Proceedings of the Canadian Society for Mechanical Engineering International Congress 2022
CSME Congress 2022
June 5-8, 2022, Edmonton, AB, Canada

Vision-based Impedance Control of a 7-DOF Robotic Manipulator for Pick-and-Place Tasks in Grasping Fruits

Qiguang Chen[1], Lucas Wan[1], Prabahar Ravichadran[1], Ya-Jun Pan[1*], Young Ki Chang[2]
[1]Department of Mechanical Engineering, Dalhousie University, Halifax, Canada
[2]Department of Engineering, Faculty of Agriculture, Dalhousie University, Halifax, Canada
*yajun.pan@dal.ca

Abstract— A vision-based impedance control method is applied to a 7-degree-of-freedom (7-DOF) Franka Emika (FE) Panda robotic manipulator to complete pick-and-place tasks with human-like grasping of fruits. The interfaces between different hardware and the controller are developed using robotic operating system (ROS). You-Only-Look-Once (YOLOv5) object detection algorithm is trained for fruits and employed for capturing the location of the targeted objects. A qb-SoftHand robotic hand is used as the end-effector for the grasping tasks. By integrating these components together through ROS, the FE Panda robot successfully achieves autonomous human-like pick-and-place tasks as shown in the experimental results.

Keywords--7-DOF robotic manipulator; impedance control; pick-and-place task; ROS; YOLOv5 object detection.

I. INTRODUCTION

As the application of robotic manipulators increases in manufacturing, healthcare, and education, it is important for robots to look and perform in a human-like manner. In cases where humans and robots interact directly, having the robots perform like a human allows the human to be able to better predict the intent of the robot's actions and increases the trust between the human and the robot [1]. In applications such as automatic vehicle assembly and crop picking, the integration of human-like manipulators with different sensors and end-effectors is necessary to complete complex tasks. Recently, the interaction between manipulators and the surrounding environment using different end-effectors and sensors has become a popular research topic. In [2], a YOLOv3 based semi-automated pick-and-place task is completed by a 7-DOF robotic manipulator and an RGB-Depth camera. In [3], an object detection method based on the improved YOLOv5 object detection neural network is applied to manipulator grasping tasks. To achieve accurate control performance, different control theories are applied to the dynamic model of robotic manipulators. In [4], a novel adaptive robust integral Radial Basis Function Neural Network (RBFNN) impedance control method is applied to telerehabilitation with robotic exoskeletons, which shows the effectiveness of impedance control for tasks interacting with human. In [5], a gain scheduled sliding mode control (SMC) scheme is proposed for tracking control tasks of multilink robotic manipulators. Admittance control is applied to a multi-robot system to achieve the bilateral teleoperation of cooperative manipulators [6].

YOLO is an innovative approach to object detection where a single convolutional network predicts the bounding boxes and class probabilities of objects in an image at once [7]. This approach was comparatively faster than other object detection networks such as Faster R-CNN and R-FCN making it suitable for real-time applications. Since then, multiple versions have been released. The most recent release of the YOLO family is YOLOv5 which is written in the PyTorch framework compared to the darknet framework used by the earlier versions that were written in C and CUDA. With the PyTorch framework, YOLOv5 has become easier to implement.

In this paper, the YOLOv5 algorithm is applied with a low-cost web camera and integrated with a 7-degree-of-freedom (7-DOF) Franka Emika (FE) Panda robot to autonomously complete a fruit pick-and-place task in a human-like manner using impedance control. This work shows promising integration of vision module for real-time robotic applications.

