

Robot Arm System Based on Augmented Reality Approach

Muhammad Aqif Izzat Mazlan, Mohd Hairi Mohd Zaman, Muhammad Fikri Nasir, Mohd Faisal Ibrahim, Asraf Mohamed Moubark

Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
aqifizzat@gmail.com, hairizaman@ukm.edu.my, mfn2704@gmail.com, faisal.ibrahim@ukm.edu.my, asrafmohamed@ukm.edu.my

ABSTRACT

Augmented reality (AR) is a form of information and communication technology that provides virtual visual information (i.e., images, animation, or video) augmented upon physical reality through computer devices. This study aims to use AR technology on a robot arm system and act as feedback data for this system. The AR technology is applied to an AR marker (i.e., ARTag marker), and ARTag data are acquired using 3D or depth cameras. This robot arm system operates to retrieve target objects located in the scanning area scanned by cameras. After the target object has been identified, the robot arm moves the target object to the target location. The system works by combining the robot arm control and AR algorithms. The control algorithm is used for controlling the robot arm movement, whereas the AR algorithm is for identifying ARTag markers. Therefore, this study focuses on building an efficient AR algorithm to identify the target and non-target objects and test this algorithm on an actual robot arm. Results indicate that the robot arm can capture the target object successfully, and the camera can accurately read the coordinates of the ARTag marker located on the target object. Therefore, this research successfully developed an AR-based robot arm system.

Key words : Augmented reality, ARTag marker, robot arm, 3D, depth camera.

1. INTRODUCTION

The emergence of the industrial 4.0 era has resulted in the acceleration of technological development, which involves the continuous evolution of technology. Therefore, the development of advanced technology has become one of the essential aspects to generate the national economy, maintain security, and make tasks easier. Amongst the technologies that are gaining ground in this modern era is digital image processing. Image processing is a form of signal processing, in which the input is an image, such as a picture or video frame, and the image processing output may be either an image or a video frame or a set of features or parameters related to an image using computer algorithms. This process is

done to improve the quality of images by modifying or analyzing them. Digital image acquisition typically results in unwanted camera errors and is caused by random and unstable camera movements. Therefore, image algorithms are needed to improve picture quality and eliminate unwanted camera blur. This technology is widely used to perform tasks for humans. The use of this technology is also evident in the fields of security, industry, and education. For example, this camera technology is implanted into robot arms in the industry to be able to identify objects in the surrounding area [1]–[6].

Robot arms that use this image processing application can facilitate tasks in which robots can identify objects that should be identified. An example is the ability to identify objects in a scattered place and damaged objects [7]–[9]. This system is beneficial in the industrial sector, thereby enhancing the productivity of product manufacturing and the quality of products. This situation is in contrast to using the human workforce, which has a high probability of making mistakes [10]–[12].

Therefore, this study focuses on the problem in which a robot arm is unable to identify whether the target object is correct or not. The reason is that the majority of the current robotic arms do not use the cameras as feedback sensors in the control system. The augmented reality (AR) method is used to address this issue and provides the feedback data for the robot arm through AR marker usage [13]. There are many types of AR markers, such as AprilTag, ARTag, and CALTag [14]–[17].

Accordingly, this study has three objectives; firstly, to develop an AR algorithm for a robot arm system, then to build an AR algorithm that can distinguish between target and non-target objects, and finally, to verify the performance of the algorithms developed using a real robot arm. The outline of this paper is based on these objectives.

2. METHODOLOGY

2.1 Hardware Architecture

The system design in this study consists of a 4 degrees-of-freedom (DOF) robot arm (model

OpenManipulator from Robotis) [18], a serial communication converter (U2D2 from Robotis), a depth camera (RealSense D435i from Intel) [4], [5], [19], and a computer with Ubuntu OS and Robot Operating System (ROS) installed [20], [21]. This system was created to identify objects using the ARTag marker as the ID for the object. Figure 1 shows the hardware used in this study. Thereafter, a performance test was conducted to the robot arm and camera to determine the error of the equipment.

