Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 10004–10009

Object Recognition and Localization for Pick-and-Place Task using Difference-based Dynamic Movement Primitives

Qiguang Chen * **Lucas Wan** * **Ya-Jun Pan** *

* *Department of Mechanical Engineering, Dalhousie University, Halifax, Canada, B3H 4R2 (emails: qg234631@dal.ca, lucas.wan@dal.ca, yajun.pan@dal.ca)*

Abstract: In this paper, a framework is developed for three dimensional (3D) object recognition, localization, and anthropomorphic manipulation using YOLOv5, two low-cost webcams, and a Franka Emika (FE) Panda manipulator. Autonomous pick-and-place tasks are executed in a human-like manner using impedance control and novel difference-based Dynamic Movement Primitives (DMP) trajectory planning. The high accuracy vision system is developed with two webcams by using triangular relation to determine the object's position. The difference-based DMP algorithm and task execution plan are used to learn unique trajectories for different objects based on human demonstration and the object's coordinates. By integrating these components together, experimental studies on pick-and-place tasks are successfully performed for two different objects.

Keywords: Robotic Manipulator, Human Machine System, Nonlinear Control, Object Recognition, Dynamic Movement Primitives (DMP), Learn from Demonstration (LfD)

1. INTRODUCTION

As robotic manipulators continue to be integrated into our daily lives, three requirements are safe operation, anthropomorphic motion, and machine vision. Compliant, human-like movements of a robotic arm facilitates safe, physical collaboration between the robot and an operator. Vision systems to recognize and locate objects are necessary for the robot to make intelligent actions. You Only Look Once (YOLO) is an object recognition algorithm with high accuracy and fast processing speed Redmon et al. (2016).

In Rajpathak et al. (2021), a depth camera and YOLOv5 are implemented to complete a semi-automated pick-and-place task with a 7-degree-of-freedom (7-DOF) robotic manipulator. Depth cameras are expensive and processing of 3D point clouds can be computationally intensive. In Takahashi et al. (2020), a light-weight object detector is developed using a 3D depth camera and YOLOv3 with the goal of increasing the processing speed. Alternatively, researchers have taken the approach of using 2D cameras for 3D object localization. In Strbac et al. (2020), a stereo camera setup of two forward facing cameras is used with YOLOv3 to estimate the depth of detected objects but results in depth errors between 4% and 50%. The work in Wang et al. (2022) locates the 3D position of objects using stereo cameras and YOLOv5 and collects object using a robot arm with errors between 10% and 20%. Our approach uses the triangular relationship between two perpendicularly placed cameras and YOLOv5 to perform 3D object recognition and localization, achieving significantly higher accuracy than the stereo camera setup in Strbac et al. (2020) and Wang et al. (2022).

Learning from demonstration (LfD) is a popular method to embed human-like behavior and skills into the robot. Ijspeert et al. (2002) proposes a LfD framework based on Dynamic Movement Primitives (DMP). In Wu et al. (2022), a robot learns from human demonstrations for human–robot handovers and the accuracy of the task is improved by using an adaptive learning and control framework-based DMPs. The work in Lauretti et al. (2017) improves the accuracy of the regenerated trajectory by increasing the kernel density in specific areas to assistant a human in their daily activities. The work in Stulp et al. (2012) optimizes the parameters of motion primitives for efficiency and robustness in high dimensional problems. In our approach, a difference-based DMP algorithm is implemented to improve the practicability and safety of the regenerated trajectory.

Impedance control is a popular control method for human-robot interaction and has been applied in numerous studies Wan and Pan (2022), Bauer et al. (2020), Ficuciello et al. (2015). Song et al. (2019) gives an overview and comparison of basic concepts and principles, implementation strategies, crucial techniques, and practical applications concerning the impedance control of robotic manipulation. In our approach, impedance control was chosen to let manipulator interact with the environment.

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 Redmon et al. (2016). This approach was comparatively

⋆ This work was supported by NSERC (Natural Sciences and Engineering Research Council of Canada) and NSRIGS (Nova Scotia Research and Innovation Graduate Scholarship), Canada.

