Conceptual design and simulation study of an autonomous indoor medical waste collection robot

Shawn Shaju[1], Thomas George[1], Jithin Kunnath Francis[1], Manu Joseph[1], Mervin Joe Thomas[2]
[1]Department of Mechanical Engineering, Rajagiri School of Engineering and Technology, Ernakulam, India
[2]Indian Institute of Technology, Palakkad, India

Article Info

Article history:

Received Sep 27, 2022
Revised Nov 18, 2022
Accepted Dec 21, 2022

Keywords:

Artificial neural network
Mobile robot
Robot Operating System
Robotic manipulator
ROS Navigation Stack

ABSTRACT

Solid waste management is one of the critical challenges seen everywhere, and the coronavirus disease (COVID-19) pandemic has only worsened the problems in the safe disposal of infectious waste. This paper outlines a design for a mobile robot that will intelligently identify, grasp, and collect a group of medical waste items using a six-degree of freedom (DoF) arm, You Only Look Once (YOLO) neural network, and a grasping algorithm. Various designs are generated before running simulations on the selected virtual model using Robot Operating System (ROS) and Gazebo simulator. A lidar sensor is also used to map the robot's surroundings and navigate autonomously. The robot has good scope for waste collection in medical facilities, where it can help create a safer environment.

Corresponding Author:

Mervin Joe Thomas
Indian Institute of Technology
Palakkad, India
Email: mervinjoe4@gmail.com

1. INTRODUCTION

In many developing and even some developed countries, the novel coronavirus disease (COVID-19) has overburdened the public healthcare system. First reported in Wuhan, China, in December 2019, the pandemic continues to be a significant public health threat worldwide [1]. With the pandemic prolonging its stay, there are many hospitalizations. In addition to ensuring proper treatment of the infected patients, it becomes crucial to manage the healthcare waste properly. It is common to see used face masks, plastic bags, bottles, gloves, personal protective equipment (PPE), and other dangerous waste materials littered in public areas [2]. Due to the highly transmissible nature of the disease, medical waste could harbor coronavirus and other infectious pathogens, putting healthcare workers and citizens at risk, and raising several challenges in the safe disposal of affected waste [3]. This brings forth the importance of having an autonomous waste collection system to reduce physical contact between the worker and the waste to reduce contracting an infectious disease. An autonomous robot helps guarantee compliance with social distancing measures likely to remain in place after the COVID-19 crisis. Research in robotic technology shows the capability of robots to aid humans in performing hazardous tasks and rescue services in real-time [4]. Similarly, various researchers have suggested using robotic systems for waste collection and sorting to minimize the risk of infection [5]. Visual technology and computer vision advancements have become practical tools for robots to sense the external environment [6]. Simulation studies are significant for robot programmers to assess and predict the behavior of the robot. It also aids in verifying and optimizing the process path planning of the robot [7], [8]. Research has been conducted into making smarter robotic systems that can learn to process various new data for manufacturing and other industries [9].

This work explains the development of an autonomous vision-based multi-terrain waste collection robot to collect infectious healthcare waste in hospitals. To reduce direct human interaction with contagious environments, this work integrates soft computing techniques and features such as neural network-based computer vision, motor-controlled robotic manipulator for accurate pick and place operation, object grasping by sensing depth, autonomous navigation, and obstacle avoidance in real-time. In previous work in robotic cleaning, Sivakumar *et al.* [10] indicated simple ultrasonic sensors for navigation and incorporates a vacuum suction mechanism to collect waste. This requires a high-powered motor and a bigger battery to power the mobile robot. This increases the weight of the robot and poses limitations on the size and weight of objects that get picked up and is dependent on the size of the vacuum hose and the motor power rating. Our research presents an improved waste collection mechanism with replacing the collection of waste items by vacuum suction with a gripper-based approach. The navigation mechanism is also improved by adding a lidar range sensor to sense the environment better and help create a map for navigation. Sengupta *et al.* [11] also used ultrasonic sensors for their robot to navigate the environment but have experimented with a neural network for object detection. A neural network helps identify waste objects better, but its full potential is realized when combined with a gripper. Our research adds a robotic gripper mechanism combined with a vision system and a trained neural network to produce a mobile unit that delivers more efficient waste collection.

A design for the robot will be finalized after considering different parameters that directly affect the working of the robot. Finite element analysis (FEA) tools will need to be utilized to ensure that the robot frame can withstand the necessary loads. Different materials and their yielding characteristics will be studied to choose a suitable material for the frame. The robot operating system (ROS) has good potential to be used as the brain and command center of the robot due to the vast amount of software packages and customization it offers. Chitta *et al.* [12] detailed the implementation of the ros_control framework within the ROS and how it can be used to establish communication between various hardware controllers and programs like MoveIt and ROS Navigation Stack to enact path planning for arm and autonomous navigation, respectively. Since ros_control is robot-agnostic, it should be able to help create a control mechanism for this proposed robot. Gorner *et al.* [13] detailed the use of the task constructor within MoveIt software in setting tasks for planning actions or motions for the arm or its components. Furthermore, Hernandez-Mendez *et al.* [14] showed successful utilization of MoveIt for object grasping using a three degree of freedom (DoF) arm. This paper plans to apply MoveIt and its Task Constructor to accurately move a six DoF arm and execute each motion task in its order. MoveIt's integration with the Gazebo and Rviz helps test and simulate the working of the arm in a virtual environment considering all physical parameters like gravity, inertia, etc. To implement computer vision and object detection, this paper explores the flexibility and efficiency of the you only look once (YOLO) algorithm, first designed by Redmon *et al.* [15]. Bochkovsky *et al.* [16] improved upon the work of Redmon, with the YOLO algorithm with improvements in the speed and accuracy of object detection. This network will be studied to determine its efficiency when running on a low-power computing unit with a custom-trained dataset. Pas *et al.* [17] described their breakthrough in creating a very accurate object grasping algorithm that can work with parallel and suction grippers. This work aims to combine the grasping algorithm with MoveIt and establish a successful grasping action of the waste object.

The research contribution here is implementing a design approach that creates a compact, rugged, and reliable robot while combining different navigation and vision sensors using software that is easily configurable and adaptable to the needs of the robotic system. This paper also contributes to the knowledge pool of robotic grasping by showcasing efficient robot grasping on low-power computers with fast object detection and accurate grasping. This paper is organized as follows. The conceptual designs are explored in section 2, followed by their structural design and FEA in section 3. A description of the motor control and the simulation study performed for the robot control is explained in section 4. Finally, the conclusions derived and the scope for future work are presented in section 5.

2. CONCEPTUAL DESIGN AND SELECTION

Based on the application, many criteria exist for designing a suitable robot. This section explains the various conceptual designs proposed for the waste collection robot and explores the parameters used to judge the different concepts. Figure 1 showcases four robot concept designs proposed for waste collection, incorporating arm and wheel variations. The designs were modeled using SolidWorks modeling software and scored and ranked on various parameters to choose the most appropriate design.

