Journal of Engineering and Technology

ISSN 2180-3811 eISSN 2289-814X https://jet.utem.edu.my/jet/index

VISION-BASED SAFETY SYSTEM FOR INDUSTRIAL COLLABORATIVE ROBOT

J. H. Ho[1] and T. F. Chay*[1]

[1] Faculty of Engineering and Technology,
Tunku Abdul Rahman University of Management and Technology
Jalan Genting Kelang, Setapak 53300, Kuala Lumpur, Malaysia.
corresponding: chaytf@tarc.edu.my

Article history:
Received Date: 23 April 2024
Revised Date: 10 August 2024
Accepted Date: 28 November 2024

Keywords:
Collaborative Robots, Vision-based Safety System, ROS, RViz, Kinect V2

Abstract— This paper proposes a new method to improve the safety operation of collaborative robots (cobots) by combining two existing safety features: distance-based speed control and obstacle avoidance. The system utilizes a Kinect V2 camera to capture depth data of the robot's workspace. This data is used to determine the distance between humans and the robot, allowing for speed regulation to ensure safety. Additionally, the depth data is converted into a map to enable obstacle avoidance by the robot. The proposed method was evaluated in a simulated environment using various pick-and-place scenarios. The results demonstrate the effectiveness of the system in ensuring safe human-robot collaboration.

The paper also identifies limitations, such as the limited depth perception due to a single camera. Future work will explore using multiple cameras and implementing the system on a physical cobot.

I. Introduction

Collaborative robots (cobots) are designed to assist humans in performing specific tasks within the same workspace. Cobots have numerous applications, but they are most commonly used in the manufacturing industry. Due to market demands for high-volume production, robots are employed to improve manufacturing efficiency. However, certain manufacturing sections or processes still require human operators. Cobots address this issue by taking over repetitive, heavy tasks, allowing human operators to focus on more complex tasks that robots cannot perform.

While operational, a cobot is not aware of its surroundings and functions solely based on its programming. This lack of awareness can lead to accidents during operation. To ensure human safety when working with cobots, advanced sensors are necessary. Basic sensors can only detect human presence and lack the ability to provide additional details. This is where machine vision becomes a superior alternative. Machine vision utilizes camera inputs, processes the data, and makes decisions based on the collected information. Essentially, it allows a computer to process and interpret real-time images. With sufficient data obtained through machine vision, cobots can safely maneuver alongside humans in a shared work environment.

Several previous studies have explored vision systems as safety mechanisms for cobots. However, most research has focused solely on controlling cobot speed based on the distance between the human and the cobot, or on implementing collision avoidance based on a human model detected by cameras. To improve these

ISSN: 2180-3811 Vol. XX No. X

existing systems, this research proposes a method utilizing a 3D camera, specifically the Kinect V2 camera. This method combines two functionalities:

- Integrating human-robot distance-based speed control
- Implementing obstacle avoidance features when the human gets too close to the cobot

The remainder of this paper follows a structured organization. Section II presents a review of previous work. Section III describes the methodology for implementing the proposed method. Section IV discusses the results obtained, and Section V provides a summary, identifies limitations, and outlines future work.

II. Literature Review

Several researchers have explored vision-based systems for obstacle avoidance and safety in collaborative robots. These approaches can be broadly categorized into three main areas: safety zones, collision dodging, and reactive path planning.

Safety Zones: This approach defines zones around the cobot with varying risk levels. Sensor data, often from RGB-D cameras like Kinect V2, is used to determine the distance between the cobot and potential obstacles (humans in most cases). The robot's speed and state are then adjusted based on the occupied zone (e.g., safe zone, warning zone, danger zone) [1-4].

Collision Dodging: This approach focuses on real-time obstacle detection and avoidance maneuvers. Techniques like Rapidly-exploring Random Trees (RRTs) are used to navigate around static obstacles, while repulsive vector fields are employed for dynamic obstacles [5]. Additionally, depth cameras can be used to predict human movement patterns and allow the cobot to adjust its path accordingly [6].