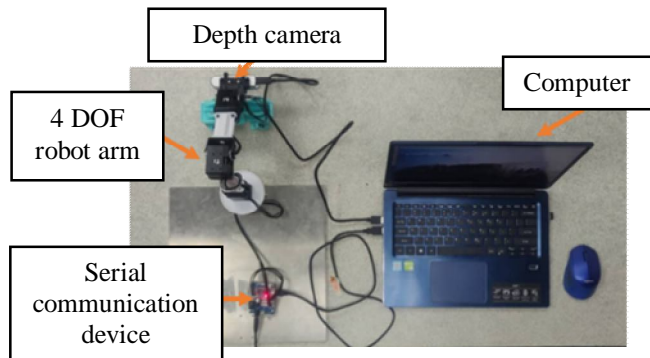


Figure 1: Hardware Setup

The performance test of the robot arm requires several steps. Amongst these steps is to identify the ID and check the servo motor settings found on the robot arm using the DYNAMIXEL WIZARD 2.0 application, as depicted in Figure 2, and allow the servo motor to interact with the ROS software. After that, the control program GUI shown in Figure 3 was utilized to test the robot arm movement.



Figure 2: Dynamixel Wizard 2.0

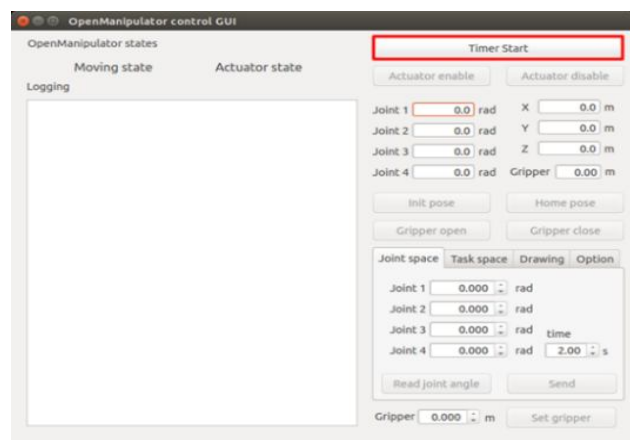


Figure 3: Control Program GUI

To test the performance of the RealSense D435i camera, we used the software provided by Intel or the open-source Rviz software. Accordingly, we can determine if the camera is functioning well or not. If it does not remove the image from the software, then it indicates that the camera or camera setup is faulty or damaged.

After testing was completed, the hardware and computer were integrated into the ROS framework together with the AR function. Using Ubuntu OS and ROS, the robot arm system was programmed (using C++) to identify the ARTag markers. After building the algorithm, it was included in ROS and became a node for the ROS system.

After that, ARTag recognition was evaluated using the 3D visualization tool Rviz, which is available in ROS. The ARTag marker is attached to a rectangular object, as displayed in Figure 4. Then, the camera, installed on top of the robot arm's end effector, scanned its scanning area thereafter to detect the ARTag marker, obtain the data of the detected ARTag marker and extract the ARTag information on Rviz. The information was displayed on Rviz in the form of a 3D ARTag, as shown in Figure 5.