SolidWorks was selected due to the ease of modeling and exporting the design to a unified robotics description format (URDF) format, and the variety of modeling and FEA tools offered [18]. All designs feature an onboard bin to collect the grasped waste items. The depth camera is placed on a raised cylindrical pedestal for a wider-angle view of the ground. A lidar sensor is placed between the camera and the robotic manipulator. The continuous tracked and conventional wheel mobile bases were considered during the design stage for the robot locomotion. Legged robots were not considered since they have a more complicated control architecture and are expensive. Continuous tracks enable the robot to traverse rough and uneven terrain with low effort. However, the added weight on tracks would pose a challenge in providing the necessary power to the robot. Wheeled designs help the robot move quickly within indoor environments. Since this study focuses on indoor applications, wheeled robots are preferred over the tracked robot base. A six DoF robotic arm is desired to suit the pick-and-place application of biological and non-biological waste materials. The robot manipulator uses a gripper to easily grip waste materials like used masks, syringes, tissue paper, water bottles, plastic bags, and other materials. A four-bar linkage mechanism for the end effector provides a smooth open-close action [19]. Table 1 lists the various characteristics considered during the concept selection phase.

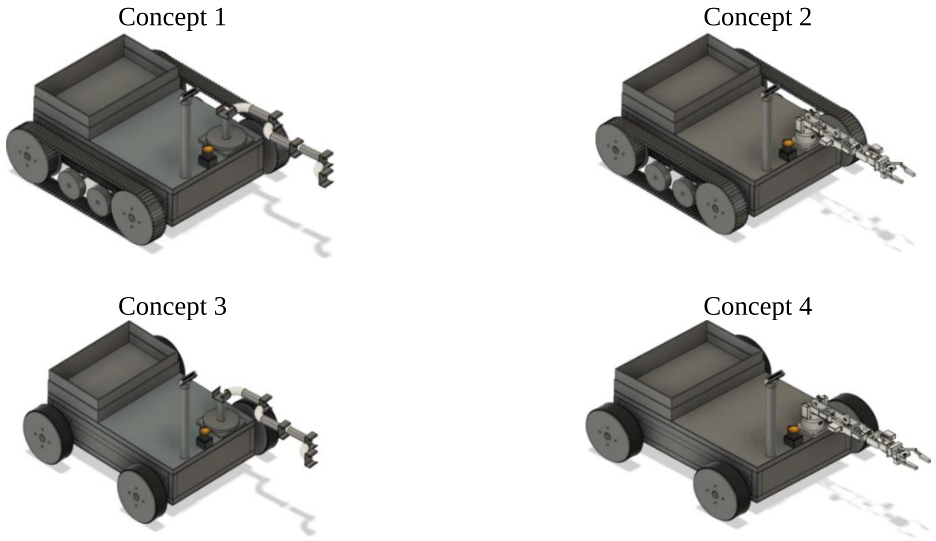


Figure 1. CAD model of the various conceptual designs

Table 1. Concept scoring and selection criteria

Selection Criteria	Conceptual Design Alternatives			
	Concept 1	Concept 2	Concept 3	Concept 4
Ease of Mobility	0.4	0.5	0.8	0.9
Manufacturability	0.3	0.4	0.5	0.6
Stability and Controllability	0.4	0.4	0.5	0.8
Weight	0.3	0.4	0.7	0.8
Workspace	0.6	0.7	0.6	0.7
Score/continue	2/No	2.4/No	3.1/No	3.8/Yes

Table 1 lists the scores corresponding to each concept design for five different parameters. These parameters include ease of mobility, manufacturability, stability and controllability, weight, and workspace. Concepts 1 and 3, from Figure 1, flaunts a six DoF cylindrical link design that focuses on easily navigating through narrow spaces. The major drawback of this design is the difficulty in clamping the motors and increased link weight over the arm. On the other hand, concepts 2 and 4 employ a six DOF rectangular link design with reduced link weight and enough sturdiness to hold the motors without much deflection. From the four design concepts, concept 4 was chosen as it had the best score out of all five parameters. The robot's design is vital to its longevity and allows for surface area allocation for various other robot parts. The design easily accommodates motor placement for movement, camera position for object recognition, mobility characteristics, and arm placement for optimal waste grasping.

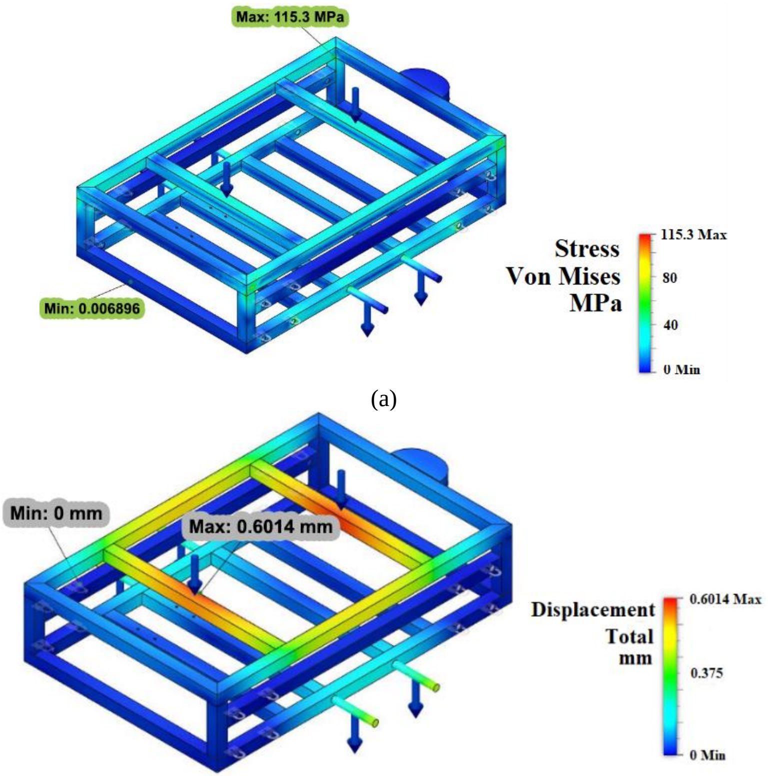
3. STRUCTURAL DESIGN AND ANALYSIS

As described in the previous section, the proposed robot consists of two sub-parts: the robotic arm and chassis. It is intended to fabricate the six DoF arm using the additive manufacturing technique to lower the overall cost of the robot while maintaining a solid structure. Polylactic acid is considered a good option as the material for the robotic arm due to its being biodegradable, having good mechanical strength and wear resistance, and being readily available [20]. The workspace of the robotic manipulator is analyzed through motion analysis in computer-aided design (CAD) software. The moment acting at the joint is maximum in its outstretched configuration. Hence, a simple moment-based approach reduces complexity in inverse dynamics. This inverse dynamic model is necessary to decide the suitable actuator for every joint. The moveable base of the robot is designed to have a cuboidal shape with a generous amount of surface area on top to house the bin, robotic manipulator, camera with its mount, laser, and other electronic components. The dimensions for the chassis are decided based on a static structural analysis performed in the FEA tool in SolidWorks. The results show that a structural cage made of Mild Steel (ASTM A36 Steel) with dimensions 1×1 inch and a thickness of 1.6 mm exhibits minimal deformation and stresses induced. Table 2 shows the material properties of ASTM A36 steel. This steel, containing 0.29% carbon per weight, is easily weldable, ductile, and strong enough to handle heavy loads.