Reactive Path Planning: This approach utilizes various algorithms to plan safe trajectories for the cobot in dynamic environments. K-Nearest - Neighbor (KNN)

ISSN: 2180-3811 Vol. XX No. X

algorithms and Cartesian force calculations can be used to identify and avoid obstacles detected by Kinect cameras [7-10]. More advanced methods involve training neural networks on human pose data to allow the cobot to recognize human actions and plan collision-free paths [11-12].

In summary, vision systems offer a powerful tool for enhancing cobot safety. Existing research explores various techniques for obstacle detection, risk assessment, and path planning. Future advancements in sensor technology, computer vision algorithms, and robot control systems hold promise for even safer and more collaborative human-robot interactions.

III. Methodology

Based on the reviewed literature, a gap has been identified: the lack of integration between safety zones and collision dodging methods. This paper proposes a novel approach that combines these two safety systems. The Kinect V2 camera will be employed as the vision system to acquire point cloud data. The Panda Robot developed by Franka Emika will serve as the cobot platform for testing the proposed method. The research framework is illustrated in Figure 1.

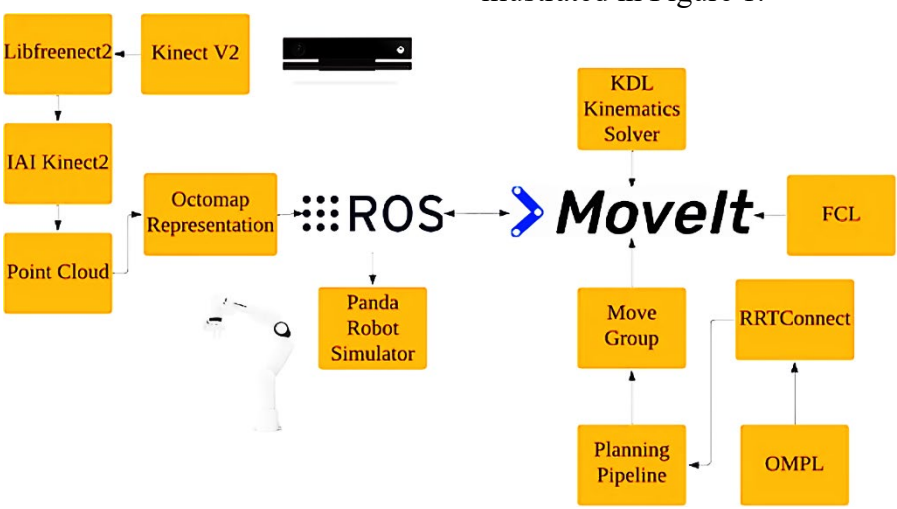

Figure 1: Research Framework

ISSN: 2180-3811 Vol. XX No. X

A. Data Acquisition and Robot Control

The Kinect V2 camera, equipped with the Libfreenect2 driver and IAI Kinect2 open-source libraries, will capture point cloud data of the environment. This data will be visualized using Octomap for a comprehensive understanding of the workspace.

For robot control, the KDL Kinematics solver will handle the robot's kinematic calculations. Additionally, MoveIt! will be employed for path planning, leveraging the Open Motion Planning Library (OMPL) and Flexible Collision Library (FCL) for trajectory planning and collision detection.

B. Robot Operating System (ROS) Integration

ROS, a highly flexible open-source software framework, will serve as the platform for robot control and integration. Due to its compatibility with various robots, including the Panda Robot chosen for this research, ROS allows for seamless integration.

Within ROS, the Panda Robot's description will be defined using URDF (Unified Robot Description Format) and SRDF (Robot Description Format) files. This configuration enables visualization of the robot model in RViz, a built-in ROS tool for robot simulation and visualization. As shown in Figure 2, proper configuration of the URDF and SRDF files will result in an accurate representation of the Panda Robot in RViz.

C. MoveIt! Integration and Path Planning

MoveIt! itself is a powerful suite of open-source tools encompassing trajectory planning, collision detection, and kinematic solvers. For this research, several open-source components will be integrated with MoveIt! to achieve the proposed method.

D. KDL Kinematics Solver

This solver automates the inverse kinematic calculations for the Panda Robot, ensuring precise movement control.

