third axis at the same axis. There are six lock points on the outside of the second axis fixed to the first axis, and the slider axis is divided into the last one with a diameter of 22mm and an inner diameter of 8mm. The purpose is to allow the optical axis to rotate smoothly without wearing out the printed objects due to rotation.

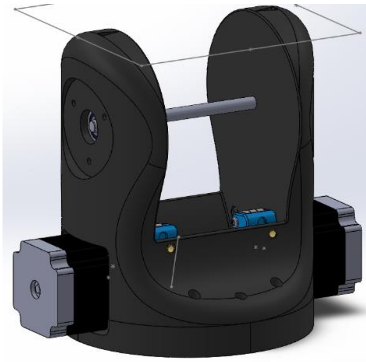


Figure 2. Design of the second axis.

3) Third Axis

The rotation range of the third axis is ±108°, and the zigzag part at the lower right is the belt fixing area. The belt is driven by the second shaft motor synchronous wheel to drive the third shaft to rotate. From Figure 3, it can be seen a stepping motor and an inelastic coupling. The coupling and the inelastic coupling are the connection points between the third shaft and the fourth shaft, and the rotation of the motor drives the fourth shaft to rotate.

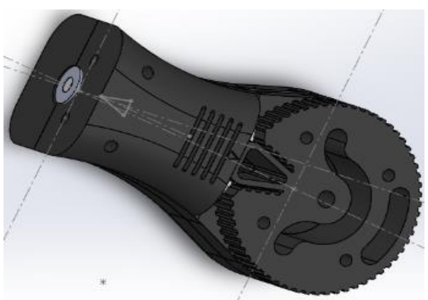


Figure 3. Design of the third axis.

4) Fourth Axis

The rotation range of the fourth shaft is ±180°. Figure 4 shows that the motor is fixed to the right side of the fourth shaft joint, and the motor shaft end is locked with a 14-tooth T5 tooth synchronous wheel; in the meantime, the belt tool is fixed to the synchronous wheel to tighten the belt on one side so that the fourth and fifth axis joints will not cause physical drooping. The upper side of the synchronous wheel is the optical axis, fixing the axis position of the fifth axis so that the fifth axis will not deviate. A bearing with the outer diameter 22mm and inner diameter 8mm is placed on the same side of the optical axis to avoid wearing-out situation.

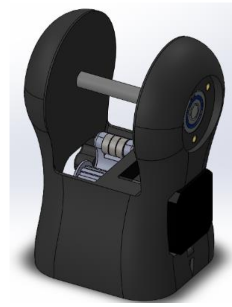


Figure 4. Design of the fourth axis.

5) Fifth Axis

The rotation range of the fifth axis is ±97°, as shown in Figure 5. The outer side of the fifth axis is the fixed place to fix the T5 toothed belt. At the center of this axis is the position of the optical axis of the fourth axis. However, there is a lock at the center of the front of this axis, which is the fixed place of the gripper.

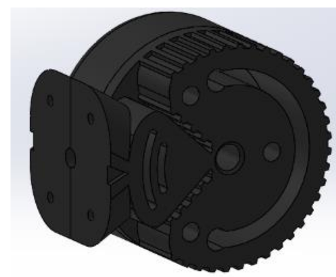


Figure 5. Design of the fifth axis.

6) Gripper

The clamping jaw is divided into two parts. The first part is the body in Figure 6 and the arm is printed using 3D printing technology. The second part is to increase the clamping force. The end surface of the gripper is manufactured by casting. The material used is epoxy resin with hardener, and its blending ratio is 100:1, which is the white part in the figure.

Figure 6. The designed gripper.

7) 3D Printing Density Experiment

Repetier-Host, a comprehensive 3D printing software, is used in this paper, which can perform slicing, viewing and modifying G-Code, manually controlling 3D printer, or modifying certain firmware parameters on the printer, etc. Considering the robot arm weight problem, the shell is tested from a density of 50%, 30%, to the final 20%. The results are shown in Table 1, where 20% density is the most suitable for robot arm and its production because the problem of the arm not being able to lift or drop itself will not happen. The final assembled robot arm with gripper can be seen in Figure 7.

Table 1. 3D printing test results

		Density		
		50%	30%	20%
Joint	Joint1	V	V	V
	Joint2	X	V	V
	Joint3	X	X	V
	Joint4	V	V	V
	Joint5	V	V	V

V: suitable; X: not suitable

Figure 7. The final assembled robot arm

B. Simulation in Robot Operation System

In this study MoveIt, an easy-to-use open source robotics manipulation platform, is applied. Specifically, the MoveIt Setup Assistant can be used as a simulated arm. Its function is to convert the robot model into a custom robot description format file, and generate other necessary related configuration files for an easier use in the MoveIt interface.