Table 2. Material properties of ASTM A36 steel

Property	Value
Density	7.85 10^{-6} kg/mm^3
Young's Modulus	200,000 MPa
Poisson's Ratio	0.26
Yield Strength	250 MPa
Ultimate Tensile Strength	400 MPa

Figures 2(a) and 2(b) show the static structural analysis results for the chassis. Two middle beams on the chassis are subjected to a load of 900 N, while a static load of 400 N is applied on the road wheel shaft. Von Mises stress criterion is used to analyze yielding in the metal. The analysis results show that the maximum stress induced within the structure is within the safe limits of the material. This work employs the 24 V brushless direct current (BLDC) motors to drive the robot, considering its benefits such as high-power density, reduced weight, compact build, and low maintenance [21]. Six servo motors are used to provide actuation to the robotic arm. They are the MG995, MG996R, and MG959 servo motors. The advantages of these motors are their small size, lightweight build, and the provision for closed feedback control to obtain accurate motions.



(a)

(b)

Figure 2. FEA results of chassis, in the maximum and minimum (a) stresses and (b) deformation

4. ROBOT CONTROL AND SIMULATION STUDY

The simulation study of the robot control is performed on the ROS Melodic platform in an Ubuntu 18.04. ROS enables easy connection and interfacing between the various hardware. When flashed onto a low-cost single board computer like Jetson Nano, this distribution creates a portable computing platform that enables the operator to control and guide the robot, including from a remote location. Simulation of the robot in a virtual environment is necessary before implementing the design onto the appropriate hardware devices. Gazebo Simulator was chosen as the virtual environment for testing the robot code as it is easily compatible with ROS and can also emulate real-world physics [22]. Figure 3 shows the data flow diagram of the proposed waste collection robot. It illustrates how control is centralized into the Jetson Nano computer and how data flows between the Arduino and the Jetson Nano. A motor driver accepts velocity and other parameters from the Arduino controller, sent out by the ROS program on the Jetson Nano and actuates the motor [23], [24].

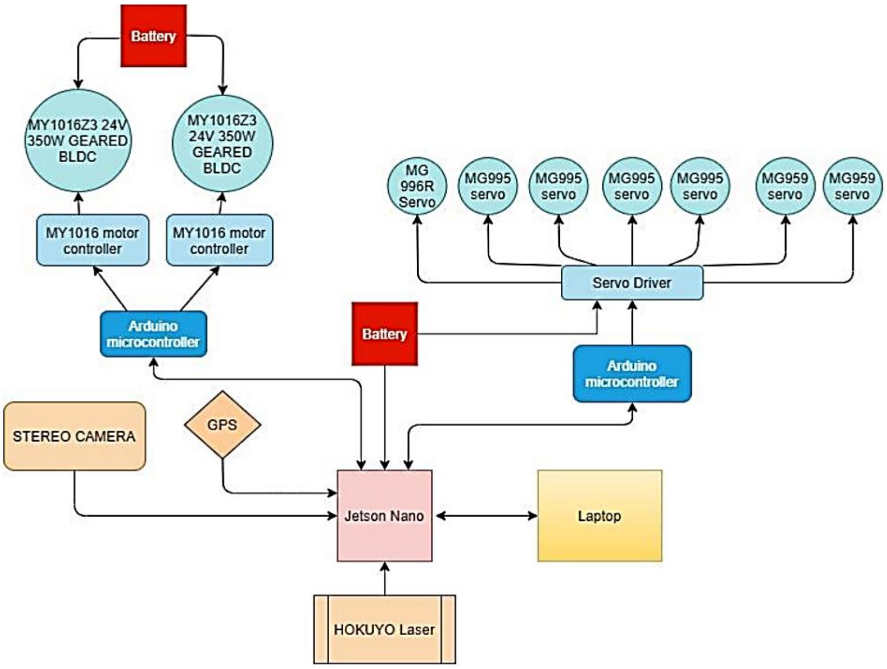
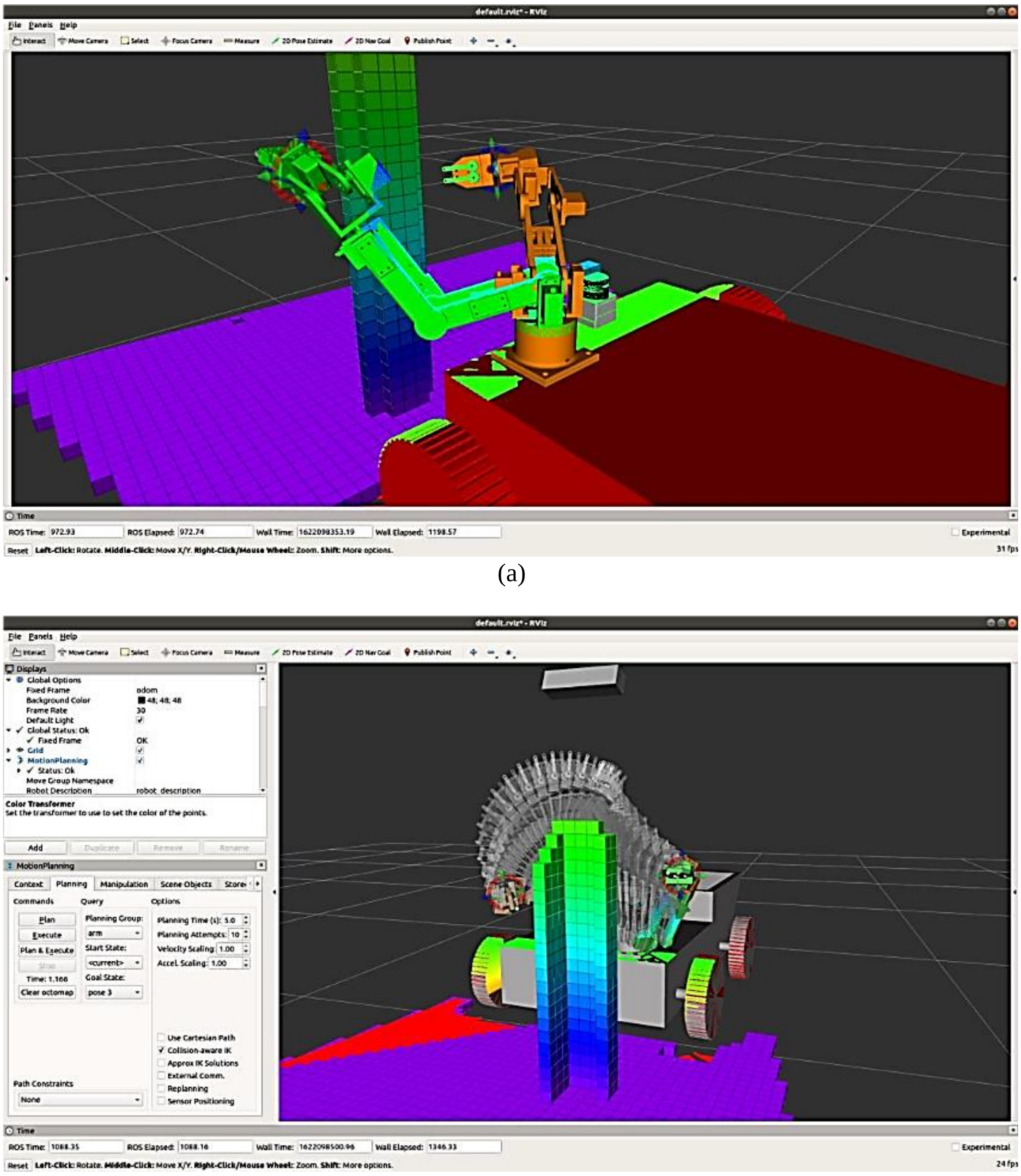