ISSN: 2180-3811 Vol. XX No. X

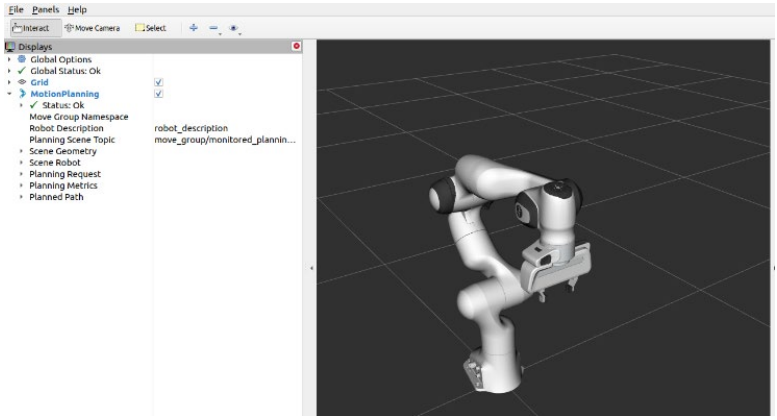

Figure 2: Display of Panda Robot in RViz

E. Open Motion Planning Library (OMPL) [13]

This library offers a vast collection of search algorithms for motion planning. The proposed method utilizes the RRT-Connect algorithm, which employs two simultaneously expanding Rapidly-Exploring Random Trees (RRTs). One RRT starts at the starting point, and the other starts at the goal point. Both trees expand outwards, dodging obstacles, until they connect, forming a collision-free path. Figure 3 illustrates the RRT-Connect algorithm in action [14].

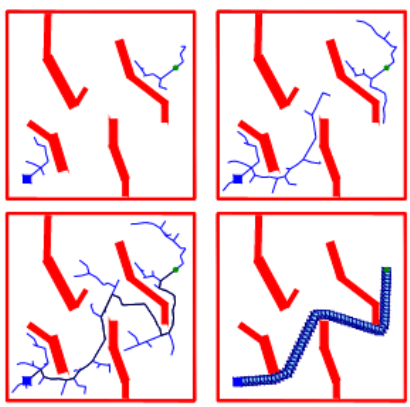

Figure 3: Illustration of RRT-Connect

F. Collision Detection with FCL

The Flexible Collision Library (FCL) [15] is incorporated for its efficient collision checking and proximity calculations. FCL excels at handling probabilistic collisions involving noisy point cloud data obtained from 3D cameras or LIDAR sensors. Additionally, it supports Octree

ISSN: 2180-3811 Vol. XX No. X

data structures, the spatial decomposition technique employed by Octomap for efficient environment representation.

G. Kinect V2 Camera Integration with Libfreenect2

The Libfreenect2 driver [16] serves as the middleware for integrating the Kinect V2 camera. This driver facilitates the transfer of various data streams, including RGB images, infrared (IR) images, depth images, and even combined RGB-D data. Figure 4 showcases example image data captured using Libfreenect2.

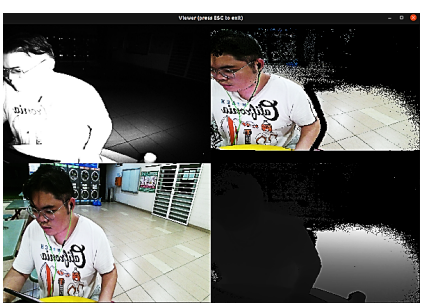

Figure 4: Data images extracted from Kinect V2 using Libfreenect2

H. Image Data Processing and Calibration

Following the acquisition of image data from the Kinect V2 camera, the open-source IAI Kinect2 library [17] is employed for data visualization within RViz. IAI Kinect2 offers an additional benefit: calibration of the Kinect V2 to enhance the quality of RGB, depth, and IR images. This calibration process involves displaying a sample image of a Checkered Board Pattern in front of the Kinect V2 camera. The calibration procedure itself is visualized in Figure 5. Once calibrated, IAI Kinect2 publishes the improved images as topics within the ROS framework.