The configuration file mentioned in MoveIt requires a URDF (Unified Robot Description Format) file as the main file, which needs to be exported through the SolidWorks plug-in URDFconverter (solidworks_urdf_exporter) as a medium. After exporting, a ROS special folder will be generated, and its generated content includes Arm URDF narrative files, meshes, textures and other file directories.

Kinematics and Dynamics Library (KDL) is the default kinematics plugin in MoveIt, which can be configured when using MoveIt Setup Assistant for model configuration. The features are as follows:

- Type: numerical solution.
- Advantage: can solve inverse kinematics in closed state.
- Disadvantage: slow speed, may not find the solution.

KDL regards the problem of robot inverse solution as a multivariate nonlinear equation (1), which is solved by using the Newton-Raphson method. Here fk_m shows there are m degrees of freedom, and q_n means that there are n joints. If n> m, it means kinematic redundancy. p_{goal} means the target point at the end of the arm

$$f = \begin{bmatrix} fk_1(q_1 \dots q_n) & - & p_{goal1} \\ \vdots & \vdots & \vdots \\ fk_m(q_1 \dots q_n) & - & p_{goalm} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (1)$$

The Jacobian matrix is given by

$$f' = J = \begin{bmatrix} \frac{\partial fk_1}{\partial q_1} & \cdots & \frac{\partial fk_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial fk_m}{\partial q_1} & \cdots & \frac{\partial fk_m}{\partial q_n} \end{bmatrix} \quad (2)$$

Applying the Newton-Raphson method to solve the joint position at the next iteration, one obtains

$$\begin{bmatrix} q_1(k+1) \\ \vdots \\ q_n(k+1) \end{bmatrix} = \begin{bmatrix} q_1(k) \\ \vdots \\ q_n(k) \end{bmatrix} + J^{-1} \begin{bmatrix} Err_1 \\ \vdots \\ Err_m \end{bmatrix} \quad (3)$$

where $Err = f$ represents the current error

Combining the linear velocity v and angular velocity w of the end of the robot into a vector, the relationship between the velocity and the mechanical joint can be expressed as follows

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} j_{1,1} & \cdots & j_{1,n} \\ \vdots & \ddots & \vdots \\ j_{m,1} & \cdots & j_{m,n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = \dot{q}_1 \begin{bmatrix} j_{1,1} \\ \vdots \\ j_{m,1} \end{bmatrix} + \cdots + \dot{q}_n \begin{bmatrix} j_{1,n} \\ \vdots \\ j_{m,n} \end{bmatrix} \quad (5)$$

Each column at the right hand side in (5) represents the end linear velocity and angular velocity components each joint produces.

Next, the ROS structure used for the robot arm is briefly discussed. The main ROS structure consists of move_urdf, move_moveit_congif, and move_moveit as the three major items. In addition, the rosserial node for Arduino communication is established. They are introduced separately as follows:

1) move_urdf:

As mentioned above, it contains all the description files of the robotic arm, and is configured as the necessary files for simulating the arm in RViz and MoveIt.

2) move_moveit_config:

Configuration for moveit, a motion planning framework that has a plugin in RViz, which is what we use here.

3) move_moveit:

a. moveit_convert.cpp: will simulate the rotation of "joint_state"(convert from "move_group/fake_controller_joint_states" theme to steps) and publish on the /joint_steps theme. Joint_steps is an array of 6 Int16 values (although we only have 5 joints in this case) representing the cumulative steps performed by each joint since the moveit_convert node started running.

b. move_group_interface_coor_1.cpp: You can hard-code the posture/position of the end effector in the script and plan/execute the trajectory there. The current posture/position of the end effector is also readable and can be output to control the robot arm.

4) Arduino rosserial:

Arduino is a plug-in software for the ROS system. Therefore, it is needed to create a ROS library in the Adruino IDE and write the node information to subscribe to ROS to properly integrate with the ROS system.

III. Experimental testing And validation

A. Robot Arm Accuracy Testing

This experiment is to test the accuracy of the three-axis direction movements of the robot arm. Because the robot arm is 3D printed with the PLA material, the dimension can be slightly deviated. Through the accuracy experiments, the displacement error between the ideal (yellow) and the actual (blue) moving can be obtained, in which 20 mm displacement in X, Y, Z axis is assumed for 20 number of times of testing. The measurement resolution is 0.01mm. The following plots in Figures 8, 9, and 10 are the experiment results.