10.1016/j.ifacol.2023.10.865

faster than other object detection networks such as Faster R-CNN and R-FCN making it suitable for real-time applications, which is critical for manipulators operating in a dynamic environment, as in human-robot interaction.

In this paper, the YOLOv5 algorithm is implemented with two webcams and integrated with a 7-DOF Franka Emika (FE) robot to autonomously perform a pick-and-place task in a human-like manner using impedance control and difference-based DMP trajectory planning. Compared to our previous work in Chen et al. (2022), this approach uses two cameras to identify and locate objects in 3 dimensions rather than 2 dimensions. Additionally, the human-like motion in Chen et al. (2022) was manually coded, whereas in this paper the robot is taught different human-like motions using DMP algorithms to learn and generate desired trajectory.

The main contributions of this work are as follows:

1) A high accuracy object recognition and localization vision system (depth error is less than 1.5%) is developed using two low-cost webcams and YOLOv5.
2) A novel difference-based DMP algorithm is developed to learn and generate human-like motion specific to tasks associated with different objects.
3) A 7-DOF FE robot is enabled to autonomously recognize and locate objects, and execute the appropriate pick-and-place task using impedance control.

2. HARDWARE SETUP

Fig. 1 shows the equipment setup for this paper in the Advanced Control and Mechatronics (ACM) Lab at Dalhousie University. As labeled in Fig.1, the system includes a 7-DOF FE Panda robotic manipulator, one qb-SoftHand end-effector, a Logitech C920 HD webcam, and a Logitech C930E HD webcam.

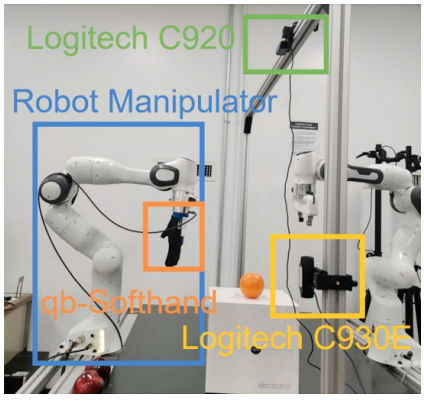

Fig. 1. Hardware setup

2.1 Robotic Manipulator

The FE Panda robot is a state-of-the-art 7-DOF robotic manipulator. The manipulator has 14-bit-resolution encoders and 13-bit-resolution torque sensors that provide high-level accuracy and sensitivity. The joint electronics provide 1kHz signal transmission frequency to ensure a smooth and stable signal exchange process. The control interface is available through Robot Operating System (ROS) packages and C++ API.

2.2 qb-Softhand Setup

A qb-SoftHand anthropomorphic robotic hand is employed as the end-effector on the FE robot for object pick-and-place tasks. The qb-SoftHand closely resembles a human hand and has several advantages. First, the qb-SoftHand is physically 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 with different sizes and ductility. Third, the qb-SoftHand has 19 anthropomorphic DOFs and only one motor, allowing for simple control implementation through ROS packages.

2.3 Cameras

Both Logitech cameras used for the YOLOv5 object detection in this paper have 1080p video resolution, 30 fps frame rate, 78° field of view.

3. METHODOLOGY

3.1 Localization of an Object with Two Cameras

The localization of an object in the 3D space can be achieved with two cameras and the YOLO algorithm. As shown in Fig. 1, one camera is placed at the top of the workspace, and another camera is placed at the side of the workspace. The position of the side camera is adjusted so that it is at the center of the left boundary of the top camera's frame. A YOLO algorithm is deployed on each camera. The YOLO algorithm can identify the object in the frame, encompass the object with a bounding box, and return the dimensions and the position of the bounding box in pixels. With known view angle and position of the cameras, the distance between the center of the bounding box and the cameras can be determined, so that the position of the object can be calculated and sent to the robotic arm. Figs. 2 and 3 show the schematics of the side and top view of the experimental setup. The distances

