Person-Following Robot

Machine Learning & Robot Operating System Lab

PROJECT REPORT

Submitted by

Team 02

01FE23BAR030 – Tanmay Hiremath

01FE23BAR032 – Shashank Kurtakoti

01FE23BAR034 – Prajwal S Mullolli

01FE23BAR047 – Shrigovinda T Kulkarni

Under the Guidance of:

Prof. Rakesh P Tapaskar

Prof. Vijayamahantessh kanavi

Bachelor of Engineering

IN

AUTOMATION AND ROBOTICS

KLE TECHNOLOGICAL UNIVERSITY

HUBBALLI. 2025- 2026

KLE TECH.
KLE Technological University
Creating Value
Leveraging Knowledge

ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude and appreciation to everyone who helped bring this project to a successful conclusion.

First and foremost, we want to express our sincere thanks to Prof. Vinayak N kulkarni, who oversaw our research and gave us tremendous advice and assistance. Their opinions and suggestions were really helpful in helping us to refine and improve our work.

We also like to express our gratitude to the teachers and employees of Automation and Robotics, KLE Technological University in Hubballi for their support and use of the project's resources. The development of the initiative was greatly aided by their dedication to supporting new ventures and to academic achievement.

Prof. Rakesh P Tapaskar
Prof. Vijayamahantessh kanavi

01FE23BAR030
01FE23BAR032
01FE23BAR034
01FE23BAR047

ABSTRACT

Person-following mobile robots are an important class of service robots with applications in assistive robotics, logistics, and human–robot interaction. This project presents the simulation and implementation of a vision-based person-following robot using the TurtleBot3 platform in a ROS 2 Humble environment. A human is represented by a Gazebo human model and is manually controlled within the simulation. The robot uses onboard camera data and a deep learning–based object detection model to detect and track the human in real time. Visual serving is employed to align the robot with the detected person, while a LiDAR-based safety mechanism monitors the robot's front sector and prevents forward motion in the presence of obstacles. The entire system is developed and tested in the Gazebo simulation environment, demonstrating reliable human-following behavior and safe operation without the need for physical hardware.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The rapid advancement of robotics has led to the development of autonomous mobile robots capable of interacting intelligently with dynamic environments. One important application of such robots is person following, where a robot detects and tracks a human target and autonomously moves in response to the person's motion. This capability is widely used in service robots, assistive robotics, warehouse automation, and human–robot interaction systems.

The Robot Operating System 2 (ROS 2) provides a flexible and modular middleware framework for developing intelligent robotic applications. By integrating ROS 2 with perception algorithms and real-time control strategies, autonomous behaviours such as person following can be efficiently implemented and evaluated. Simulation platforms such as Gazebo play a crucial role in validating robotic systems by allowing safe and cost-effective testing without physical hardware.

This project focuses on the simulation of a vision-based person-following robot using the TurtleBot3 platform in a ROS 2 Humble environment. A human is represented using a Gazebo human model and is manually controlled within the simulation. The robot uses onboard camera data to detect and track the human and follows the person based on visual feedback, while ensuring safe operation through sensor-based control logic.

1.2 Problem Statement

Traditional mobile robots often rely on predefined paths or manual teleoperation, which limits their ability to operate effectively in dynamic environments involving human movement. Designing a robot that can autonomously perceive, track, and follow a moving human in real time while maintaining stable and safe motion—remains a significant challenge in mobile robotics.

Problem statement

To design and simulate an autonomous mobile robot using ROS 2 that can detect, track, and follow a moving human in real time within an environment.