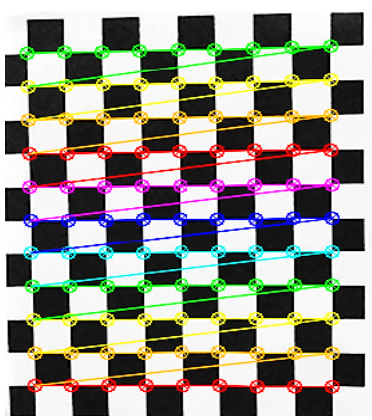

Figure 5: Kinect V2 Calibration using Checkered Board Pattern

I. Transformation for Accurate Point Cloud Visualization

When displaying depth images as 3D point clouds in RViz, it's

ISSN: 2180-3811 Vol. XX No. X

critical to ensure they align correctly with the real-world environment. This requires proper translation and rotation of the point cloud data using a transformation matrix.

In this research, the Kinect V2 will be positioned approximately one meter above the Panda Robot, capturing depth data in front of it. To achieve accurate visualization, a transformation matrix is necessary. Equation (1) defines the translation, while Equations (2), (3), and (4) represent rotations around the x, y, and z axes, respectively. These transformations ensure the point cloud aligns with the robot and its surroundings in Rviz as shown in Figure 6.

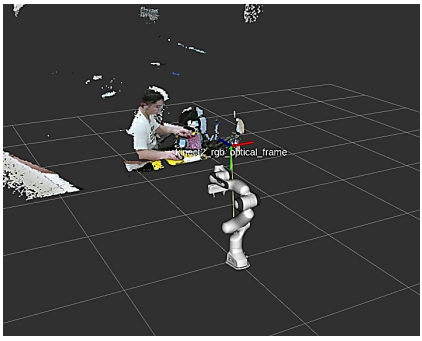

Figure 6: Alignment and transformation of the Kinect V2 with the environment in Rviz

$$\mathbf{T} = \mathbf{T}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \quad (1)$$

$$\mathbf{R_x} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{cos\theta} & \boldsymbol{-sin\theta} \\ \mathbf{0} & \boldsymbol{sin\theta} & \boldsymbol{cos\theta} \end{bmatrix} \quad (2)$$

$$\mathbf{R_y} = \begin{bmatrix} \boldsymbol{cos\theta} & \mathbf{0} & \boldsymbol{sin\theta} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{-sin\theta} & \mathbf{0} & \boldsymbol{cos\theta} \end{bmatrix} \quad (3)$$

$$\mathbf{R_z} = \begin{bmatrix} \boldsymbol{cos\theta} & \boldsymbol{-sin\theta} & \mathbf{0} \\ \boldsymbol{-sin\theta} & \boldsymbol{cos\theta} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (4)$$

J. Real-Life Application Scenario and Testing

To evaluate the proposed method in a practical setting, a pick-and-place task is simulated within RViz. A virtual table and object are configured, and the Panda Robot's movement is planned using MoveIt!'s Move Group function. The trajectory is calculated using the RRT-Connect search algorithm, as described earlier. The simulated pick-and-place scenario is visualized in Figure 7.

The next step involves converting the depth data from Figure 6 into an Octomap representation. This representation registers the obstacles detected by the Kinect V2 camera for collision avoidance by the Panda Robot.

ISSN: 2180-3811 Vol. XX No. X

The configuration is done within MoveIt!, and the resulting Octomap is shown in Figure 8.