Figure 3. Data flow diagram of the waste collection robot

The ROS_control package takes input regarding the position of links from the Joint State Interface. These are passed to the Joint State Publisher, which publishes the position of the links via the joint_state topic. For creating a motion trajectory for the arm, the joint states of each joint are read by a motion planning program, and a trajectory is created for the arm to reach the goal pose. The details of joint positions to be followed are sent to the Joint Trajectory Controller, which transmits the necessary effort, position, and velocity values to move each joint in the simulated as well as a real hardware interface. Interfacing of ROS with the motors and microcontroller is established using the MoveIt Motion Planning Framework [12].

The MoveIt program, designed by PikNik Robotics, has native support in ROS and can perform complex tasks such as motion planning, collision avoidance, inverse kinematics, control, and 3D perception for robotic manipulators and grippers. The CAD model is exported to the MoveIt Setup Assistant platform as a URDF file. The robot's physical parameters, such as gravity, frictional coefficients, and collision occurrence, are computed based on the defined geometry. To generate a collision-free path, MoveIt uses the open motion planning library (OMPL) and adds a collision detection module using a 3D representation of the robot's workspace [25], [26].

The first step with this package is to calculate the self-collision matrix that checks and describes whether a link collides with every other link. The sample size for generating a collision matrix ranges from 10,000 to 100,000, with higher values enabling better collision checking. Hence, the largest sample size was chosen for self-collision checking in this study. The 3D perception module facilitates a three-dimensional vision sensor in the robot model. In this study, for simulation, the depth topic of the Kinect 3D camera is defined for obtaining point clouds from Gazebo simulated environment and connecting the same to MoveIt Rviz Visualizer. Motion planning for the manipulator is done with the help of the joint state, joint position, and joint trajectory controllers. The joint state controller fetches information regarding the current state of each joint from the hardware interface and sends them to the controller manager. MoveIt module takes the data, calculates the path to the goal pose from the current pose, creates a trajectory for the arm to follow, and commands the Joint Position Controllers to move each joint accordingly. Figure 4(a) shows the robotic arm in the MoveIt platform during motion planning. The start and goal pose of the arm are shown in green and orange colors, respectively. Figure 4(b) depicts the arm moving around a cylindrical obstacle to avoid a collision. The depth sensor aids in creating an Octomap within MoveIt to help the arm since the obstacle. The position controllers are a better option than other controllers such as effort controllers due to the heavy arm vibrations caused by the latter. Joint Trajectory Controllers use proportional integral derivative (PID) loops to send commands to the joint actuators via a Hardware Interface that communicates with the Arduino microcontroller and motor driver. The controller inputs the joint efforts, velocities, and the lower and upper limits specified in the URDF file. These values are used to operate each joint within a safe range of movement, helping prevent collision between adjacent links.


default.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Time
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.
Displays
Add Duplicate Remove Rename
MotionPlanning
Context Planning Manipulation Scene Objects Stored
Commands Query Options
Plan Execute Plan & Execute Stop Clear octomap

ROS Time: 972.93 ROS Elapsed: 972.74 Wall Time: 1622098353.19 Wall Elapsed: 1198.57 □ Experimental 31 fps

(a)

Global Options
Fixed Frame: odom
Background Color: 48; 48; 48
Frame Rate: 30
Default Light: ☑
Global Status: Ok
Fixed Frame: OK
Grid: ☑
MotionPlanning: ☑
Status: Ok
Move Group Namespace
Robot Description: robot_description

Color Transformer
Set the transformer to use to set the color of the points.

Time: 1.166
Planning Group: arm
Start State: <current>
Goal State: pose 3
Planning Time (s): 5.0
Planning Attempts: 10
Velocity Scaling: 1.00
Accel. Scaling: 1.00
Path Constraints: None

□ Use Cartesian Path
☑ Collision-aware IK
□ Approx IK Solutions
□ External Comm.
□ Replanning
□ Sensor Positioning

ROS Time: 1088.35 ROS Elapsed: 1088.16 Wall Time: 1622098500.96 Wall Elapsed: 1346.33 □ Experimental 24 fps

(b)

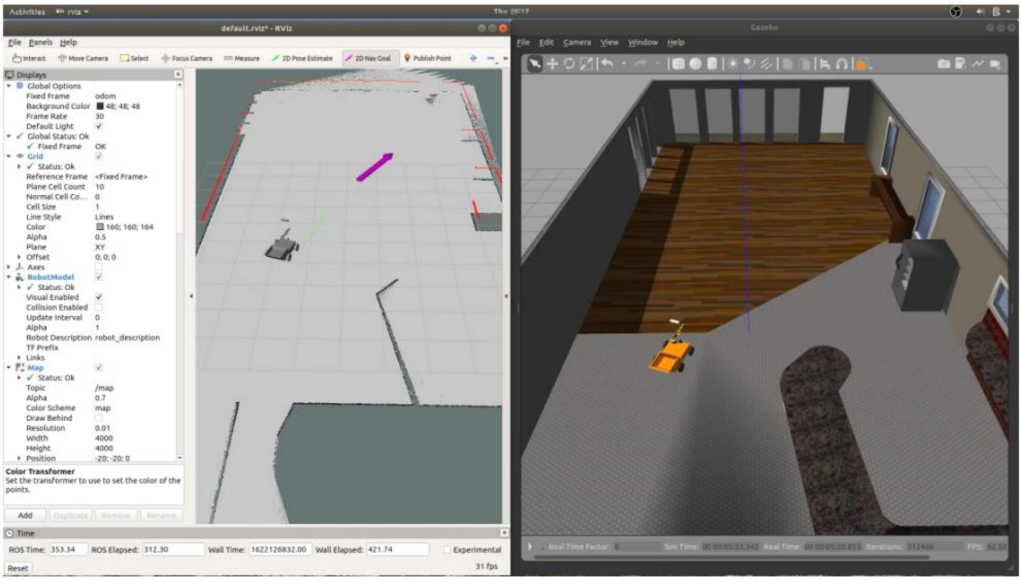
Figure 4. Motion planning for arm in MoveIt platform: (a) start and goal poses of arm and (b) obstacle avoidance of arm

Collision-aware IK ensures that the arm does not collide with any obstacle scanned by the depth sensor. R.O.S. is equipped with a navigation stack that can navigate the robot within a map. The three necessary steps involved with autonomous navigation are to create an environment map, to localize the robot within the map, and create path planning for navigation to the desired location.