II. HARDWARE SETUP

Fig. 1 shows the equipment set up for this paper in the Advanced Control and Mechatronics (ACM) Lab at Dalhousie

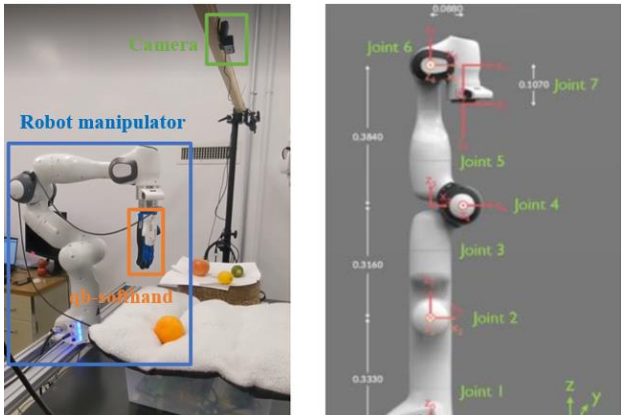

Figure 1. Equipment setup in the Advanced Control and Mechatronics (ACM) Lab at Dalhousie University

University. As shown in the figure, the 7-DOF FE Panda robotic manipulator, the qb-SoftHand end-effector, and the camera is a Logitech C922 HD webcam.

A. *Robotic Manipulator*

The FE Panda robot is a state-of-the-art 7-DOF robotic manipulator (www.franka.de/technology) suited for both research and industry. Its advantages are recognized on both the hardware side and the software side. In the aspect of hardware, 14-bit-resolution encoders and 13-bit-resolution torque sensors provide high-level accuracy and sensitivity. Brushless DC motors produce high movement execution speed and high control efficiency. The joint electronics provides 1kHz signal transmission frequency to ensure a smooth and stable signal exchange process. In the aspect of software support, the control interface is available through ROS packages and C++ API. The complete ecosystem makes the robot able to be easily integrated with other technology, enabling object detection and complex grasping tasks.

B. *qb-Softhand Setup*

A qb-SoftHand anthropomorphic robotic hand is employed as the end-effector on the FE Panda robot for fruit pick-and-place tasks. The qb-SoftHand closely resembles a human hand and has several advantages. First, the qb-SoftHand is compliant and safe to operate near humans. Second, the qb-SoftHand's five mechanically independent fingers allow this end-effector to have both a large gripping surface and excellent adaptability to grasp objects of different sizes and ductility. Third, the qb-SoftHand has 19 anthropomorphic DOFs and only one motor, allowing for simple control implementation.

C. *Camera*

The camera used for YOLOv5 object detection in this paper is the Logitech C922 webcam. The Logitech C922 has 1080p video resolution, 30 fps frame rate, 78° field of view, and is mounted directly above the operational space of the robot.

III. Methodology

A. *Dynamics of Franka Emika Panda Robot*

The dynamic model of a robot relates the position, velocity, and acceleration of the robot's joints and the force/torque exerted at each joint to generate motion. It is important to get the corresponding dynamic model of a nonlinear robotic arm system to achieve accurate control. The general form for a multi-DOF serial robotic manipulator can be written as

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + G(q) = \tau + \tau_f, \qquad (1)$$

where $M(q)$ is the symmetric and positive definite inertia matrix of the robot, $C(\dot{q}, q)$ is the Coriolis and Centrifugal torque matrix, $G(q)$ is the gravity torques matrix, τ is the control input torque, τ_f is the friction torque, and $q, \dot{q}$, and $\ddot{q}$ are the joint positions, joint velocities, and joint accelerations respectively. In this paper, for a 7-DOF robotic manipulator, the dynamic model of FE Panda robot is described by seven coupled and nonlinear differential equations. Therefore, the dimensions of $M(q)$ and $C(\dot{q}, q)$ are 7×7 and the dimension of $G(q)$ is 7×1. The dynamic matrices, M, C, and G are calculated internally by the FE Panda robot model library. The simulations used in the paper use the estimated dynamic parameter values as determined by [8].

B. *YOLOv5 Overview*

For the pick-and-place task, the YOLOv5 model was trained on custom images of fruits. Fig. 2 shows the series of tasks that was completed to train and optimize the YOLOv5 model to detect custom objects. The process starts with images acquisition where the images were acquired by placing the fruits (apples, oranges, lemons, limes, and pears) under the webcam held one meter above the table. A program was written in Python using OpenCV library to acquire the images with resolutions of 1437 × 1080. Fifty frames/images with fruits placed at different orientations and positions in the frame were obtained. Further, the fruits of different classes were placed at equal proportions in the frames to have a balanced dataset.