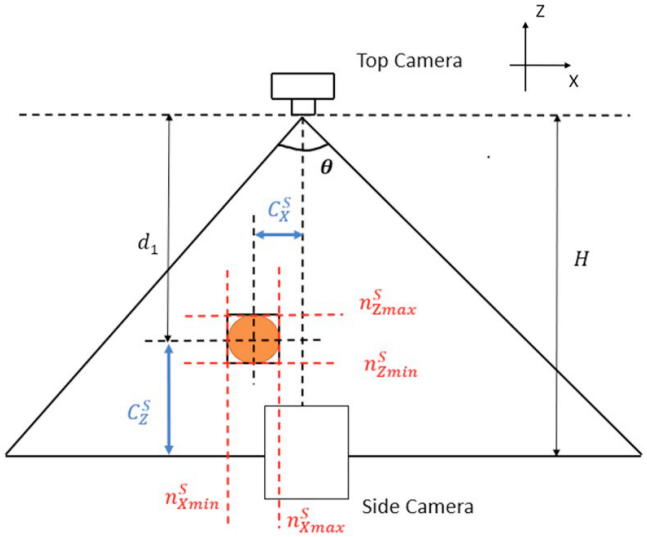

Fig. 2. Schematic of the side view

between the center line in X, Y, and Z axes of the side

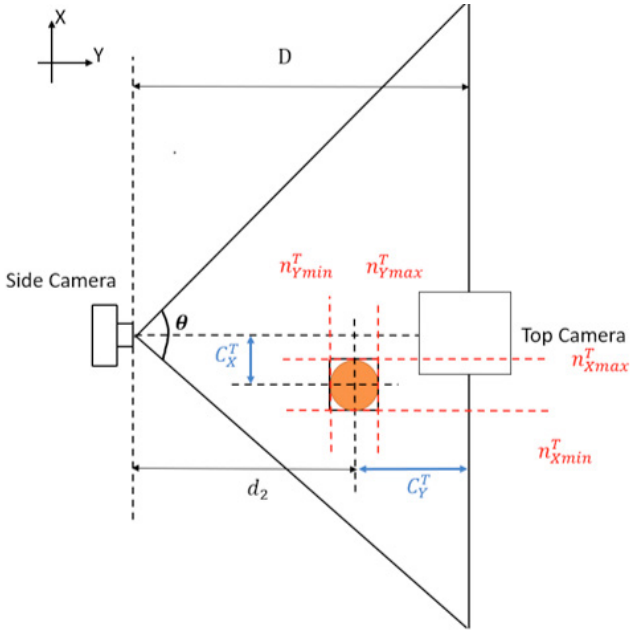

Fig. 3. Schematic of the top view

camera and top camera and object, $C^*_\bullet$ where $* = T, S$ for top and side and $\bullet = X, Y, Z$, can be calculated as

$$C^T_X = \frac{N_1 - (n^T_{X_{min}} + n^T_{X_{max}})}{2} \mathrm{d}_1 \tan(\frac{\theta}{N}), \quad (1)$$

$$C^T_Y = \frac{N_2 - (n^T_{Y_{min}} + n^T_{Y_{max}})}{2} \mathrm{d}_1 \tan(\frac{\theta}{N}), \quad (2)$$

$$C^S_X = \frac{N_2 - (n^S_{X_{min}} + n^S_{X_{max}})}{2} \mathrm{d}_2 \tan(\frac{\theta}{N}), \quad (3)$$

$$C^S_Z = \frac{N_1 - (n^S_{Z_{min}} + n^S_{Z_{max}})}{2} \mathrm{d}_2 \tan(\frac{\theta}{N}), \quad (4)$$

where N_1 and N_2 are the horizontal and vertical dimensions of the top and side cameras' frame in pixels, $n^*_{\bullet_{min}}$ and $n^*_{\bullet_{max}}$ are the pixel values of the detection box edges in X, Y, and Z axes in side and top camera's frame, θ is the view angle of the cameras, and N is the diagonal dimension of the top and side cameras' frame in pixels. Each pixel in the Logitech C920 camera represents an equal distance in reality, due to the Pixel-Perfect technology employed in the device. Note that the distance in the X-axis can be determined by either camera in Eqs. (1) and (3). The distances between the object and top camera, d_1, and side camera, d_2 can be calculated as:

$$d_1 = H + C^S_Z, \quad (5)$$

$$d_2 = D + C^T_Y. \quad (6)$$

Substituting (2) and (4) into (5) and (6), d_1 and d_2 can be calculated by:

$$d_1 = H + \frac{N_1 - (n^S_{Z_{min}} + n^S_{Z_{max}})}{2} \mathrm{d}_2 \tan(\frac{\theta}{N}), \quad (7)$$

$$d_2 = D + \frac{N_2 - (n^T_{Y_{min}} + n^T_{Y_{max}})}{2} \mathrm{d}_1 \tan(\frac{\theta}{N}), \quad (8)$$

where H and D are the distances between two cameras in x and z directions, respectively. By combining Eqs. (1) - (8), the X, Y, and Z distances between the object's center and the top camera, P^C_O, can be calculated as:

$$d_x = \frac{N_1 - (n^T_{X_{min}} + n^T_{X_{max}})}{2} \mathrm{d}_1 \tan(\frac{\theta}{N}), \quad (9)$$

$$d_y = \frac{N_2 - (n^T_{Y_{min}} + n^T_{Y_{max}})}{2} \mathrm{d}_1 \tan(\frac{\theta}{N}), \quad (10)$$

$$d_z = d_1, \quad (11)$$

$$P^C_O = [d_x,\ d_y,\ d_z]\,. \quad (12)$$

3.2 Transformations

To calculate the object's position in the manipulator's reference frame, the following equation is applied

$$P^M_O = P^C_O + P^O_C, \quad (13)$$

where P^M_O is the object's position in the manipulator's reference frame and P^M_C is the top camera's position in the manipulator's reference frame.

3.3 Dynamics of the Franka Emika Robot

The general form for a k-DOF serial robotic manipulator can be written as:

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + G(q) = \tau + \tau_f, \quad (14)$$

where $M(q) \in \mathbb{R}^{k\times k}$ is the symmetric and positive definite inertia matrix of the robot, $C(\dot{q}, q) \in \mathbb{R}^{k\times k}$ is the Coriolis and Centrifugal torque matrix, $G(q) \in \mathbb{R}^k$ the gravity torques matrix, $\tau \in \mathbb{R}^k$ is the control input torque, $\tau_f \in \mathbb{R}^k$ is the joint friction torque, and $q \in \mathbb{R}^k$, $\dot{q} \in \mathbb{R}^k$, and $\ddot{q} \in \mathbb{R}^k$ are the joint positions, joint velocities, and joint accelerations respectively. The dynamic matrices, M, C, and G are calculated internally by the FE robot model library and the friction, τ_f, is internally compensated for.

3.4 Impedance Controller

An impedance controller is used in this framework because it is easy to implement, robust, and can achieve safe, compliant physical interaction with human users. The feedback controller is designed in Cartesian space considering that the position of the objects are in Cartesian space. The control torque is computed as

$$\tau = J^T(-K\tilde{x} - B(J\dot{q})) + C(\dot{q}, q)\dot{q}, \quad (15)$$

where

$$K = \begin{bmatrix} K_t & 0 \\ 0 & K_r \end{bmatrix}, \quad (16)$$

$$B = \begin{bmatrix} B_t & 0 \\ 0 & B_r \end{bmatrix}, \quad (17)$$

$\tilde{x} \in \mathbb{R}^6$ contains the Cartesian position and orientation errors, $J \in \mathbb{R}^{6\times k}$ is the Jacobian matrix, $K_t, K_r, B_t, B_r \in \mathbb{R}^{3\times 3}$ are diagonal matrices that contain the translational and rotational impedance stiffness and damping parameters respectively. Note that the Cartesian orientation errors are the imaginary components of the quaternion error.

3.5 Dynamic Movement Primitives

The aim of DMP is to build an attractor model with a stable second order system (Ijspeert et al. (2013)). To change the target trajectory, the final state of the system can be changed by adjusting the attracting point. An

n-dimensional DMP trajectory can be represented by a second-order ordinary differential system,

$$\zeta \dot{v} = \alpha(\beta(g - y) - v) + f(t), \tag{18}$$
$$\zeta \dot{y} = v, \tag{19}$$

where ζ is a value-control parameter, $g \in \mathbb{R}^n$ is the goal state, $y \in \mathbb{R}^n$ current state, $v \in \mathbb{R}^n$ is the current velocity, and α, β are constants similar to P and D gains in the PD controller. The f term is a trajectory learning term which is expressed as