The navigation stack accepts laser sensor data, environment maps from the map server, and odometry data. These are fed into various cost maps, which help fine-tune the parameters before sending a cmd_vel message to the robot base controller.

This study applies the differential drive controller compatible with the ROS navigation stack and enables easy control and steering of the vehicle using the cmd_vel ROS topic. The primary input values taken by this controller include the names of left and right wheel joints, wheel radius, track width, and base frame name. This controller uses a PID controller to receive feedback from the encoder of the left and right motors and provide commands to the same [27]. The complete robot model is spawned in the Gazebo simulator along with the desired world to be mapped. A map was created in the turtlebot_playground world from the Turtlebot robot package and built-in cafe world within Gazebo. The environments were mapped by fixing a Hokuyo laser range finder on top of the base frame of the robot, and robot movement was initiated with the help of the turtlebot_teleop ROS package, which brings up key bindings for moving the robot for map creation. The laser creates scans of its surrounding objects and passes these scans into the Rviz visualizer [28]. The mapping technique here is G mapping which utilizes Rao-Blackwellized particle filters for learning grid maps [29]. The occupancy map shows a blueprint of the objects and boundaries in the environment. This information is stored as an image file and is accompanied by a YAML file pointing to the created map. The navigation stack uses the adaptive Monte Carlo localization algorithm to localize the robot within the map. It is a technique that involves using particles, which converge as the robot moves, to identify its position and orientation [30].

Figure 5 shows the map creation of the proposed robot in the Gazebo environment. The 2D Pose Estimate tool in Rviz is used to orient the robot on the map to face the same direction as the robot in the gazebo world. The 2D Nav Goal tool, shown as a purple arrow in the figure, is used to set the direction the robot needs to move and the endpoint it needs to reach. Laser data from the Hokuyo lidar sensor, displayed as a thick red line, helps avoid obstacles newly introduced into the map [31].



Global Options
Fixed Frame odom
Background Color ■ 48; 48; 48
Frame Rate 30
Default Light ✓
✓ Global Status: Ok
✓ Fixed Frame OK
Grid ✓
✓ Status: Ok
Reference Frame <Fixed Frame>
Plane Cell Count 10
Normal Cell Co... 0
Cell Size 1
Line Style Lines
Color ▨ 160; 160; 164
Alpha 0.5
Plane XY
Offset 0; 0; 0
Axes □
RobotModel ✓
✓ Status: Ok
Visual Enabled ✓
Collision Enabled □
Update Interval 0
Alpha 1
Robot Description robot_description
TF Prefix
Links
Map ✓
✓ Status: Ok
Topic /map
Alpha 0.7
Color Scheme map
Draw Behind □
Resolution 0.01
Width 4000
Height 4000
Position -20; -20; 0

Color Transformer
Set the transformer to use to set the color of the points.

ROS Time: 353.34 ROS Elapsed: 312.30 Wall Time: 1622126832.00 Wall Elapsed: 421.74 □ Experimental
31 fps

Figure 5. Map creation and path planning simulation in Gazebo environment

Local and global cost maps store data regarding the proximity of obstacles locally and globally and aid in path planning. Common cost maps contain information common to both cost maps, such as obstacle range and inflation radius: parameters that decide how far away an obstacle must be detected and kept in memory for creating a new map. Similarly, a transformation tree (tf) shows how the coordinate frame of each link relates to the adjacent and base links. It allows information on individual coordinate frames to be viewed as a scene graph. In the case of a gripper, tf translates the coordinate position of the gripper from the viewpoint of the depth camera to that of the baseline or robot_footprint links. This way, motion planning can be easily performed, and the entire arm can be moved to grasp an object from the base onward.

Object detection is the crucial element that helps the robot identify waste objects before initiating grasping action. Several researchers have used neural networks, which can be trained to detect objects from a specific dataset. For instance, Li *et al.* [32] have used the YOLOv3 neural network to identify plastic objects in water bodies to help combat water pollution. Moreover, works by Ayoub and Schneider [33]. Have used the YOLO algorithm to successfully detect faults on power lines using a unmanned aerial vehicles (UAV). This leads us to believe that the YOLO algorithm is more than capable of easily handling various object detection tasks. For this research, several variations of the YOLO neural network were tested, and they were compared for speed, accuracy, and computational efficiency. It was observed that YOLOv4-tiny deep neural networks (DNN) having 416 layers and TensorRT support, could deliver faster detection and utilize less computing power to deliver detection results [16]. Table 3 lists the number of images used in this study to train the network. The training dataset contained several images of 7 objects: a syringe, glove, mask, bottle, plastic bag, paper cup and paper. The training was performed on an Nvidia GTX 1060 GPU, and the transfer learning technique was used to train the custom dataset using pre-trained weights. Table 4 shows the four different neural networks tested against the dataset and the obtained frames per second (FPS) when these programs were run on the Jetson Nano.

Table 3. Details of training dataset

Class Name	Number of Images
Gloves	105
Mask	80
Paper Cup	89
Plastic Bag	94
Bottle	68
Paper	57
Syringe	94

Table 4. FPS comparison of various neural networks

Network Used	Obtained FPS
YOLOv3	6
YOLOv3-tiny-prn	8
YOLOv4	13.6
YOLOv4-tiny-416	24

Figure 6 depicts the graph of loss vs the number of iterations. The training was run for 6,200 iterations, where an average total loss of 1.0529 was obtained. The training was stopped when a reasonable level of prediction accuracy, more than 60%, was observed. Using more images for the training dataset and a validation dataset may further improve detection accuracy. Figure 7 shows the bounding boxes and confidence scores of some of the trained objects detected by the YOLOv4 DNN when tested with a group of test images.