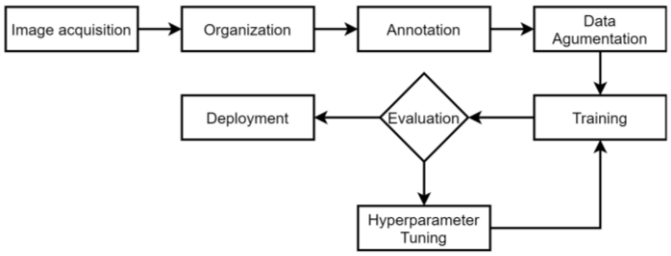

Figure 2. Block diagram for training YOLOv5 on custom objects

The training images were uploaded to Roboflow, an online platform to organize, annotate and augmentation images. The images were annotated by drawing a bounding box around the object and identifying its class. The annotated images were randomly split into training, validation, and test dataset. The data was further augmented by randomly flipping, rotating, and adjusting the brightness and hue of the images. Altogether, 110 images were used for training, 10 images were used for validation, and 5 images were as an independent test set.

The model was trained in a Google Colab environment where GPU resources were available to accelerate the training process. A code snippet from Roboflow was copied into the Colab environment to download the prepared dataset. There are multiple model configurations and architectures available for YOLOv5. However, the smallest and the fastest pre-trained YOLOv5 model was chosen for fine-tuning taking the implementation computer configuration into consideration.

The model was repeatedly trained by optimizing the some key hyperparameters such as image size, batch size, learning rate, etc., and evaluating the metrics such as mean Average Precision (mAP@0.5:0.95), precision and recall. The model was initially trained at an image size of 416 and gradually increased to 624, 832 and 1024 at a batch size of 16 for 300 epochs. An increase of mAP@0.5:0.95 was observed from 0.652 to 0.835 in detecting all classes when the image size was increased from 416 to 1024. An inference made on a sample

image from the test dataset is shown in Fig. 3. The trained weights were downloaded for deployment on to the application.

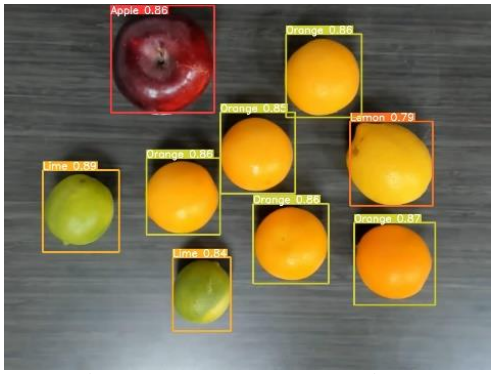

Figure 3. An inference made on a sample image from the test dataset

C. *Impedance Controller*

An impedance control is chosen as the control methodology in this framework because it is easy to implement, robust, and can achieve safe, compliant physical interaction with human users. The control torque is computed as

$$\tau = J^T\left(-K\Delta X - B(J\dot{q})\right) + C(\dot{q}, q)\dot{q}, \tag{2}$$

where

$$\Delta X = [\Delta\tilde{X}^T \quad \Delta\bar{X}^T]^T, \tag{3}$$

$$K = \begin{bmatrix} K_t & 0 \\ 0 & K_r \end{bmatrix}, \tag{4}$$

$$B = \begin{bmatrix} B_t & 0 \\ 0 & B_r \end{bmatrix}. \tag{5}$$

$\Delta\tilde{X}$ is a 3×1 vector containing the cartesian position error, $\Delta\bar{X}$ is a 3×1 vector containing cartesian orientation error, J is a 6×7 Jacobian matrix of the system, K_t and K_r are 3×3 diagonal matrices containing the translational and rotational impedance stiffness parameters and B_t and B_r are constructed similarly as the impedance damping parameters.

IV. Experimental Results

A. *Robot Operating System*

Robot Operating System (ROS) is a system architecture designed for interfacing with robot hardware and integrating different robotic hardware together. The function of ROS is like a universal plugin, which has the ability to connect different software and coding languages. ROS has built-in developing tools, such as Gazebo and Rviz, which provides a powerful simulation environment that allows developers to test their results before applying them to the real environment.