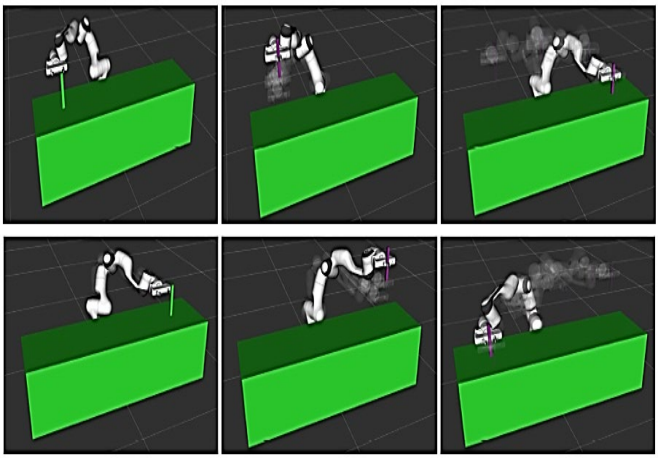

Figure 7: Visualisation of Robot Pick and Place scenarios in Rviz

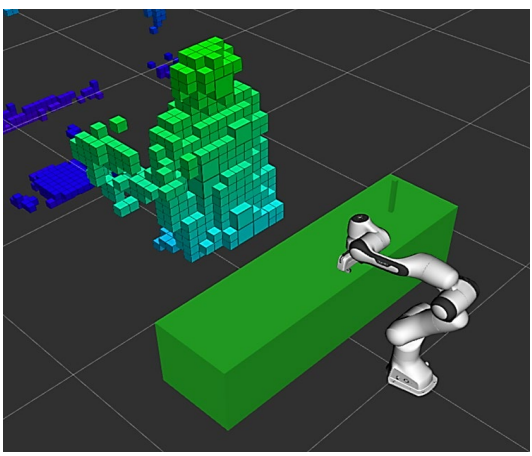

Figure 8: Example of Octomap representation of objects in the surrounding

K. Collision Avoidance with Octomap and FCL

The Octomap serves as a dynamic obstacle map that the robot actively avoids during its movements. When the planned trajectory encounters an obstacle in the Octomap, FCL detects the potential collision and triggers the Panda Robot to stop. Subsequently, a new collision-free trajectory is calculated using OMPL's RRT-Connect algorithm.

ISSN: 2180-3811 Vol. XX No. X

L. Human-Robot Distance and Speed Regulation

IAI Kinect2 displays depth data as point clouds in RViz, where each point has its own 3D coordinates. To determine the closest human to the Panda Robot, the system iterates through these coordinates and continuously updates the closest point. The Euclidean distance formula (shown in Equation (5) and its simplified form in Equation (6)) is used to calculate the distance between the robot and the closest point.

If the closest human is within 0.6 meters of the robot's workspace, the robot's speed is reduced to 10% of its original speed.

$$\boldsymbol{d(p,q)} = \sqrt{\sum_{i=1}^{n}(\boldsymbol{q_i} - \boldsymbol{p_i})^2} \quad (5)$$

$$\boldsymbol{d(p,q)} = \sqrt{\begin{matrix}(\boldsymbol{x_1} - \boldsymbol{x_0})^2 + (\boldsymbol{y_1} - \boldsymbol{y_0}) \\ +(\boldsymbol{z_1} - \boldsymbol{z_0})^2\end{matrix}} \quad (6)$$

M. Experimental Design and Testing Scenarios

With all configurations in place, the proposed method is tested in four different scenarios around the Panda Robot (illustrated in Figures 9 to 12):

1. No human presence in the workspace.
2. Human is near the workspace.
3. Human blocks the robot's path at a fixed position.
4. Human blocks the robot's path at a dynamic (changing) position.