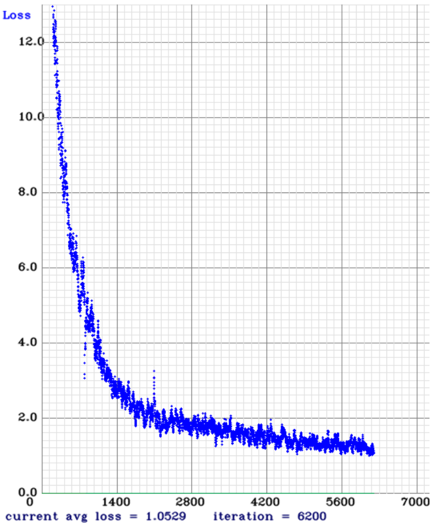


Figure 6. Loss versus iteration number during training

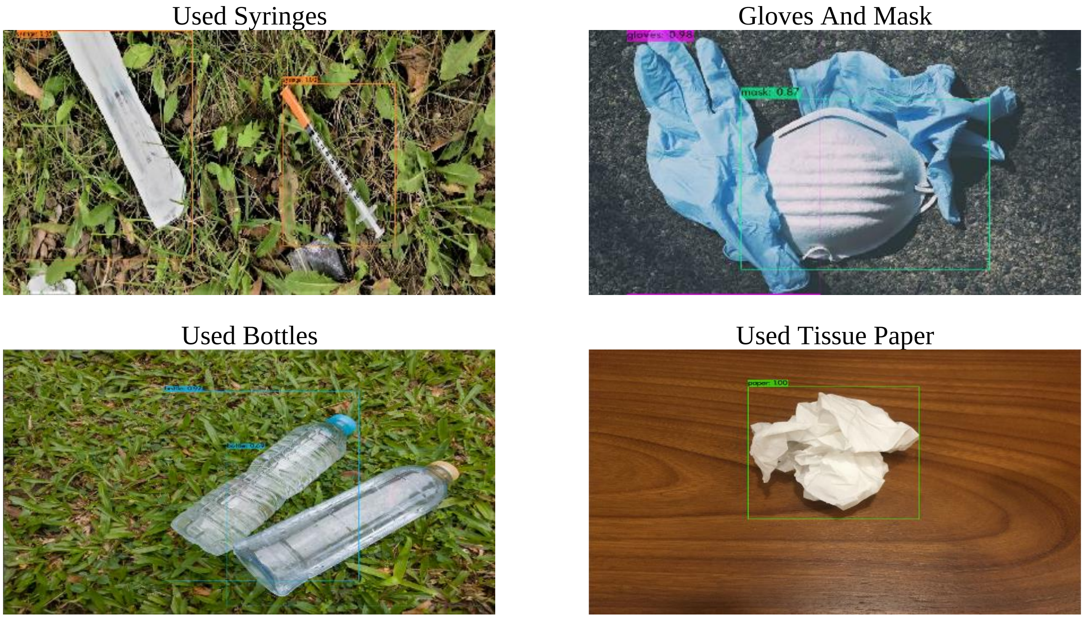


Figure 7. Result of object detection using trained YOLOv4

After the neural network detects a trained waste object, the grasping algorithm's job coupled with MoveIt motion planning is to initiate the movement of the arm to object coordinates and execute a successful grasping action before moving it to the onboard bin. The Realsense D415 stereo camera sensor attached to the robot frame can capture and enhance accurate depth data. The depth data is collected in a point cloud format, which can capture the dimensions and physical identity of the object for easy grasping. MoveIt reads this point cloud data via the pointcloud2 depth topic published by the camera. Figure 8 shows a simulated depth camera that detects primitive shapes in front of the robot in the Gazebo simulator and sends their point cloud data to Moveit [34]. The grasp pose detection algorithm analyzes the point cloud and predicts the best gripper pose for picking the object and placing it in the bin. The grasp pose detection algorithm analyzes the point cloud. It detects the narrowest area where the fingers of the gripper can accurately attach themselves without slipping and further lifts the object. The algorithm achieves a 93% success rate and can grasp objects even from dense clutter [17]. It generates all possible grasp combinations based on the object's geometry and executes a pose with the highest probabilistic success score.

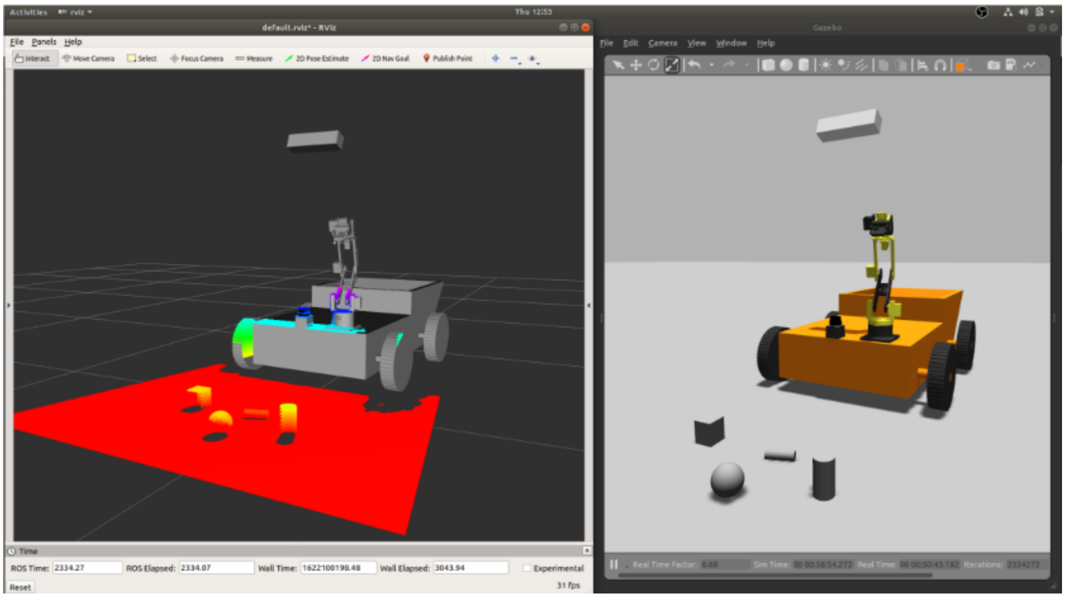
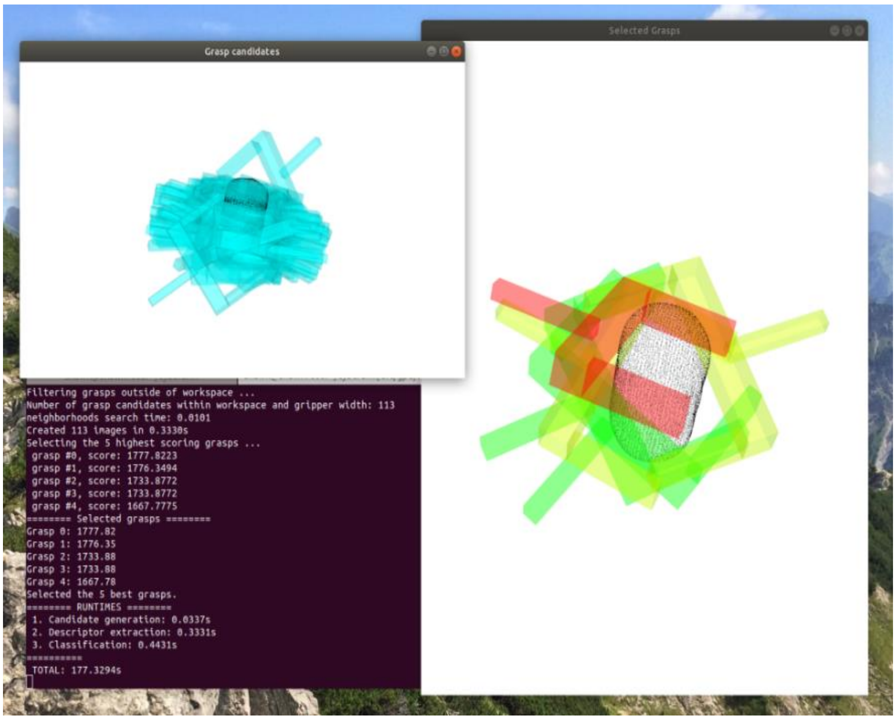