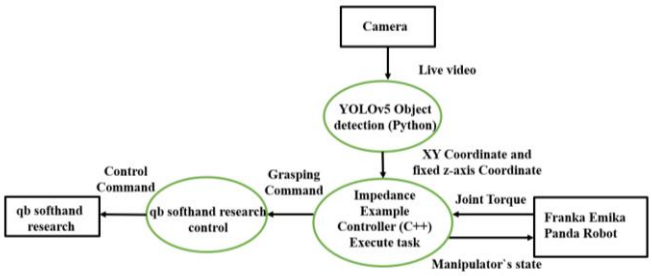

Figure 4. ROS block diagram of the experiment

Fig. 4 shows the ROS block diagram of the experimental hardware framework. Black square boxes represent the hardware and green ellipses represent the ROS nodes. First, live video is taken by the web camera and input to the YOLOv5 object detection algorithm. Next, the XY coordinates of desired target (apple and orange for the experiment in this paper) is calculated. The computer vision-based XY coordinates and a fixed Z coordinate is input to the impedance controller as the desired coordinates of the robotic manipulator`s end-effector. A set of actions are prescribed in the impedance example controller to perform the pick-and-place task which is shown in the following section. The desired joint torque is commanded to the robot and the manipulator's state is measured by encoders and torque sensors and sent back to the controller to complete the closed-loop control. On the other side, open and close commands are sent to the qb-SoftHand to allow the hand to execute the desired actions at the correct times.

B. *Task Execution*

The task execution is developed in C++ in sequence with the impedance controller provided by franka_ros. The flowchart for the pick-and-place task is provided in Fig. 5. The controller is executed at 30 Hz. For each subtask, the error between the current and desired position is calculated until the error reaches a specified threshold. Once the error has been calculated according to the appropriate subtask, the impedance controller is computed, and the control torque is applied to the robot.

Additionally, counters are implemented in each subtask to check that the error is within the error threshold for 500 loops. This is to avoid moving on to the next subtask without ensuring that the robot is locked on to the desired position in the case of overshooting or external interference. To avoid sudden motions while maintaining position accuracy, the position error is calculated incrementally as

$$\Delta\tilde{X} = \frac{C}{C_{max}}\left(\tilde{X} - \tilde{X}_d\right), \tag{6}$$

where $\tilde{X}$ is the current position, $\tilde{X}_d$ is the desired state, C is a counter, C_{max} is a maximum counter threshold. In this application, the path that the robot takes to the goal points is not prioritized and therefore the robot takes the path as generated by the impedance control torques. The orientation error is calculated as

$$\Delta\bar{O} = \bar{O}^{-1}\bar{O}_d, \tag{7}$$

$$\Delta\bar{X} = -\bar{O}_{xyz}\Delta\bar{O}_{xyz}, \tag{8}$$

where $\bar{O}$ is the current orientation quaternion, $\bar{O}_d$ is the desired orientation quaternion, and $*_{xyz}$ indicates that only the *x*, *y*, and *z* components of the quaternion are used.