1.3 Objectives

The main objectives of this project are:

- To simulate a TurtleBot3 mobile robot equipped with a camera sensor using Gazebo
- To model a human as a dynamic entity within the simulation environment
- To implement a vision-based person-following algorithm using ROS 2
- To enable real-time interaction between the robot and the human model
- To analyse the robot's behaviour in a dynamic and interactive scenario

1.4 Scope of the Project

- The project is implemented entirely in simulation using ROS 2 Humble and Gazebo
- Vision-based person following is achieved using onboard camera data
- The system follows a modular ROS 2 architecture and can be extended to real hardware
- Advanced machine learning models and additional sensors can be integrated in future versions

1.5 Applications

- Service robots in hospitals, hotels, and public spaces
- Assistive robots for elderly or physically challenged individuals
- Mobile robots for warehouse support and logistics
- Surveillance and monitoring systems involving human interaction

CHAPTER 2

LITERATURE SURVEY

2.1 Overview

Person-following robots have been extensively studied in mobile robotics research. Early approaches relied on distance sensors and colour tracking, while recent systems employ computer vision and machine learning techniques such as CNN-based human detection and tracking.

ROS has become a standard platform for implementing person-following behaviours due to its modular architecture and extensive sensor support.

2.2 Related Work

Several research works have explored different techniques for implementing person-following behaviours in mobile robots. Laser-based leg detection methods utilize LiDAR sensors to identify and track human leg patterns, offering reliable performance in controlled indoor environments but struggling in crowded or cluttered spaces. RGB camera-based human tracking approaches rely on visual features such as colour, shape, or motion to follow a person; these methods are cost-effective but are highly sensitive to lighting variations and occlusions.

To improve robustness, depth sensors and RGB-D cameras have been employed to estimate the three-dimensional position of humans, enabling more accurate distance measurement and tracking. However, these sensors increase system cost and often have limited operational range. More recently, deep learning-based object detection models, such as convolutional neural networks, have been applied for person detection due to their high accuracy and adaptability to complex scenes. Despite their effectiveness, these models require significant computational resources and often demand powerful hardware such as GPUs.

As a result, many existing person-following systems become complex, hardware-dependent, and computationally expensive, making them less suitable for lightweight robots or real-time applications with limited processing capabilities. This motivates the need for simpler, modular, and simulation-based approaches that balance performance and computational efficiency.

Table 2.1 Literature Comparison Table

Author(s)	Year	Approach	Sensor Used	Limitation
Megalingam et al.	2022	Vision-based person following implemented in ROS using Gazebo simulation and YOLO-based human detection	RGB Camera, Depth Camera, LiDAR	High computational load and sensitivity to lighting conditions
Della Monica et al.	2024	ROS2 multiprocessing used to generate human-like robot motions through kinematic control	Joint encoders, motion sensors (simulated)	Not focused on mobile person-following behavior
Mayellaro	2022	Person-aware navigation achieved by integrating human detection into the ROS2 Nav2 cost map	RGB Camera	Emphasizes navigation safety rather than continuous following
Lim and Noh	2024	ROS2-based safety monitoring system using sensor data for human-robot interaction	Proximity sensors, vision sensors	Does not implement tracking or following algorithms
Nguyen et al.	2019	Vision-based object tracking used to enable service robots to follow target objects	RGB Camera	Limited robustness in dynamic and crowded environments

CHAPTER 3
SYSTEM OVERVIEW

3.1 Proposed System

The proposed system implements a vision-based person-following mobile robot using the ROS 2 framework in a simulated environment. A TurtleBot3 robot equipped with an onboard RGB camera and a 2D LiDAR sensor serves as the autonomous follower, while a human is represented using a Gazebo human model. The robot perceives the environment through camera input for human detection and LiDAR data for obstacle safety, enabling reliable and safe operation.

The system follows a modular ROS 2 node architecture, where perception, decision-making, and motion control are logically separated within the ROS ecosystem. Vision-based perception is used to detect and track the human in real time, while LiDAR-based sensing provides a safety layer that prevents collisions by monitoring obstacles in the robot's forward path. This modular design improves system scalability and allows individual components to be modified or enhanced independently. Human movement is controlled manually using keyboard teleoperation, enabling dynamic interaction and effective testing of the robot's following behaviours.

3.2 Functional Description

Initially, the TurtleBot3 robot is launched in a predefined Gazebo simulation world with its camera and LiDAR sensors enabled. Once the simulation environment is active, a human model is spawned at a specified location within the world using Gazebo's model spawning mechanism. The human is then moved manually using keyboard-based teleoperation commands, simulating realistic and dynamic human motion.