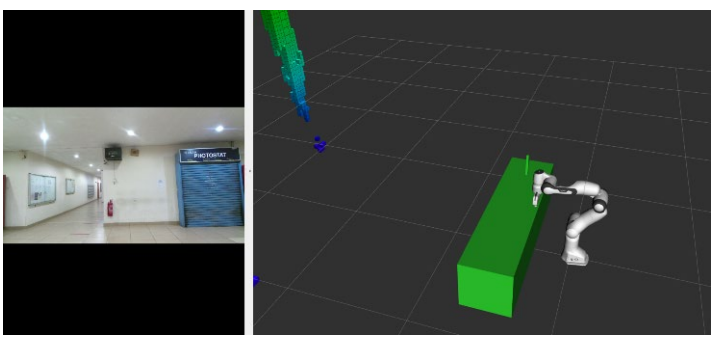

Figure 9: No human presence at the workspace

ISSN: 2180-3811 Vol. XX No. X

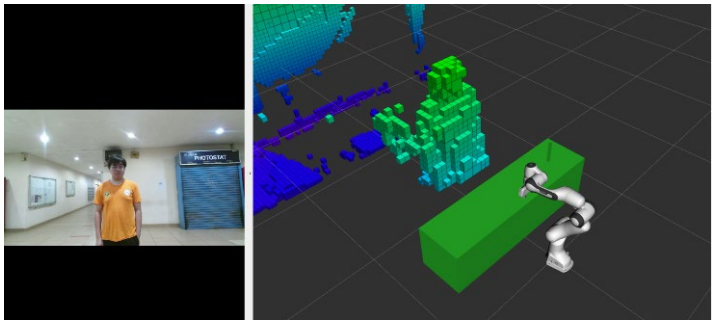

Figure 10: Human is near the workspace

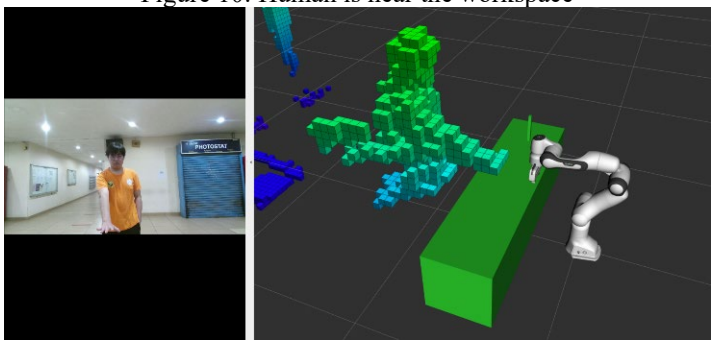

Figure 11: Human blocks the robot's path at fixed position

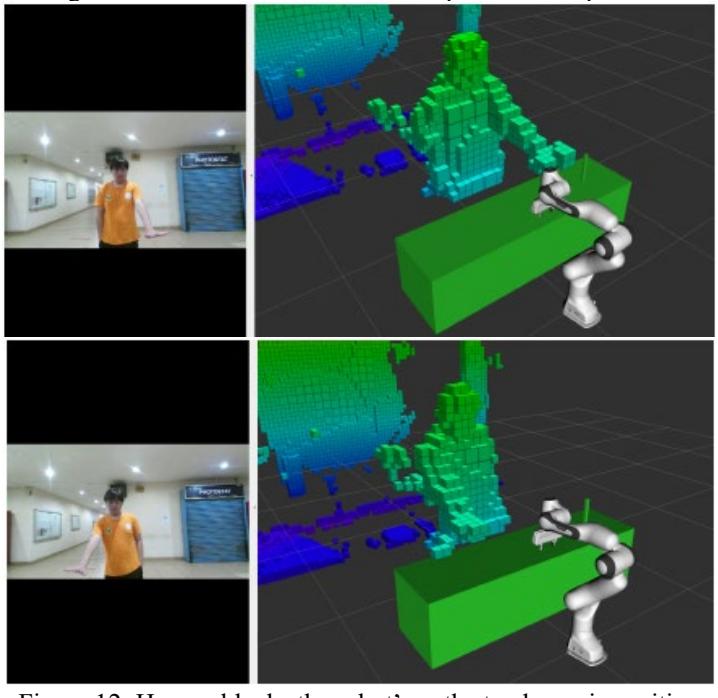

Figure 12: Human blocks the robot's path at a dynamic position

IV. Result and Discussion

Experiments were conducted to test the efficiency of the proposed method in collision avoidance and speed regulation. To test the efficiency, trajectory computational time and overall pick-and-place process handling time were captured and compared. Ten data sets were collected for each of the four testing scenarios. These data sets were then tabulated and visualized using box-and-whisker plots (Figures 13 and 14). The data captured the total computational time required by the RRT-Connect search algorithm to calculate new trajectories and the overall time taken by the Panda Robot to complete a single pick-and-place process.

A. Analysis of Trajectory Computational Time

Figure 13 presents the box-and-whisker plot for the total trajectory computational time. As observed, in Scenario 1 (no human presence, orange) the average time to calculate a new path is 0.968 seconds. Similarly, Scenario 2 (human near, green) shows minimal deviation from Scenario 1, as there are no obstacles present in either case. However, the computational time increases significantly in Scenarios 3 (human blocking at fixed position) and 4 (human blocking at dynamic positions, purple) due to the presence of obstacles. The RRT-Connect algorithm requires additional time to find collision-free paths around these obstacles. Notably, Scenario 4 exhibits the highest computational time because the constantly changing human position necessitates frequent path recalculations.