$$f(t) = \frac{\sum_{i=1}^{N} \Psi_i(t) w_i}{\sum_{n=i}^{N} \Psi_i(t) w_i}, \tag{20}$$

where Ψ_i are the kernel functions, w_i are weight parameters, and N is the number of kernel functions. Using Eq. 20, a complex trajectory can be described by changing the weight of different kernels. Since the trajectory learning term f is highly dependent on time, it is difficult to directly add other dynamic systems to it or synchronize multiple DOF trajectories. Therefore, a time-independent term x is used to replace the time for discrete type DMP,

$$\zeta \dot{x} = -\alpha_x x, \tag{21}$$

where α_x is a constant. The system will converge to zero for any initial value x_0. The parameter α_x will influence the convergence speed of the system. The kernel function Ψ is constructed as a radial basis function (RBF),

$$\Psi_i(x) = \exp(-h_i(x - c_i)^2) = \exp(-\frac{1}{\sigma_i^2}(x - c_i)^2), \tag{22}$$

where σ_i and c_i represent the width and the center location of the kernel function, Ψ, which are obtained by the demonstration trajectory. To ensure that the forcing term f converges to zero when the state of the system converges to g, the expression of f is changed to

$$f(t) = \frac{\sum_{i=1}^{N} \Psi_i(t) w_i}{\sum_{n=i}^{N} \Psi_i(t) w_i} x(g - y_0), \tag{23}$$

where y_0 is the initial state. In Eq. (23), x is used to ensure that f is converging to zero with decreasing x.

3.6 Difference-Based DMP

The main advantage of DMP is the ability to generate demonstration-based trajectories for different start and goal points. However, it performs poorly when there is a significant difference between the execution start and goal points and the demonstrated start and goal points. Since DMP is suited for small differences between the demonstrated and execution start and end points, a difference-based DMP approach is proposed and applied. In this case, the regenerated DMP trajectory is generated based on the current state of the manipulator as follows

$$\begin{aligned} \bar{x}_{start} &= x, \\ \bar{x}_{end} &= x + \delta_{DMP}, \\ \delta_{DMP} &= \hat{x}_{end} - \hat{x}_{start}, \end{aligned} \tag{24}$$

where x is the current position of the manipulator, $\bar{x}$ is the position used to regenerate the DMP trajectory, and $\hat{x}$ is the position of the demonstrated trajectory.