As the human moves, the robot continuously receives compressed camera images and processes them using a deep learning–based object detection algorithm to identify and track the human. The horizontal position of the detected human in the camera frame is used to compute angular velocity commands, allowing the robot to align itself toward the target. Forward motion commands are generated to enable the robot to follow the human when the path is clear.

Simultaneously, LiDAR sensor data is processed to monitor obstacles in the robot's front sector. If an obstacle is detected within a predefined safety distance, the system

overrides the forward motion command and brings the robot to a stop. The final velocity commands are published to the robot's velocity control topic (/cmd_vel), allowing the robot to follow the human smoothly while ensuring collision-free operation.

3.3 Assumptions and Constraints

To simplify system design and focus on the core functionality of person-following behaviours, several assumptions and constraints are considered. The project is implemented entirely in a simulated environment, and the human is represented using a predefined Gazebo human model, which does not fully capture complex human behaviours or appearance variations.

Although the robot includes LiDAR-based obstacle detection, the simulation environment does not model all real-world uncertainties such as sensor noise, lighting variations, or dynamic environmental disturbances. The system performance is also dependent on available computational resources, as deep learning–based perception and real-time simulation may affect responsiveness on lower-end hardware. Additionally, the system is designed for single-person tracking and does not support multiple human targets simultaneously.

CHAPTER 4

SYSTEM ARCHITECTURE

4.1 Overall Architecture

The overall system architecture of the person-following robot is illustrated in Fig. 4.1. The system follows a modular ROS 2–based design, where each functional component is implemented as an independent ROS 2 node. This modular architecture enables efficient inter-node communication, simplifies debugging and maintenance, and supports future scalability and extensibility of the system.

The Gazebo Simulation Environment serves as the virtual world in which both the TurtleBot3 robot and the human model operate. Gazebo provides realistic physics simulation, sensor data generation, and visualization of robot behavior. It interfaces seamlessly with ROS 2 by publishing sensor data and receiving actuation commands for robot motion.

The TurtleBot3 Robot Node represents the mobile robot platform and is responsible for executing low-level motion commands. It subscribes to velocity commands published on the /cmd_vel topic and converts them into wheel velocities using a differential drive controller, enabling translational and rotational motion within the simulated environment.

The Camera Sensor Node simulates the onboard RGB camera mounted on the TurtleBot3 robot. It continuously publishes image data to a ROS 2 topic, which serves as the primary perception input for human detection and tracking in the person-following system.

The LiDAR Sensor Node publishes laser scan data obtained from the robot's 2D LiDAR sensor. This data is used to detect obstacles in the robot's surroundings and provides a safety layer to prevent collisions during robot navigation.

The Person Follower Node functions as the core decision-making component of the system. It subscribes to camera image data for human detection and to LiDAR scan data for obstacle monitoring. Based on visual perception and safety constraints, it computes appropriate linear and angular velocity commands, which are published to control the robot's motion in real time.

The Teleoperation Node allows manual control of the human model using keyboard inputs. It publishes velocity commands to the human model, enabling dynamic and unpredictable motion patterns. This setup effectively evaluates the robot's ability to detect, track, and follow a moving person in real time.

Overall, sensor data flows from the camera and LiDAR nodes to the person follower node, which performs perception and control logic. The computed velocity commands are then sent to the TurtleBot3 robot node, forming a closed-loop autonomous control system.