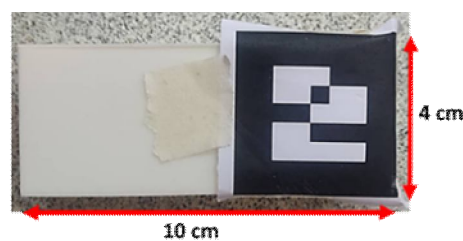


Figure 4: Target Object with ARTag Marker

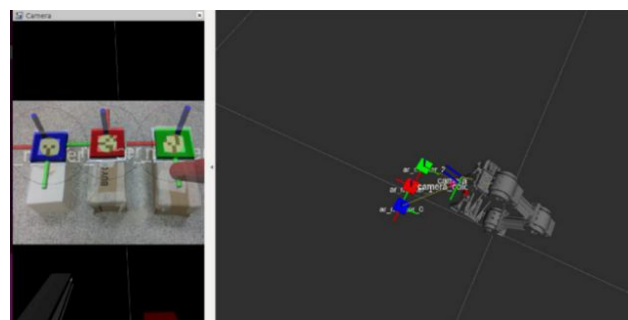


Figure 5: Example of Information Appearing in Rviz

After completing all the necessary tests for this system, the next step is to execute the primary system for the proposed robot arm. Figure 6 shows a brief flowchart of the system operation. Several steps are needed to execute the robot arm system. The first step is to run the 'roscore' command for the ROS node to communicate with the system. The 'roscore' command is a collection of nodes and programs that are prerequisites of the ROS-based system.

The second step is to run the control instructions for this system. These instructions detect each joint in the robot arm and allow the robot arm joint to communicate with the nodes

run by the computer. The third step is related to the camera, in which Rviz can produce 3D visual data captured through the camera and robot arm joints in real-time.

The last step is to run the main program for this system. The system implemented can provide ARTag coordination information. Given this coordination information, the system converts the coordination into orientation values and assigns these values to the robot arm joint for the robot arm to take the ARTag. The system has three commands. The first command is the initial position when pressing the number '1' key. The second command is to start the operation, in which the system starts to operate and detect the ARTag marker. The system holds the object when ARTag is identified and repeat the operation until ARTag is not found before exiting the system. This operation is performed using the number '2' key. The third command is to stop the operation as soon as the number '3' key is pressed and return to the initial position. The difference between the first and third commands is that the former will stop operating entirely and the latter will stop temporarily and operate again on the stopped data. For example, if the robot arm stops at the fourth ARTag, then it will operate with the fifth ARTag data when the operation is resumed.

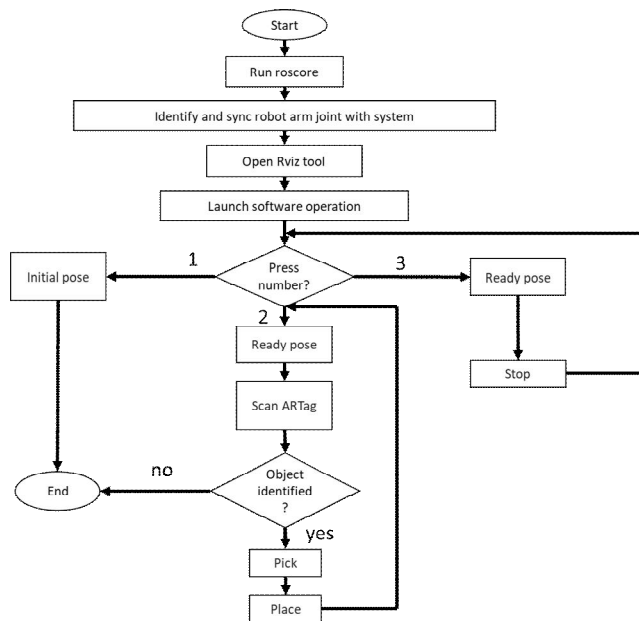


Figure 6: Operating Flowchart

2.2 Software Architecture

For the algorithm architecture, this study uses two separate algorithms. The first algorithm is the AR algorithm, and the second is an operational or control algorithm. Both algorithms are integrated to become the overall algorithm that will enable this research to achieve its objectives.

The AR algorithm is used for identifying the AR markers, namely, ARTag. This algorithm is also used to obtain the data

available on ARTag. The block diagram of the AR algorithm operation is shown in Figure 7.

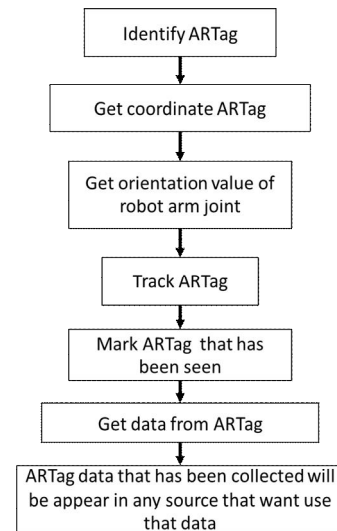


Figure 7: Block Diagram of the AR Algorithm

The operational algorithm is for controlling the robot arm movement based on the identified AR markers. Figure 8 shows the step-by-step operation to use this algorithm and to understand how this algorithm works.