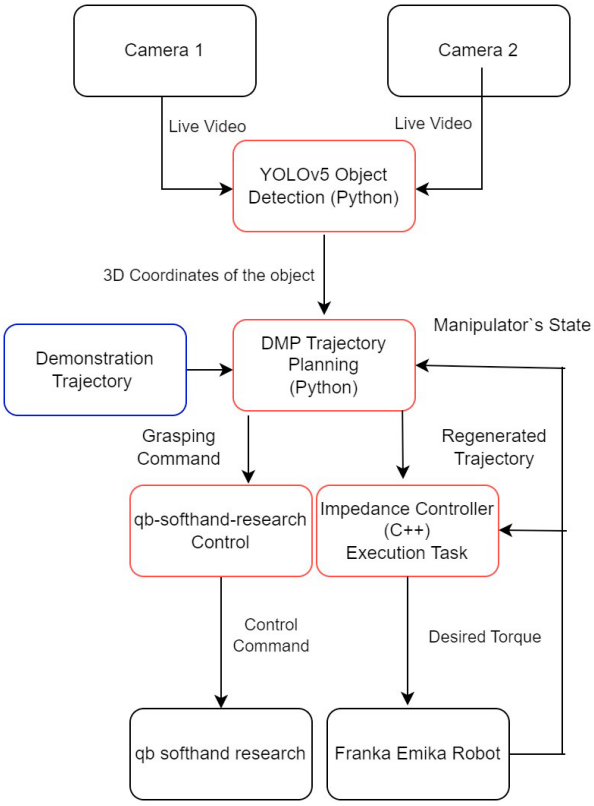

Fig. 4. ROS block diagram of the experiment

4. EXPERIMENTAL RESULT

4.1 Robot Operating System

ROS is used as the software to facilitate the communication between the multiple hardware components. Fig. 4 shows the ROS block diagram of the experimental framework, where black boxes represent the hardware, red boxes represent the ROS nodes, and blue boxes represent the data. First, two live videos are taken by two webcams and input to the YOLOv5 object detection algorithm. Next, the XYZ coordinates of the target object is calculated. The coordinates are input to the DMP trajectory planning algorithm to generate a desired trajectory based on the current end-effector and object positions. A set of actions is prescribed to perform the pick-and-place task which is shown in the following section. The generated trajectory is then input to the impedance controller. The desired joint torques are 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. Simultaneously, open and close commands are sent to the qb-SoftHand to allow the hand to execute the desired actions at the designated times.

4.2 Task Execution

The process of the pick-and-place action of the robotic manipulator that is executed within the DMP trajectory planning node is introduced in this section. First the manipulator is directed to the X-coordinate of the object. It is intentionally not directed to the Y-and Z-coordinates of the object to avoid collision. Next, the DMP trajectory is generated using Eq. (24) and executed to adjust the

robotic arm's end effector to the desired pick-up orientation. Then the manipulator is directed to the Y-and Z-coordinates of the object. When the object position is reached, the grasping action will be performed and the object can be picked by the end effector. The end effector will then move to a suitable moving orientation and the robotic arm will move to the target place location. Once the target location is reached, the end effector will be adjusted again to suitable place orientation via the generated DMP trajectory and drop the object at the desired location. Finally, the manipulator will return to the starting position and orientation.

4.3 Vision System Validation

This part of experiment is used to verify Eqs.(1)-(12). Fig. 5 shows the screenshots of YOLO systems running with top and side cameras. The object is identified and bounded by each YOLO algorithm. Each YOLO algorithm will return the pixel location of bounding box centers in the camera frame to the computer. Using Eqns. (1) - (13) listed in previous sections, the coordinates of the object are determined.

Table 1. Object positions with reference to the top camera $[x,\ y,\ z]$

Measured (cm)	Calculated (cm)	Absolute Error (cm)
$[-9.5,\ 23.2,\ 93.3]$	$[-9.3,\ 21.2,\ 91.9]$	$[0.2,\ 0.1,\ 0.4]$
$[7.5,\ 32.1,\ 93.5]$	$[8.0,\ 34.0,\ 92.4]$	$[0.5,\ 1.9,\ 0.9]$
$[0.0,\ 24.1,\ 93.5]$	$[0.0,\ 24.7,\ 92.1]$	$[0.0,\ 0.6,\ 1.4]$
$[0.8,\ 29.8,\ 74.4]$	$[0.3,\ 28.9,\ 74.2]$	$[0.5,\ 0.9,\ 0.2]$
$[-9.0,\ 1.9,\ 74.4]$	$[-8.7,\ 1.1,\ 74.1]$	$[0.3,\ 0.8,\ 0.3]$
$[8.5,\ 1.2,\ 74.6]$	$[8.0,\ 1.0,\ 74.2]$	$[0.5,\ 0.2,\ 0.4]$

Table 1 presents the validation data for calculating the position of an object, including the the measured distance between the top camera and the object, the distance between the top camera and the object calculated by our system, and the absolute error. As shown in Table 1, the maximum error is 1.9 cm and the maximum percentage error is under 1.5% in the Z-axis, which is suitable for this application. Some errors may be due to the inaccuracies of the bounding boxes, which can be improved by additional training of the YOLO algorithm.

4.4 Experimental Results

Two objects were considered separately in the experimental testing, an orange and a bottle. Considering that the orange must be approached from the top and the bottle must be approached from the side, two demonstration trajectories were recorded to regenerate different end-effector motions for each object. Because the coordinates of the object have already been determined at the beginning of the task, the movement of the robotic arm will not affect the experiment execution. The parameters used for regenerating the trajectory in (23) are $\alpha = 60$, $\beta = 15$, $\alpha_x = 1$, $N = 1000$, $\tau = \frac{1}{N_{demo}}$, and N_{demo} is the number of data points for each demonstration. The impedance parameters are $K_t = 525$ N/m, $K_r = 52.5$ N/m, $B_t = 35$ N/m, and $B_r = 11$ N/m. The top camera's position relative to the manipulator's base is $P_C^M = [0.64,\ 0.375,\ 1.065]\ m$. The parameter of cameras are $N_1 = 640$, $N_2 = 480$, and $N = 800$.