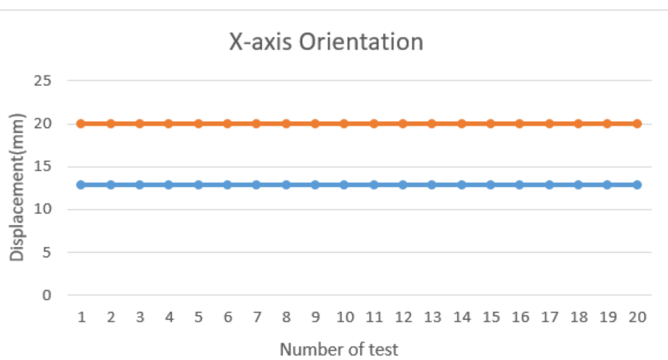


Figure 8. X-axis result (deviation: 7.15mm; repeatability: ±0.02mm)

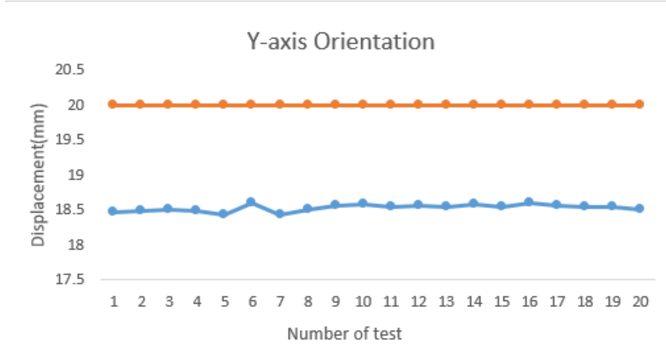


Figure 9. Y-axis result (deviation: 1.47mm; repeatability: ±0.085mm)

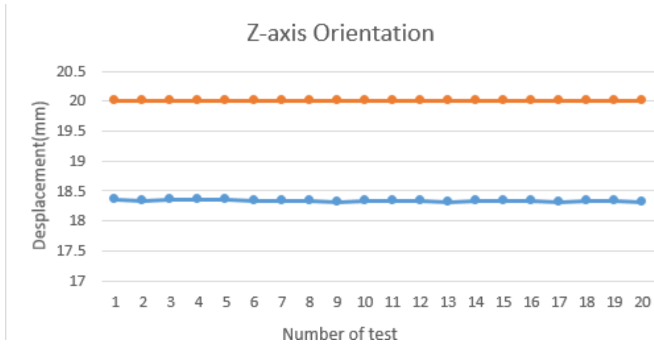


Figure 10. Z-axis result (deviation: 1.67mm; repeatability: ±0.015mm)

B. Path Planning and Real-time Dynamic Testing

In this experiment, the RViz visual interface is used to observe the planned path of the arm URDF model, to achieve the synchronized movement of the arm and to observe the target position and the reaction process. The experimental process is as follows:

1) RViz displays the current state of the arm in Figure 11.

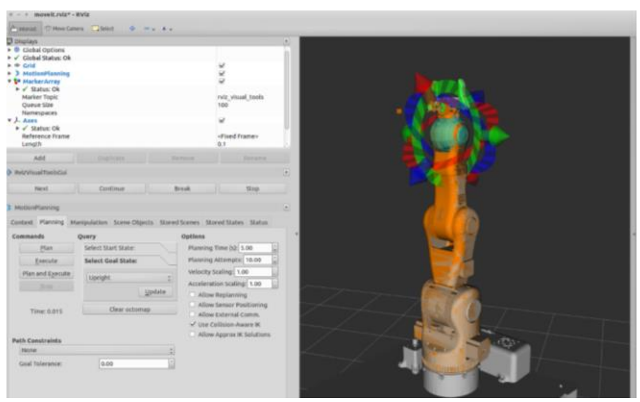


Figure 11. Current state of the robot arm.

2) Selection of the path planning, which is divided into custom path planning, random path planning, and mouse-drawn target position method, etc. In Figure 12, it is dynamically displayed after planning.