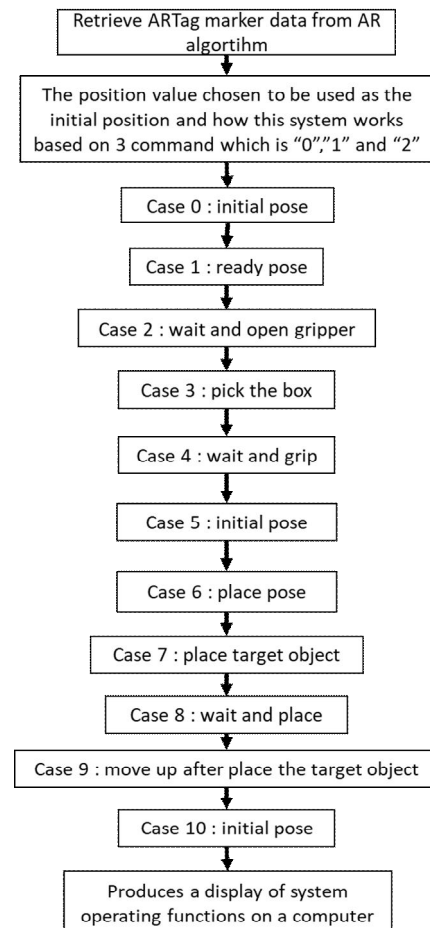


Figure 8: Block Diagram of the Operation Algorithm

3. RESULTS AND DISCUSSION

3.1 System Operation Results

The operation of the robot arm system is successfully implemented and meets the requirements of the current study's objectives. Figure 9 shows how the actual robot arm moves step-by-step. Figure 10 shows the real-time robot arm movement using Rviz. The diagrams in Figures 9 and 10 show that the robot arm operates after pressing the number '2'. Next, the robot arm moves to the starting position and proceeds to the standby position, where it starts scanning the area and identifying the ARTag marker. The robot arm eventually moves the target object to the target location. This process is repeated until its scanning area no longer has no ARTag.