Figure 8. Primitive shapes for object detection

Figure 9 shows the five different grasp possibilities arranged from grasp0 to grasp4 from the point cloud of a cylindrical object and sends the grasping coordinates with the highest probability of grasping to Moveit. The ROS tf tree helps transform the coordinates of detected objects with respect to the grasping claw for quickly calculating the direct path to the object. The velocity requirements for the servo motors of the arm to move the gripper to the desired locations are relayed through the ros_serial server to the Arduino microcontroller, then to the motor driver, eventually actuating the motor.



```
Filtering grasps outside of workspace ...
Number of grasp candidates within workspace and gripper width: 113
neighborhoods search time: 0.0101
Created 113 images in 0.3330s
Selecting the 5 highest scoring grasps ...
 grasp #0, score: 1777.8223
 grasp #1, score: 1776.3494
 grasp #2, score: 1733.8772
 grasp #3, score: 1733.8772
 grasp #4, score: 1667.7775
======== Selected grasps ========
Grasp 0: 1777.82
Grasp 1: 1776.35
Grasp 2: 1733.88
Grasp 3: 1733.88
Grasp 4: 1667.78
Selected the 5 best grasps.
======== RUNTIMES ========
 1. Candidate generation: 0.0337s
 2. Descriptor extraction: 0.3331s
 3. Classification: 0.4431s
==========
 TOTAL: 177.3294s
```

Figure 9. Point cloud of a cylinder and grasp predictions

5. CONCLUSION AND SCOPE FOR FUTURE WORK

This paper explains the conceptual design and simulation of a waste collection robot. Several mechanical designs were generated and studied before a wheeled base was chosen for the robot. The FEA and simulations on the robot frame showed that the robot could carry loads up to 900N with very low deflection, proving that the material selection was apt. The BLDC motor selected was also adequate to drive the robot at a decent velocity in the Gazebo simulator. The servo motors for the arm were found capable of providing enough torque to perform pick and place action on the waste item.

The robot can prove helpful in places requiring additional care due to contamination or other health hazards. Its ability to be trained gives the robot greater flexibility for use in more varied fields such as construction and manufacturing for monitoring and ensuring that the work environments are safe and clean. The advent of Industry 4.0 has only accelerated the use of robotic and intelligent systems in all aspects of modern life. This robot can help push the need for a safer approach to dealing with waste.

Future navigation improvements may include using a more advanced mapping algorithm such as Hector simultaneous localization and mapping (SLAM) or real-time appearance based (RTAB) mapping to ensure better performance when navigating unknown areas. Even though this requires a more powerful computer than a single board computer such as a Jetson Nano or Raspberry Pi, this algorithm uses more reliable point cloud data for map construction. The training dataset could also include more items so that the robot will have a wide range of things that it can pick up. A waste sorting mechanism within the robot will also make it easier for the next stage of waste processing. Future works can also add additional features such as wireless and/or solar-powered charging to the robot. A vacuum suction device coupled with the robotic arm can help collect minute and lightweight waste particles, thus helping create a cleaner area. Collecting data about the frequency and type of waste observed at different locations may help the robot identify frequent areas where waste items are likely to accumulate. Similarly, further advancements can be added to enable the robot for multi-purpose applications such as medical screening of patients under isolation or sanitization of infected spaces.

REFERENCES

[1] M. Worobey, "Dissecting the early COVID-19 cases in Wuhan," *Science*, vol. 374, no. 6572, pp. 1202–1204, Dec. 2021, doi: 10.1126/science.abm4454.

[2] A. K. Das, M. N. Islam, M. M. Billah, and A. Sarker, "COVID-19 pandemic and healthcare solid waste management strategy – A mini-review," *Science of The Total Environment*, vol. 778, no. 1, Jul. 2021, doi: 10.1016/j.scitotenv.2021.146220.

[3] S. A. Sarkodie and P. A. Owusu, "Impact of COVID-19 pandemic on waste management," *Environment, Development and Sustainability*, vol. 23, no. 5, pp. 7951–7960, May 2021, doi: 10.1007/s10668-020-00956-y.

[4] H. Sugiyama, T. Tsujioka, and M. Murata, "Real-time exploration of a multi-robot rescue system in disaster areas," *Advanced Robotics*, vol. 27, no. 17, pp. 1313–1323, Dec. 2013, doi: 10.1080/01691864.2013.838333.

[5] J. Gutberlet and S. M. N. Uddin, "Household waste and health risks affecting waste pickers and the environment in low- and middle-income countries," *International Journal of Occupational and Environmental Health*, vol. 23, no. 4, pp. 299–310, Oct. 2017, doi: 10.1080/10773525.2018.1484996.

[6] A. Sardelis *et al.*, "2-Stage vision system for robotic handling of flexible objects," *Procedia CIRP*, vol. 97, no. 1, pp. 491–496, 2021, doi: 10.1016/j.procir.2020.07.008.

[7] K. Raza, T. A. Khan, and N. Abbas, "Kinematic analysis and geometrical improvement of an industrial robotic arm," *Journal of King Saud University - Engineering Sciences*, vol. 30, no. 3, pp. 218–223, Jul. 2018, doi: 10.1016/j.jksues.2018.03.005.

[8] J. Peter and M. J. Thomas, "Development of a low-cost multipurpose autonomous navigation robot," in *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, Jun. 2022, pp. 451–454. doi: 10.1109/AIC55036.2022.9848857.

[9] C. B. Kristensen, F. A. Sørensen, H. B. Nielsen, M. S. Andersen, S. P. Bendtsen, and S. Bøgh, "Towards a robot simulation framework for E-waste disassembly using reinforcement learning," *Procedia Manufacturing*, vol. 38, no. 2019, pp. 225–232, 2019, doi: 10.1016/j.promfg.2020.01.030.

[10] G. Sivasankar, B. Durgalakshmi, and K . Seyatha, "Autonomous trash collecting robot," *International Journal of Engineering Research and*, vol. V6, no. 04, pp. 162–165, Apr. 2017, doi: 10.17577/IJERTV6IS040231.

[11] A. Sengupta, V. Varma, M. S. Kiran, A. Johari, and M. R, "Cost-effective autonomous garbage collecting robot system using IoT and sensor fusion," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 1–8, Nov. 2019, doi: 10.35940/ijitee.A3880.119119.

[12] S. Chitta *et al.*, "ros_control: A generic and simple control framework for ROS," *The Journal of Open Source Software*, vol. 2, no. 20, p. 456, Dec. 2017, doi: 10.21105/joss.00456.

[13] M. Gorner, R. Haschke, H. Ritter, and J. Zhang, "MoveIt! Task constructor for task-level motion planning," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 190–196. doi: 10.1109/ICRA.2019.8793898.