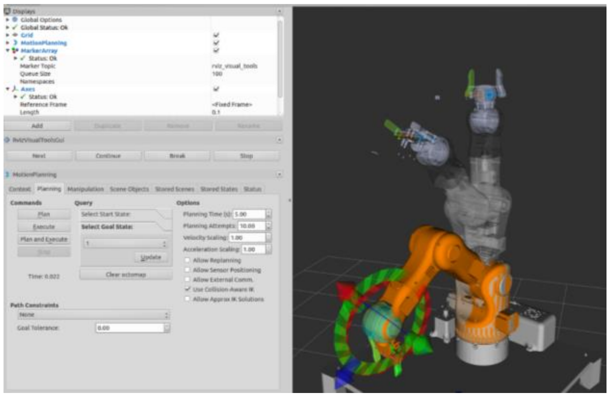


Figure 12. Path planning mode.

3) After confirming that the path planning is correct, one can execute the actual path movement for the robot arm. Figure 13 shows the actual moving of the robot arm from the start point to the target point.

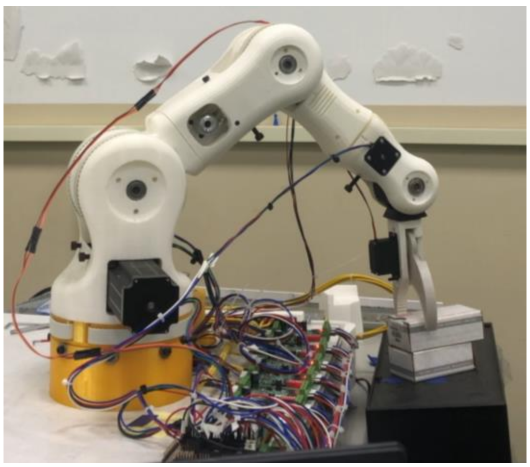


Figure 13. Real time moving of the robot arm.

C. Object Recognition and Sorting

In this experiment, Python 3, Tensorflow's Object Detection API, OpenCV, and a webcam are integrated to the robot arm for object recognition and sorting task. Two types of objects "cup" and "bowl" are chosen as the targets. The flow chart is shown in Figure 14 and the corresponding hardware configuration for the experiment is illustrated in Figure 15. An object is placed under the camera in front of the robot arm. The camera captures the image and the object is detected by Tensorflow's Object Detection API. Then the robot arm is controlled from its initial upright position to grasp the object and place it to the designated location. Experimental results are shown in Table 2, where 4 different cups and 3 different bowls are used. Successful rate for cup recognition and sorting is 100%, while that for bowl is 90.67%.

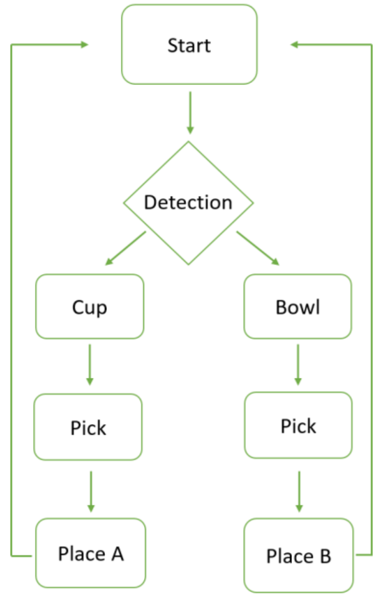


Figure 14. Flow chart for object recognition and sorting task by robot arm.

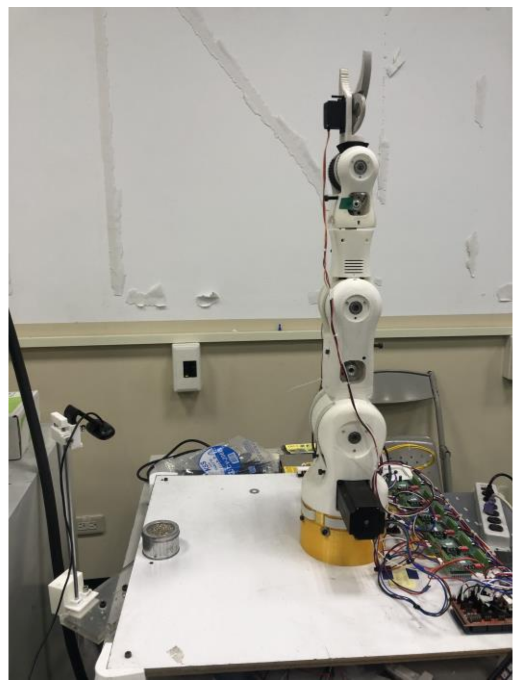
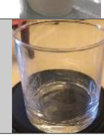


Figure 15. Experimental hardware configuration.