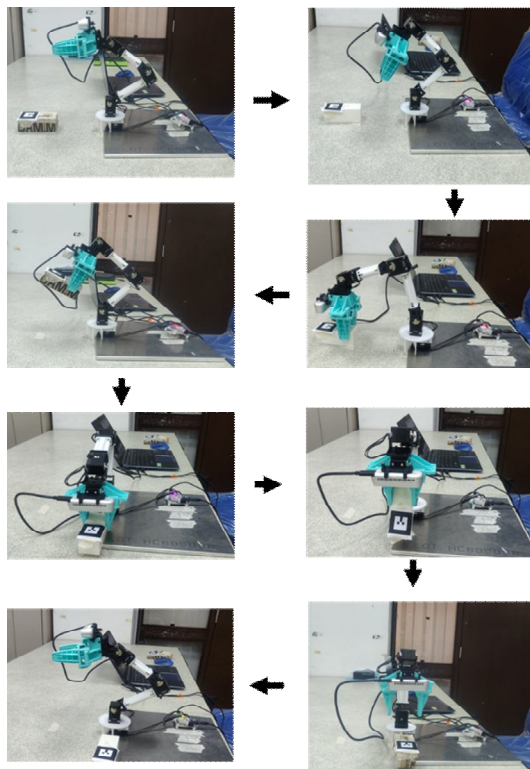


Figure 9: Robot Arm Operation Sequence

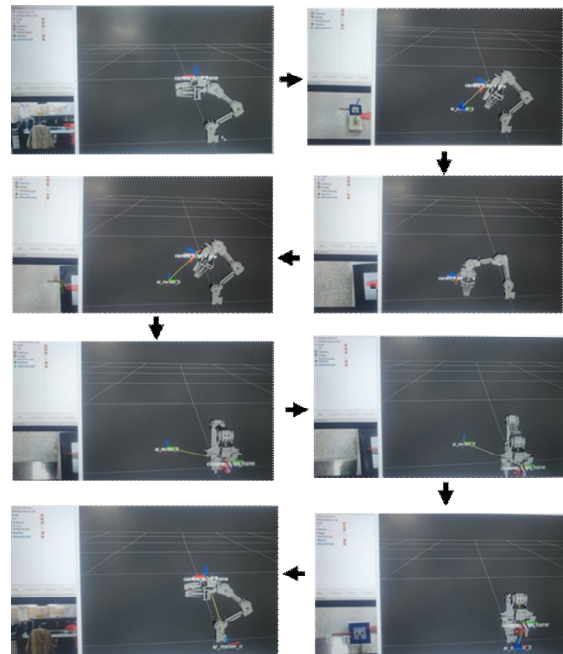


Figure 10: Robot Arm Operation in Rviz

3.2 ARTag Results

Figure 11 shows the robot arm scanning the ARTag marker. After that, proposed the system acquires information using Rviz, in which the ARTag marker ID from left to right in Figure 11 are *ar_marker_0*, *ar_marker_1*, and *ar_marker_2*.

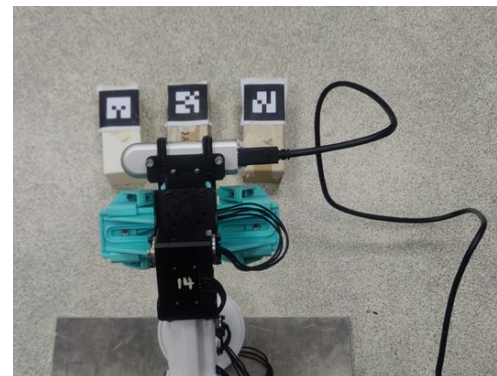


Figure 11: Robot Arm Scanning the Target Object

Figure 12 shows the data released by the camera during the when camera scanning process through the Rviz tool, where blue represents ID: *marker_ar_0*, red represents ID: *marker_ar_1*, and the green represents ID: *marker_ar_3*, respectively.

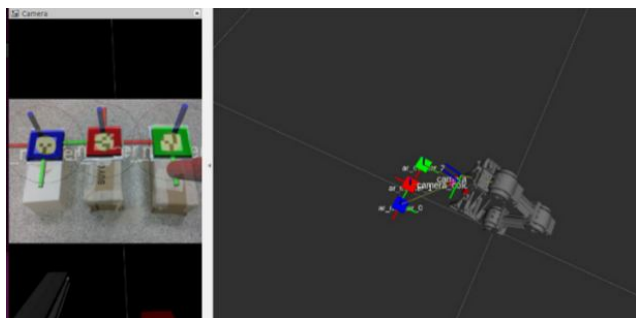


Figure 12: Data Display in Rviz

The robot arm operating algorithm has a display function to obtain the coordination information of the ARTag marker (see Figure 13). The ‘rostopic’ command was executed in ROS to obtain the joint orientation values. The ‘rostopic’ command is a type of ROS command that extracts data based on the required information, specifically by typing ‘rostopic echo <desired data from hardware>’. Figures 13 to 16 present the data for each identified ARTag through the camera. These diagrams represent each ID from 0 to 2. Table 1 lists all information related to the ARTag coordination data and robot arm orientation values after executing the ‘rostopic’ command in ROS.