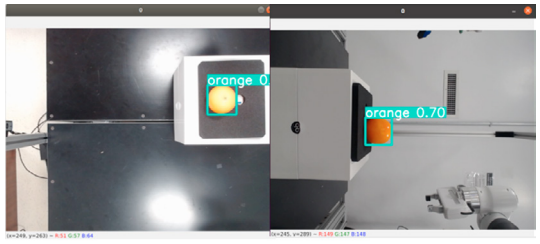

Fig. 5. Screenshots of YOLO system from top and side cameras

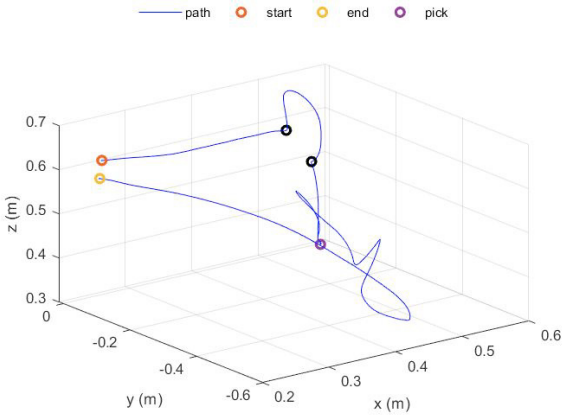

Fig. 6. The profile of the end-effector position for orange pick-and-place task

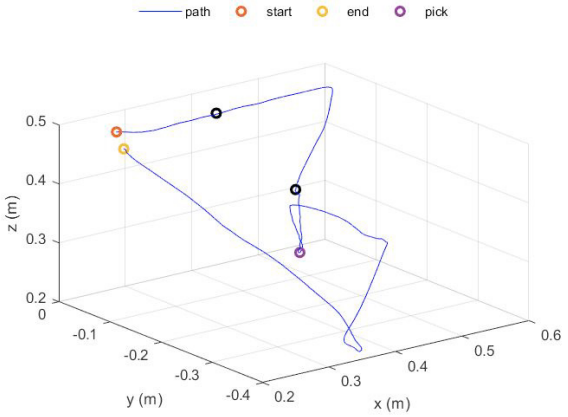

Fig. 7. The profile of the end-effector position plot for bottle pick-and-place task

Figs. 6 and 7 show the recorded end-effector's trajectories for the orange and bottle experimental pick-and-place tasks. The trajectories between black dots in Fig. 6 and Fig. 7 are generated by difference-based DMP.

Fig. 8A shows the end-effector in the initial position. Fig. 8B shows the end-effector grasping the object. Fig. 8C shows the manipulator passes an orange to a human. Fig. 8D shows the manipulator finishes the task and returns to the default position. Fig. 9A-D shows a similar task with a bottle with a different learned DMP trajectory where the manipulator approaches the bottle from the side.