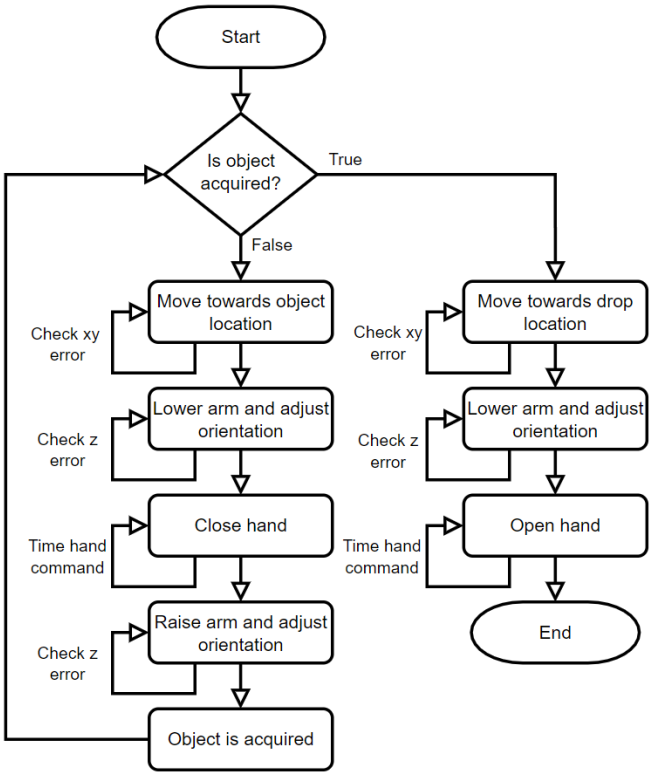

Figure 5. The flowchart for the pick-and-place task

C. *Transformations*

The method used to transform the target's pixel coordinates in the camera's frame to its cartesian coordinates in the frame of the base link of the robotic manipulator is introduced in this section.

1) Pixels to Cartesian Coordination Transformation

This section introduces how to get the target's cartesian coordinates from the video's pixel coordination. A screenshot of YOLOv5 object detect video is shown in Figure 6. First, the center pixel coordinate of the target is calculated as

$$\begin{bmatrix} P_{cx} \\ P_{cy} \end{bmatrix} = \begin{bmatrix} \frac{P_{leftx} + P_{rightx}}{2} \\ \frac{P_{topy} + P_{bottomx}}{2} \end{bmatrix}, \tag{9}$$

where P_{cx} and P_{cy} are the X-coordinate and Y-coordinate of the center of the object detection box, labeled red in Fig. 6. P_{leftx} and P_{rightx} are X coordinates for the left and right edge of the object detection box respectively. P_{topy} and $P_{bottomy}$ are Y coordinates for the top and bottom edges of the object detection box respectively.

The custom pixel resolution is set as 640 × 480 for Logitech web camera, and the table's dimension under the camera's view is measured as 1.07 × 0.8025 m. The following equation is used to calculate the target's cartesian coordination under the camera's frame,

$$P_{cartesian} = \left(P_{c\,(pixel)} - \frac{P_{\max(pixel)}}{2}\right)\frac{P_{\max(Cartesian)}}{P_{\max(pixel)}}, \tag{10}$$

where $P_{cartesian}$ is the cartesian coordination in the camera's frame. $P_{c\,(pixel)}$ is the pixel's coordinate of the target`s center. $P_{\max\,(pixel)}$ is the maximum pixel range for the web camera and $P_{\max\,(Cartesian)}$ is the maximum cartesian range, defined here as {640, 480} and {1.07 m, 0.8025 m}, respectively.

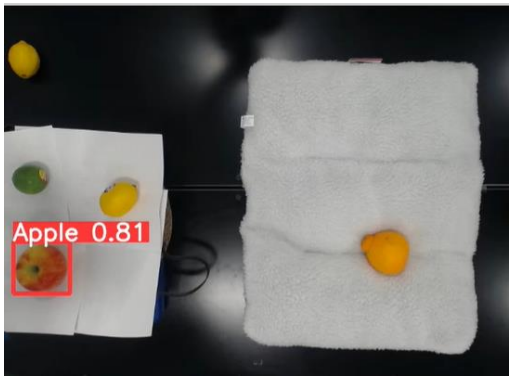

Figure 6. A screen shot of YOLO object detect video

2) Cartesian Coordination Transformation

To match the camera`s cartesian coordination system with the robotic manipulator`s coordination frame, the following equation is used

$$P_m = P_c \bullet T_A + T_B, \tag{11}$$

where P_m and P_c are the coordinates of robotic manipulator and camera`s cartesian coordinate frame, respectively. $T_A$ and $T_B$ are the transformation matrices. Since the XY-plane of the camera is mirrored compared with the manipulator`s XY-plane, T_A is set as {-1, -1, 0}. T_B T_B is used to adjust the camera`s position relative to the centroid`s position of the base link, link 1, of the manipulator and is set as {0.55, -0.14, 0.5}.