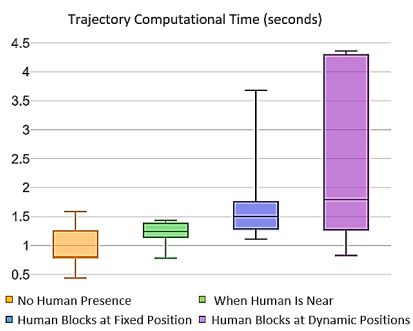

Figure 13: Graph plot of trajectory computational time

B. Analysis of Overall Process Handling Time:

The next box-and-whisker plot (Figure 14) explores the overall time taken to complete a single

ISSN: 2180-3811 Vol. XX No. X

pick-and-place process by the robot. As expected, Scenario 1 (no human presence) has the lowest average time (15.3 seconds) due to the absence of obstacles and slowdowns. Scenario 2 (human near) demonstrates an increase in overall process time because the robot's movement speed is reduced when a human is detected near its workspace. Interestingly, Scenarios 3 (fixed blocking position) and 4 (dynamic blocking positions) show similar average times despite the dynamic nature of the latter. This is because the robot's speed reduction in both scenarios dominates the overall process time. However, Scenario 4 exhibits a wider range of completion times. This can be attributed to situations where the human blocks the robot's path after the trajectory has already been calculated. In such cases (illustrated in Figure 15), FCL detects the potential collision and triggers the robot to stop, forcing a new trajectory calculation (shown in Figure 16). These additional calculations and stops contribute to the increased overall process time observed in Scenario 4.

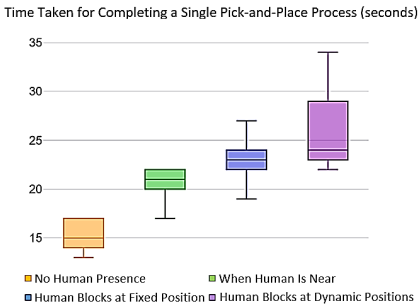

Figure 14: Graph plot for the time to complete a single pick-and-place process

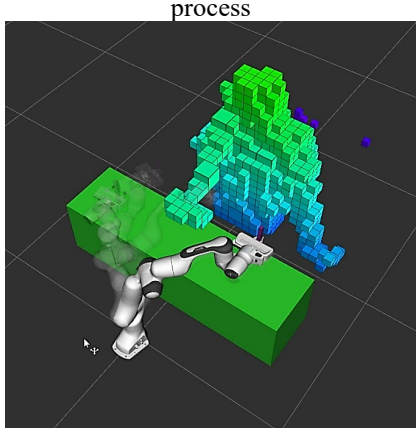

Figure 15: Human blocks the calculated robot's path

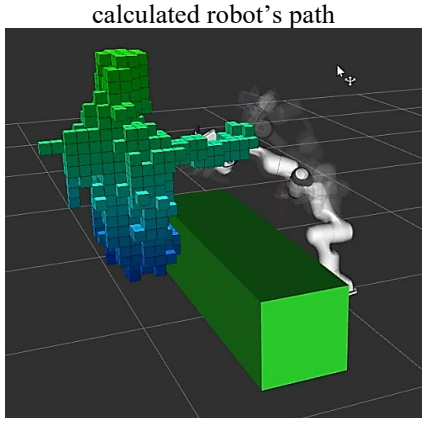

Figure 16: New trajectory is calculated and executed (moved) by the robot

ISSN: 2180-3811 Vol. XX No. X

V. Conclusion

This paper presented a novel approach for cobot safety that integrates human-robot distance speed control and obstacle avoidance features. The system utilizes a Kinect V2 camera equipped with the Libfreenect2 driver and IAI Kinect2 library to capture depth data. This data is processed and converted into an Octomap representation, which facilitates the determination of the distance between humans and the Panda Robot's workspace. The Open Motion Planning Library (OMPL) with the RRT-Connect search algorithm is employed for obstacle avoidance, while the Flexible Collision Library (FCL) handles collision detection.

The proposed method's effectiveness was evaluated through experimentation using four distinct pick-and-place scenarios that simulate real-life applications. The results demonstrate that the system allows for safe human-robot collaboration, enabling human operators to work alongside the cobot without unnecessary interruptions.