Table 1: ARTag Coordination Data and Robot Arm Orientation Values

Marker s	Coordinates (x, y, z)	Orientations (J1, J2, J3, J4)
0	(0.217,0.019,0.108)	(-0.026, -0.046,0.713,0.699)
1	(0.211, -0.037,0.113)	(-0.024, -0.044, -0.011,0.999)
2	(0.215,0.072,0.107)	(-0.037, -0.008,0.686,0.727)

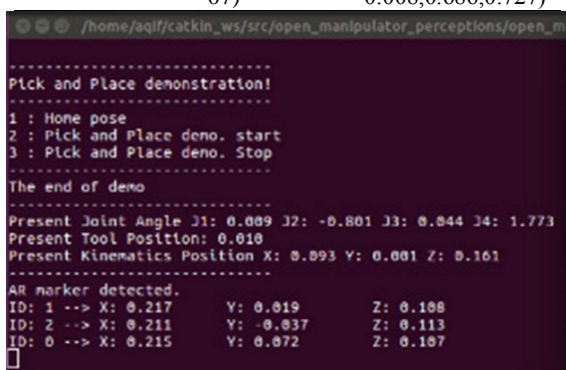


Figure 13: Acquired AR Marker Data in Computer

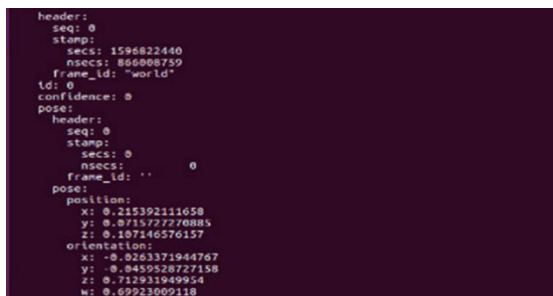


Figure 14: Data for ARTag ID: 0

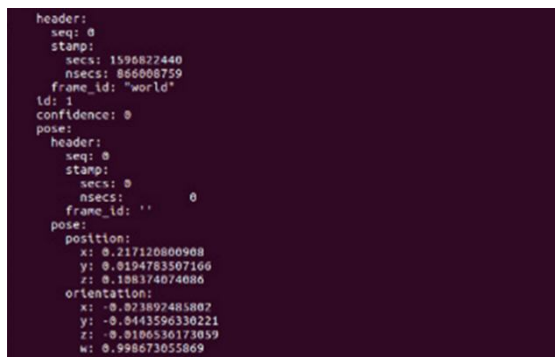


Figure 15: Data for ARTag ID: 1

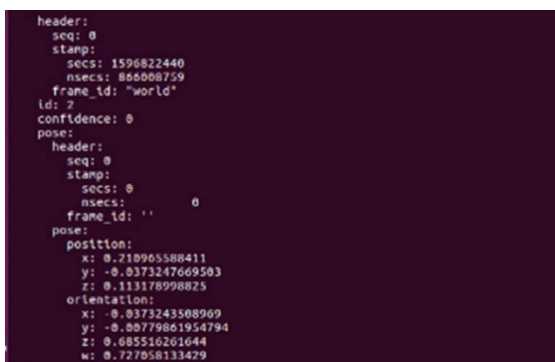


Figure 16: Data for ARTag ID: 2

ARtag was selected as the feedback data of the robot arm system in this study. Amongst the advantages of this ARTag compared with other markers is that the former has a considerably lower false positive, and the error rate and confusion between markers are also shallow. Additionally, ARTag does not need to load or create patterns because it has its own unique ID marker library. The performance of this system can compete with other object identification systems because the cost of making the marker is affordable, and its algorithm is simpler to be developed.

4. CONCLUSION

This study successfully developed a robot arm system based on the AR approach, particularly by using a 4-DOF Robotis OpenManipulator robot arm platform and Intel Realsense D435i depth camera. The proposed system can also identify the correct target objects using ARTag. The AR algorithm to detect ARTag markers printed on target objects was successfully executed. Furthermore, the performance of the developed algorithm was evaluated on an existing low-cost robot arm system.

ACKNOWLEDGEMENT

This work was financially supported by Universiti Kebangsaan Malaysia (grant no. GGPM-2019-051).