D. *Orientation Transformation*

As shown in Fig. 7, it is hard to grasp fruit with the default end-effector orientation and note that the soft hand has no joint

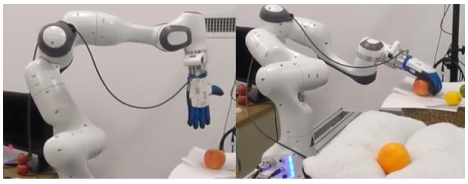

Figure 7. Default and pick-and-place orientation for the manipulator

at the wrist. To simulate the natural grasping behavior of humans, the end-effector need to rotate certain degrees about the XYZ axes. Based on several experiments, {170, 80, 30} is set as a suitable rotating vector about the X-axis, Y-axis, and Z-axis respectively in the unit of degrees. To convert the Euler angle to a quaternion, which is the default orientation expression in the Franka-Ros, the method introduced in [9] is used. By applying this method, the desired Euler angle is converted to the quaternion value {0.23, 0,723, 0.252, 0.601}.

E. Gazebo Simulation

The controller for the pick-and-place tasks is tested and the controller parameters are tuned in the Gazebo simulation environment before applying it to the hardware setup. A urdf/xacro based robotic manipulator model provided by franka-ros is used for the Gazebo simulation. Fig. 8 shows the FE Panda robot in the Gazebo environment.

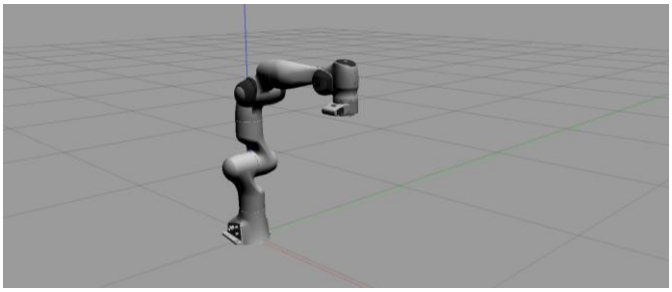

Figure 8. The Franka Emika Panda robot in Gazebo

F. Experiment Execution

Pick-and-place tasks are executed for two different fruits, apple and orange, in different locations to test the performance of the developed framework. The unit of coordinates introduced in this paragraph is meter, and they are considered in the cartesian coordination frame of the base link of the robotic manipulator. The XY coordinates are {-0.44, -0.32} for the location of the apple, labeled as the blue point, and {-0.6, -0.34} for the orange, labeled as the orange point. The XY coordinates of drop location, labeled as green point, is set as {0.6, -0.2}. Fig. 9 shows the locations of importance in the overhead camera view.

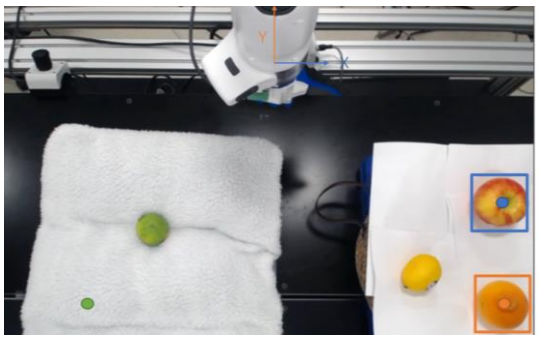

Figure 9. Important locations under camera view

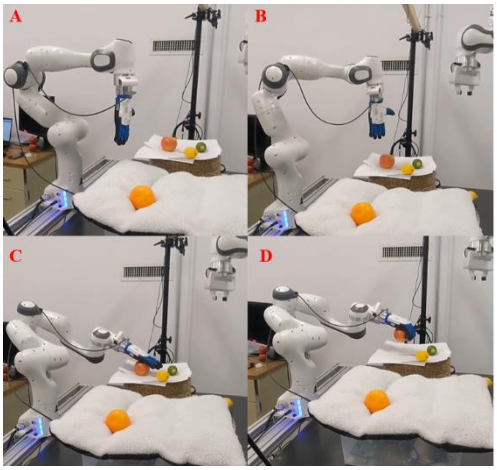

Figure 10. Action "Pick" for the apple pick-and-place task

The initial position and orientation are set as {0.555, 0, 0.521} and {0, 1, 0, 0} respectively for the end-effector. The impedance parameters with proper units are set as $K_t = 525$, $K_r = 52.5$, $B_t = 35$, and $B_r = 11$. During the execution of the code, the qb-SoftHand will perform the process which is introduced in the section task execution and move the targeted fruit to the drop location. The action "pick" is shown in Fig. 10, and the action "place" is shown in Fig. 11.