This research acknowledges certain limitations. The use of a single Kinect V2 camera restricts the system's ability to capture depth data behind humans. Future work will address this by integrating multiple Kinect V2 cameras to achieve more comprehensive depth perception. Additionally, while the pick-and-place scenarios were tested in a virtual simulation environment, future endeavors will involve implementing and testing the proposed method on a real cobot in a physical setting.

VI. References

[1] S. Kallweit, R. Walenta and M. Gottschalk, "ROS Based Safety Concept for Collaborative Robots in Industrial Applications," *Stephan Kallweit, Robert Walenta, Michael Gottschalk,* 2016.

[2] S. Pasinetti, C. Nuzzi, M. Lancini, G. Sansoni, F. Docchio and A. Fornaser, "Development and Characterization of a Safety System for Robotic Cells Based on Multiple Time of Flight (TOF) Cameras and Point Cloud Analysis," pp. 34-39, 2018.

[3] N. Nikolakis, V. Maratos and S. Makris, "A Cyber-Physical System (CPS) Approach for Safe Human-Robot Collaboration in a Shared Workplace," *Robotics and Computer-Integrated Manufacturing,* vol. 56, pp. 233-243, 2019.

[4] T. Malm, T. Salmi, I. Marstio and J. Montonen, "Dynamic Safety System for Collaboration of Operators and Industrial Robots," *Open Engineering,* vol. 9, no. 1, pp. 61-71, 2019.

[5] G. Liu and Y. Jiang, "Research on Dynamic Trajectory Planning of Collaborative Robots Based on RRT-RV Algorithm," pp. 1020-1023., 2018.

[6] J. S. Park, C. Park and D. Manocha, "Intention-Aware Motion Planning Using Learning-Based Human Motion Prediction," 2019.

[7] J.-h. Chen and K.-t. Song, "Collision-Free Motion Planning for Human-Robot Collaborative Safety under Cartesian Constraint," pp. 4348-4354., 2018.

[8] H. Nascimento, M. Mujica and M. Benoussaad, "Collision Avoidance in Human-Robot Interaction using Kinect Vision System Combined with Robot's Model and Data," pp. 10293-10298, 2020.

[9] G. Du, Y. Liang, G. Yao, R. Murat and H. Yuan, "Active Collision Avoidance for Human-Manipulator Safety," *IEEE Access,* vol. 10, pp. 16518-16529, 2022.

[10] Z. Liu, X. Wang, Y. Cai, W. Xu, Q. Liu and Z. Zhou, "Dynamic Risk Assessment and Active Response Strategy for Industrial Human-Robot Collaboration," *Computers and Industrial Engineering,* vol. 141, p. 106302, 2020.

[11] J. A. Garcia-Esteban, L. Piardi, P. Leitao, B. Curto and V. Moreno, "An interaction strategy for safe human Co-working with industrial collaborative robots," pp. 585-590, 2021.

[12] H. Liu and L. Wang, "Collision-free Human-Robot Collaboration Based on Context Awareness," *Robotics and Computer Integrated Manufacturing,* p. 101997, 2021.

[13] A. Şucan, M. Moll and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine,* vol. 19, no. 4, pp. 72-82, 2012.

[14] J. Kuffner and S. LaValle, "Rrt-connect: An Efficient Approach to Single-query Path Planning," vol. 2, pp. 995-1001, 2000.

[15] J. Pan, S. Chitta and D. Manocha, " A General Purpose Library for Collision and Proximity Queries," *2012 IEEE International Conference on Robotics and Automation,* pp. 3859-3866, 2012.

[16] L. Xiang, F. Echtler, C. Kerl, T. Wiedemeyer, Lars, hanyazou, R. Gordon, F. Facioni, laborer2008,

R. Wareham, M. Goldhoorn, alberth, gaborpapp, S. Fuchs, jmtatsch, J. Blake, Federico, H. Jungkurth, Y. Mingze, vinouz, D. Coleman, B. Burns, R. Rawat, S. Mokhov, P. Reynolds, P. Viau, M. Fraissinet-Tachet, Ludique, J. Billingham and Alistair, "libfreenect2: Release 0.2," 2016.

[17] T. Wiedemeyer, "IAI Kinect2," 2014-2015. [Online]. Available: https://github.com/code-iai/iai_kinect2. [Accessed 12 April 2023].