Table 2. Experimental results

Item	**Successful**	**Failed**	**Successful rate**
	100	0	100%
	100	0	100%
	100	0	100%

	100	0	100%
	90	10	90%
	86	14	86%
	96	4	96%

- Cup successful rate:100%
- Bowl successful rate:90.67%

IV CONCLUSION

A. Conclusion

In this study, a ROS-based robot arm with five-axis is designed and manufactured using 3D printing technology. By using RViz and MoveIt, robot arm path planning simulation and real-time arm actuation are feasible. In addition, deep learning in Tensorflow with a vision system is integrated to the robot arm for intelligent object detection and sorting task. Summaries of the study results are as follows:

1) Using the ROS robot system as the operating platform has indeed greatly improved the communication stability between the hardware and software, ex: node to node structure, Language-independent interface definition language.

2) MoveIt provides faster arm parameter establishment, adds more convenience, and reduces the danger of the arm running path.

3) Improve the efficiency of arm development through the ROS client community, for example: the communication software package between the motor, Arduino and ROS.

4) Compared with other commercially available 3D printing robots, the developed robot is low-cost with reliable repeatability and can expand its functions easily under the ROS architecture.

5) Through Tensorflow as a visual recognition method, the success rate of distinguishing "cup" and "bowl" and placing them in the correct position is 100% and 90.67% respectively.

However, this study also has problems that need improvements:

1) The arm is made of plastic material (PLA). Compared with the commercially available metal material for robotic arms, the PLA body is weaker, and some parts need to be renewed after a long time of use.

2) When the motor is running, the second shaft is subjected to greater gravity, the noise and the operating temperature are greater and higher at full speed.

3) Due to high motor load and high operating temperature, the motor drive board may occasionally crash.

B. Future Work

After summarizing the results and problems in the previous section, the following suggestions are proposed for this research as future work:

1) Due to gravity in the mechanism, changing the length of the arm joint can reduce the load capacity and rotational inertia of the second-axis motor, and in the future, the position distribution between the motor and the mechanism or the combination of sound-absorbing materials can be adjusted to reduce noise.

2) As for the material of the arm, the plastic material (PLA) used in this study will increase the accumulated tolerance. In the future, one can consider using 3D metal printing to increase the strength of the arm body, and the accuracy of the body will be relatively more accurate.

3) In the case of the overheated driver board, it is necessary to design the case and install a fan in the future to establish a good working environment and the overall visual appearance can be more beautiful.

4) Add sensors, such as pressure sensor, torque sensor, ultrasonic sensor, etc., to increase user safety for human-robot collaboration.

5) ROS is mainly implemented in navigation control of autonomous mobile robots (AMR). In the near future, ROS-based robot manipulator will be installed in AMR to form a mobile robot manipulator for more versatile service works.

REFERENCES

[1]. Rich. Blake, Industrial Robotic Arms Race Leaves Experts Asking 'What Manufacturing Slowdown', Forbes, Jan, 2019.
[2]. Industrial robot sales increase worldwide by 31 percent, IFR, Sep, 2019.
[3]. What Are The Main Types Of Robots, RobotWorx, Marion, OH.
[4]. Dr. Adrian. Bowyer, The Official History of the RepRap Project, All3DP, Aug, 2016.
[5]. 3D Printing Technologies: An Overview, TechPats Blog,
[6]. Akshay. Kumar, This article shows how to create a ROS package for a simple 4-DOF robot arm and simulate a predefined joint path in Rviz,, Maler Pro, Mar, 2020.
[7]. DmitryM8, ROS MoveIt Robotics Arm, Instructables circuits, Aug, 2018.
[8]. Vinay. Lanka, Control any Robotics Arm with ROS, Medium, May, 2020

Jin-Siang Shaw received the bachelor degree in 1981 from Taipei Institute of Technology in Taiwan, and Master and PhD degrees in 1985 and 1989 from Clemson University and Michigan State University, respectively, in USA. All degrees were in the field of Mechanical Engineering. He is currently a Professor with the Mechanical Engineering Department, National Taipei University of Technology, Taiwan. His current research interests include intelligent robot, artificial intelligence, and biomedical engineering.

Yen-Ju Lin received the bachelor degree in Mechanical Engineering Department from Hua Fan University, Taiwan, in 2014, and is currently pursuing master degree in Institute of Mechatronic Engineering from National Taipei University of Technology. His current research interests is in robotics and 3D printing technology.