Fig. 10A shows the end-effector in the initial position. Figure 10B shows the end-effector move towards the object location. Fig. 10C shows that the manipulator lowers and

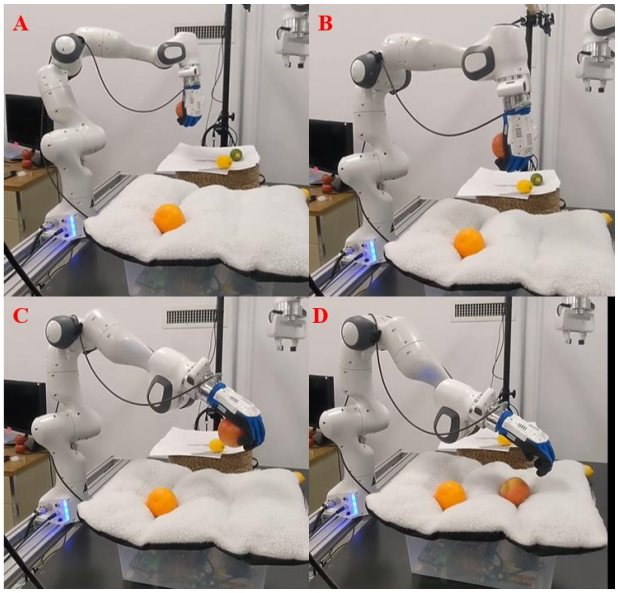

Figure 11. Action "Place" for the apple pick-and-place task

changes the orientation of the end-effector. Fig. 10D shows the action "grasping".

Fig. 11A shows the arm changes its end-effector's orientation to the default orientation to ensure that the object stays in the qb-SoftHand. Fig. 11B shows that the end-effector moves to the drop location. Fig. 11C shows that the manipulator lowers and changes the orientation of the end-effector again. Fig. 11D shows the action "place". The same process is done for the orange pick-and-place task to validate the performance

of the experimental setup. The video of the pick-and-place task is available here: https://youtu.be/snimlSePF5Q.

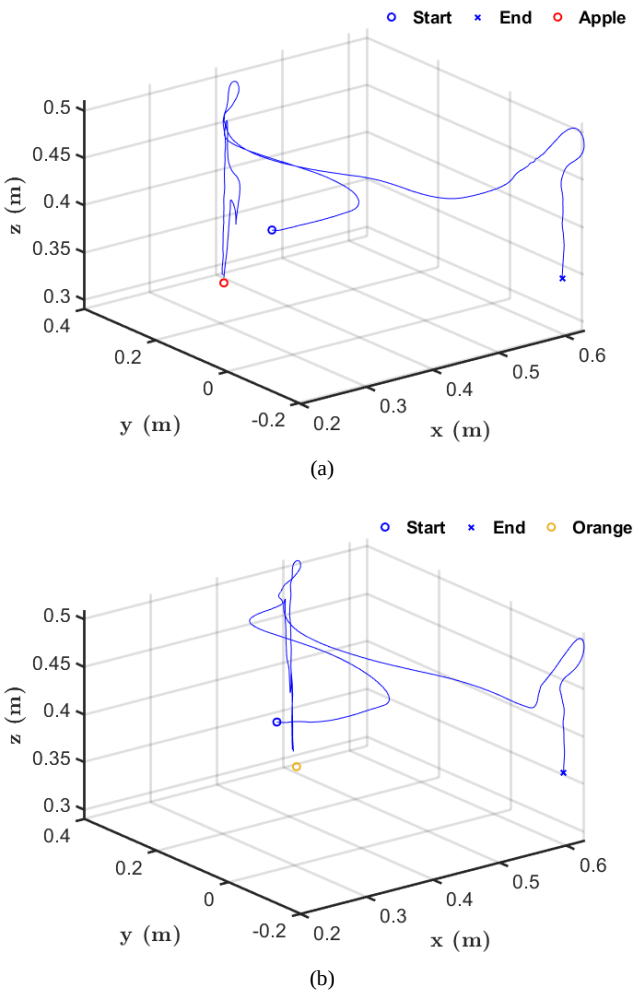

Figure 12. The end-effector's trajectories for the apple (a) and orange (b) pick-and-place tasks.