REFERENCES

- [1] C. L. Shih and Y. Lee, **A simple robotic eye-in-hand camera positioning and alignment control method based on parallelogram features**, *Robotics*, vol. 7, no. 2, pp. 1–14, 2018.
- [2] Y. Chen, G. Sun, H. Lin, and S. Chen, **Random bin picking with multi-view image acquisition and CAD-based pose estimation**, in *Proc. 2018 IEEE International Conference on Systems, Man, and Cybernetics*, 2018, pp. 2218–2223.
- [3] C. Kulpate, R. Paranjape, and M. Mehrandezh, **Precise 3D positioning of a robotic arm using a single camera and a flat mirror**, *Int. J. Optomechatronics*, vol. 2, no. 3, pp. 205–232, 2008.
- [4] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed. Springer, 2017.
- [5] B. Mathur, A. Topiwala, H. Saeidi, T. Fleiter, and A. Krieger, **Evaluation of control strategies for a tele-manipulated robotic system for remote trauma assessment**, in *Proc. 2019 SIAM Conference on Control and Its Applications*, 2019, pp. 7–14.
- [6] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, **Training deep neural networks for visual servoing**, in *Proc. 2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3307–3314.
- [7] F. Chaumette and S. Hutchinson, **Visual servo control. I. Basic approaches**, *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.
- [8] F. Chaumette and S. Hutchinson, **Visual servo control. II. Advanced approaches [Tutorial]**, *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, 2007.
- [9] F. Chaumette, **Robot visual control**, in *Encyclopedia of Systems and Control, 2nd edition*, J. Baillieul and T. Samad, Eds. Springer London, 2020, pp. 1–18.
- [10] M. H. M. Zaman, M. F. Ibrahim, N. Zainal, and M. F. Nasir, **Dual-arm robot with mobile robot platform with master-slave configuration for teleoperation application**, *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1.4 Special Issue, pp. 685–688, 2020.
- [11] O. O. Obe, E. A. Ajayi, and O. O. Odewale, **Genetic algorithm based optimal trajectories planning for robot manipulators on assigned paths**, *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 8, pp. 4888–4893, 2020.
- [12] H. Kevin, H. Tjahyadi, A. Aribowo, and A. S. Putra, **Designing the label giving robot arm in the packing box**, *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 8, pp. 4867–4874, 2020.
- [13] H. Kato and M. Billinghamurst, **Marker tracking and HMD calibration for a video-based augmented reality conferencing system**, in *Proc. 2nd IEEE and ACM International Workshop on Augmented Reality*, 1999, pp. 85–94.
- [14] E. Olson, **AprilTag: A robust and flexible visual fiducial system**, in *Proc. 2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.
- [15] M. Fiala, **ARTag, a fiducial marker system using digital techniques**, in *Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 590–596.
- [16] A. Sagitov, K. Shabalina, L. Sabirova, H. Li, and E. Magid, **ARTag, AprilTag and CALTag fiducial marker systems: Comparison in a presence of partial marker occlusion and rotation**, in *Proc. 14th Int. Conf. Informatics Control. Autom. Robot.*, 2017, pp. 182–191.
- [17] B. Atcheson, F. Heide, and W. Heidrich, **CALTag: High precision fiducial markers for camera calibration**, in *Proc. Vision, Modeling, and Visualization*, 2010, pp. 41–48.
- [18] Robotis, **OpenMANIPULATOR-X**, 2020. [Online]. Available: https://emanual.robotis.com/docs/en/platform/openmanipulator_x/overview/. [Accessed: 28-Sep-2020].
- [19] Y. Pyo, H. Cho, J. Ryuwoon, and T. Lim, *ROS robot programming: From the basic concept to practical programming and robot application*, Seoul: Robotis, 2017.
- [20] L. Garber, **Robot OS: A new day for robot design**, *Computer.*, vol. 46, no. 12, pp. 16–20, 2013.
- [21] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, **ROS: an open-source Robot Operating System**, in *proc. Open-Source Software Workshop of the Int. Conf. on Robotics and Automation*, 2009, pp. 679–686.