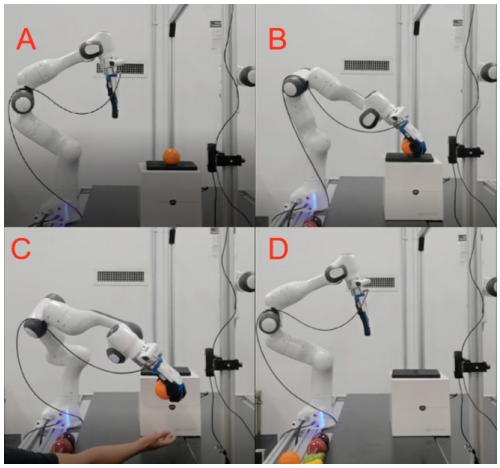

Fig. 8. Orange pick-and-place task

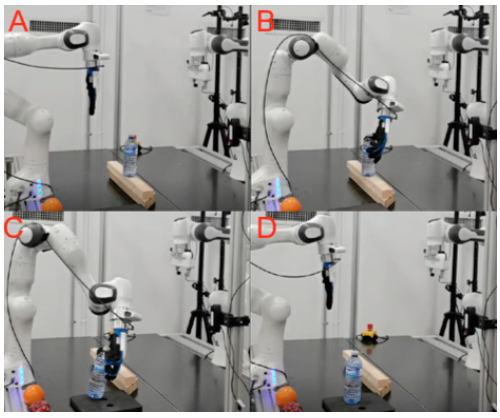

Fig. 9. Bottle pick-and-place task

The video of the pick-and-place task is available here: https://youtu.be/qlepM2Lrlno

5. CONCLUSIONS AND FUTURE WORK

In this paper, the YOLOv5 algorithm is implemented with two low-cost webcams to identify and locate objects in three dimensions with depth errors less than 1.5%. A difference-based DMP trajectory planning was designed to autonomously complete a pick-and-place task in a human-like manner using impedance control and a 7-DOF FE robot. In the future, the triangular localization method can be improved by using fixed focus length cameras, which allows more camera parameters to be used in calculating the object's coordinates to generalize to more situations. The YOLO datasets can be trained for specific objects to improve the accuracy of the object detection. Additional DMP trajectories will be demonstrated to complete different tasks, for example open a door. As well, several DMP trajectories will be integrated into a single motion to enable the execution of more complex tasks.

REFERENCES

Bauer, G., Pan, Y., and Shen, H. (2020). Adaptive impedance control in bilateral telerehabilitation with robotic exoskeletons. *In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 719–725.

Chen, Q., Wan, L., Ravichadran, P., Pan, Y.J., and Chang, Y.K. (2022). Vision-based impedance control of a 7-dof robotic manipulator for pick-and-place tasks in grasping fruits. *In proceedings of the 2022 CSME International Congress of Canadian Mechanical Engineering*, 4.

Ficuciello, F., Villani, L., and Siciliano, B. (2015). Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *IEEE Transactions on Robotics*, 31(4), 850–863.

Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2), 328–373.

Ijspeert, A.J., Nakanishi, J., and Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. *n Proceedings 2002 IEEE International Conference on Robotics and Automation*, 2, 1398–1403.

Lauretti, C., Cordella, F., Guglielmelli, E., and Zollo, L. (2017). Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics. *IEEE Robotics and Automation Letters*, 2(3), 1375–1382.

Rajpathak, K., Kodur, K.C., Kyrarini, M., and Makedon, F. (2021). End-user framework for robot control. *In The 14th PErvasive Technologies Related to Assistive Environments Conference*, 109–110.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Song, P., Yu, Y., and Zhang, X. (2019). A tutorial survey and comparison of impedance control on robotic manipulation. *Robotica*, 37(5), 801–836.

Strbac, B., Gostovic, M., Lukac, Z., and Samardzija, D. (2020). Yolo multi-camera object detection and distance estimation. *In 2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*, 26–30.

Stulp, F., Theodorou, E.A., and Schaal, S. (2012). Reinforcement learning with sequences of motion primitives for robust manipulation. *IEEE Transactions on robotics*, 28(6), 1360–1370.

Takahashi, M., Ji, Y., Umeda, K., and Moro, A. (2020). Expandable yolo: 3d object detection from rgb-d images. *In 2020 21st International Conference on Research and Education in Mechatronics (REM)*, 1–5.

Wan, L. and Pan, Y. (2022). Bilateral teleoperation of a multi-robot formation with time-varying delays using adaptive impedance control. *In IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 1739–1736.

Wang, P., Wiku, K.L., Saiful, H.A., Mo, C., Kim, S., and Han, X. (2022). Development of automatic recognition and gripping system of beekeeping auxiliary device us-ing stereo camera and object detection algorithm. *In 2022 ASABE Annual International Meeting (p. 1). American Society of Agricultural and Biological Engineers*, 1.

Wu, M., Taetz, B., He, Y., Bleser, G., and Liu, S. (2022). An adaptive learning and control framework based on dynamic movement primitives with application to human–robot handovers. *Robotics and Autonomous Systems*, 148, 103935.