Fig. 12(a) and Fig. 12(b) show the recorded end-effector`s trajectories for the apple and orange experimental pick-and-place tasks. The trajectory plots shows that both trials follow similar paths. The sharp motion in the initial approach to the object in the apple pick-and-place is due to a misdetection of the manipulator for an apple. This can be improved in the future by providing more training images to the YOLOv5 model and improving the object detection process.

V. Conclusions and Future Work

This paper applies YOLOv5 object detection algorithm to a 7-DOF FE Panda robot to autonomously complete a fruit pick-and-place task by using an impedance control. An overhead web camera, an anthropomorphic robotic hand and the FE Panda robot are integrated together seamlessly using ROS. A human-like grasping of delicate fruit is demonstrated in an experimental setting. Comparing to other similar literature [3] and [10], this literature integrates Yolov5 computer vision modulus, FE manipulator, and qb-hand together to accomplish a vision-based fruit pick-and-place task automatically. The contribution of this article are: (1) a human-like grasping task rather than a robotic grasping task is performed. (2) The FE manipulator accomplishes this task automatically based on the camera sensing information, no human interaction required during the process. In the future, stereo camera will be applied to vision module to capture the 3D coordinates of the targets. Also, task-space trajectory planning will be developed to generate smoother paths with obstacle avoidances during the task.

Acknowledgement

This work was financially supported by the Natural Sciences and Engineering Research Council (NSERC), Killam Trust, and the Government of Nova Scotia, Canada.

References

[1] D. Mukherjee, K. Gupta, L. H. Chang, and H. Najjaran, "A survey of robot learning strategies for human-robot collaboration in industrial settings." Robotics and Computer-Integrated Manufacturing, 73, 102231, 2022.

[2] K. Rajpathak, K. C. Kodur, M. Kyrarini, and F. Makedon, "End-user framework for robot control," In *Proceedings of the 14th Pervasive Technologies Related to Assistive Environments Conference*, pp. 109-110, June, 2021.

[3] Q. Song, S. Li, Q. Bai, J. Yang, X. Zhang, Z. Li, and Z. Duan, "Object detection method for grasping robot based on improved YOLOv5," Micromachines, 12(11), 1273, 2021.

[4] G. Bauer and Y.J. Pan, "Telerehabilitation with exoskeletons using adaptive robust integral RBF-neural-network impedance control under variable time delays," In *Proceedings of the IEEE 30th International Symposium on Industrial Electronics (ISIE)*, Kyoto, Japan, June 2021, pp. 1-6.

[5] C. Cai, Y.J. Pan, S. Liu and L. Wan, "Task space bilateral teleoperation of co-manipulators using power based TDPC and leader-follower admittance control," In *Proceedings of the 47th IEEE Annual Conference of the Industrial Electronics Societies*, Toronto, Canada, October 2021, pp.1-6.

[6] J.X. Xu, Y.J. Pan and T.H. Lee, "A gain scheduled sliding mode control scheme using filtering techniques with applications to multilink robotic manipulators," ASME. J. Dyn. Sys., Meas., Control, 122(4): 641–649, 2000.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: unified, real-time object detection," In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, December 2016, pp. 779–788.

[8] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano and A. De Luca, "Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization," IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 4147-4154, 2019.

[9] M. Ben-Ari, "A tutorial on Euler angles and quaternions," *Weizmann Institute of Science, Israel*, *524*, 2014.

[10] Rajpathak, Kaustubh, Krishna Chaitanya Kodur, Maria Kyrarini, and Fillia Makedon. "End-User Framework for Robot Control." In The 14th PErvasive Technologies Related to Assistive Environments Conference, pp. 109-110. 2021.