[14] S. Hernandez-Mendez, C. Maldonado-Mendez, A. Marin-Hernandez, H. V. Rios-Figueroa, H. Vazquez-Leal, and E. R. Palacios-Hernandez, "Design and implementation of a robotic arm using ROS and MoveIt!," in *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Nov. 2017, vol. 2018-Janua, no. Ropec, pp. 1–6. doi: 10.1109/ROPEC.2017.8261666.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *Cornell University*, pp. 1–17, Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.10934

[17] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13–14, pp. 1455–1473, Dec. 2017, doi: 10.1177/0278364917735594.

[18] N. Bratovanov, "Robot modeling, motion simulation and off-line programming based on solidWorks API," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, Feb. 2019, pp. 574–579. doi: 10.1109/IRC.2019.00117.

[19] P. Singh, A. Kumar, and M. Vashisth, "design of a robotic arm with gripper & end effector for spot welding," *Universal Journal of Mechanical Engineering*, vol. 1, no. 3, pp. 92–97, Oct. 2013, doi: 10.13189/ujme.2013.010303.

[20] A. A. Almajid, "Mechanical, physical, and wear properties of 3D printed polyactic acid materials," *Key Engineering Materials*, vol. 833, pp. 118–125, Mar. 2020, doi: 10.4028/www.scientific.net/KEM.833.118.

[21] M. Mahmud, S. M., A. H., and A. Nurashikin, "Control BLDC motor speed using PID controller," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, pp. 477–481, 2020, doi: 10.14569/IJACSA.2020.0110359.

[22] L. Pitonakova, M. Giuliani, A. Pipe, and A. Winfield, "Feature and performance comparison of the V-REP, gazebo and ARGoS robot simulators," in *Lecture Notes in Computer Science - Springer*, vol. 10965, 2018, pp. 357–368. doi: 10.1007/978-3-319-96728-8_30.

[23] M. Vlad, P. Popov, and D. Vasile, "Simulation of arduino mega 2560 board controlling a variable frequency converter driving a BLDC motor," in *2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, Nov. 2020, pp. 1–4. doi: 10.1109/EEAE49144.2020.9278996.

[24] A. Bhargava and A. Kumar, "Arduino controlled robotic arm," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Apr. 2017, pp. 376–380. doi: 10.1109/ICECA.2017.8212837.

[25] H. Deng, J. Xiong, and Z. Xia, "Mobile manipulation task simulation using ROS with MoveIt," in *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Jul. 2017, pp. 612–616. doi: 10.1109/RCAR.2017.8311930.

[26] M. Bennulf, B. Svensson, and F. Danielsson, "Verification and deployment of automatically generated robot programs used in prefabrication of house walls," *Procedia CIRP*, vol. 72, pp. 272–276, 2018, doi: 10.1016/j.procir.2018.03.025.

[27] H. Do Quang *et al.*, "Mapping and navigation with four-wheeled omnidirectional mobile robot based on Robot Operating System," in *2019 International Conference on Mechatronics, Robotics and Systems Engineering (MoRSE)*, Dec. 2019, pp. 54–59. doi: 10.1109/MoRSE48060.2019.8998714.

[28] S. Riaz un Nabi Jafri, S. Shamim, S. Muhammad Yasir, S. Ahmed, M. Owais Ali Siddiqui, and A. Basit, "Low cost backpack scanning and mapping system for Indoor environments," in *2021 International Conference on Robotics and Automation in Industry (ICRAI)*, Oct. 2021, pp. 1–6. doi: 10.1109/ICRAI54018.2021.9651437.

[29] M. Filipenko and I. Afanasyev, "Comparison of various SLAM systems for mobile robot in an indoor environment," in *2018 International Conference on Intelligent Systems (IS)*, Sep. 2018, pp. 400–407. doi: 10.1109/IS.2018.8710464.

[30] L. Zhi and M. Xuesong, "Navigation and control system of mobile robot based on ROS," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Oct. 2018, pp. 368–372. doi: 10.1109/IAEAC.2018.8577901.

[31] M. Marian, F. Stinga, M.-T. Georgescu, H. Roibu, D. Popescu, and F. Manta, "A ROS-based control application for a robotic platform using the gazebo 3D simulator," in *2020 21th International Carpathian Control Conference (ICCC)*, Oct. 2020, pp. 1–5. doi: 10.1109/ICCC49264.2020.9257256.

[32] X. Li, M. Tian, S. Kong, L. Wu, and J. Yu, "A modified YOLOv3 detection method for vision-based water surface garbage capture robot," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, May 2020, doi: 10.1177/1729881420932715.

[33] N. Ayoub and P. Schneider-Kamp, "Real-time on-board deep learning fault detection for autonomous UAV inspections,"

Electronics, vol. 10, no. 9, May 2021, doi: 10.3390/electronics10091091.

[34] V. Vonasek, A. Vick, and M. Saska, "Motion planning with motion primitives for industrial bin picking," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–4. doi: 10.1109/ETFA.2017.8247759.

BIOGRAPHIES OF AUTHORS

Shawn Shaju is an undergraduate student pursuing a bachelor's degree in Mechanical Engineering at Rajagiri School of Engineering and Technology, Kakkanad, India. His areas of interest include robotic control systems, machine learning, and robotics operating systems (ROS). He can be contacted at mervinjoe4@gmail.com.

Thomas George is an undergraduate student pursuing a bachelor's degree in Mechanical Engineering at Rajagiri School of Engineering and Technology, Kakkanad, India. His areas of interest include robot design, optimization, and robot kinematics. He can be contacted at thomasgrg12@gmail.com.

Jithin K Francis received his bachelor's degree in mechanical engineering from Dhanalakshmi Srinivasan Engineering College (Autonomous) under Anna University in 2011. He has worked with Edutech NTTF India Pvt. Ltd., an ISO 9001-certified precision tool room cum training centre in the field of tooling, manufacturing, and tool design. He received an M.Tech. in computer-integrated manufacturing from the Federal Institute of Science and Technology under MG University in 2016. Later, he worked as a project associate (technical) at HMT MTL for the design of directing gear systems. He is currently an assistant professor in the Department of Mechanical Engineering. His areas of interest are CAD/CAM/CIM technologies, rapid prototyping, product development, and design. He can be contacted at thomasgrg12@gmail.com.

Manu Joseph received his bachelor's in mechanical engineering from Viswajyothi College of Engineering and Technology, Vazhakulam, in 2010. He completed his post-graduation in materials science and technology from the National Institute of Technology Calicut in 2013. He has more than eight years of teaching experience. His areas of academic interest are materials science and metallurgy, robotics, and robotic welding. He can be contacted at manuj@rajagiritech.edu.in.

Mervin Joe Thomas received his bachelor's degree in mechanical engineering from Mahatma Gandhi University in 2014. He did his post-graduation in machine design from the National Institute of Technology Calicut. He received Ph.D. in mechanical engineering from the National Institute of Technology Calicut in 2021. He worked as an assistant professor at the Department of Mechanical Engineering at the Rajagiri School of Engineering and Technology, Kakkanad, India, during the academic year 2021-22. He is currently a post-doctoral fellow at the Indian Institute of Technology Palakkad. His research areas include the design and development of robotic manipulators, controller design, simulation studies, and exoskeletons. He can be contacted at mervinjoe4@gmail.com.