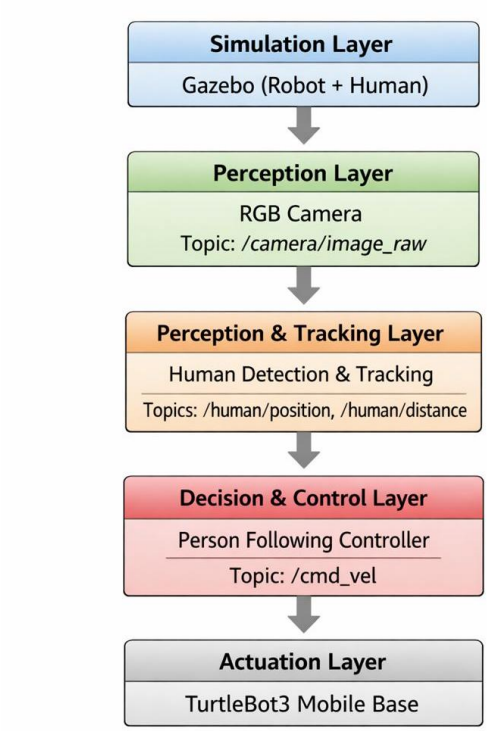


Fig. 4.1 Overall Architecture of the ROS 2–Based Person-Following Robot System

4.2 ROS Node Architecture

Node Name	Function
Gazebo Node	Simulates robot and environment
Camera Node	Publishes image data
Person Follower Node	Computes robot velocity
LiDAR Sensor Node	Publishes laser scan data for obstacle detection
TurtleBot3 Control Node	Executes velocity commands for robot motion

4.3 Communication Mechanism

The proposed system uses the ROS 2 publish–subscribe communication model to enable interaction between different functional nodes. Communication between perception, control, and actuation modules is achieved through well-defined ROS 2 topics using standard message types.

The camera sensor node publishes compressed image data on the /camera/image_raw/compressed topic using the sensor_msgs/CompressedImage message type. This topic provides real-time visual input to the person follower node for human detection and tracking.

The LiDAR sensor node publishes range data on the /scan topic using the sensor_msgs/LaserScan message type. This data is processed by the person follower node to detect obstacles in the robot's front sector and enforce safety constraints.

The /cmd_vel topic is used to control the motion of the TurtleBot3 robot. The person follower node publishes velocity commands on this topic using the geometry_msgs/Twist message type. These commands specify the robot's linear and angular velocities, enabling forward motion, rotation, or stopping based on perception and safety conditions.

The movement of the human model in the Gazebo simulation is controlled through a keyboard teleoperation node, which publishes velocity commands to the human model using geometry_msgs/Twist messages. This allows the human to move dynamically within the environment, creating realistic interaction scenarios.

Overall, this topic-based communication mechanism ensures real-time data exchange, modular system organization, and seamless coordination between sensing, decision-making, and motion control components.

CHAPTER 5
METHODOLOGY

5.1 Simulation Setup

The simulation environment is initialized by sourcing ROS 2 Humble, which configures the required environment variables and middleware for communication between nodes. The TurtleBot3 waffle_pi model is launched in the Gazebo simulator, providing a realistic mobile robot platform with differential drive kinematics, an onboard RGB camera, and a 2D LiDAR sensor.

The Gazebo simulation environment enables real-time physics simulation, sensor data generation, and visualization of robot behaviours. The camera and LiDAR sensors are activated at launch, allowing continuous acquisition of visual and range data required for person detection, tracking, and obstacle monitoring during simulation.

5.2 Human Modelling

The human is represented using a Gazebo human model rather than a simple geometric object. This model is spawned dynamically in the simulation environment at a specified location using Gazebo's model spawning mechanism. The human model provides a realistic visual target for the robot's perception system.

Movement of the human model is controlled manually within the Gazebo simulation environment using Gazebo's interactive translation tools. By dragging and repositioning the human model using the on-screen arrows, dynamic and unpredictable human motion is simulated. This approach enables effective testing of the robot's ability to detect, track, and follow a moving human target in real time without requiring an additional teleoperation node.

5.3 Person-Following Algorithm

The person-following algorithm continuously processes the compressed camera feed obtained from the robot's onboard camera. A deep learning–based object detection model is used to detect the presence of a human in each image frame. Once detected, the horizontal position of the human in the image is computed relative to the center of the camera frame.

Based on this relative position, the system estimates the angular error between the

robot's heading and the human's location. Angular velocity commands are generated to align the robot toward the human, while forward velocity commands are issued to enable the robot to follow the target. These commands are updated in real time, allowing the robot to adjust its motion smoothly as the human moves.

5.4 Control Strategy

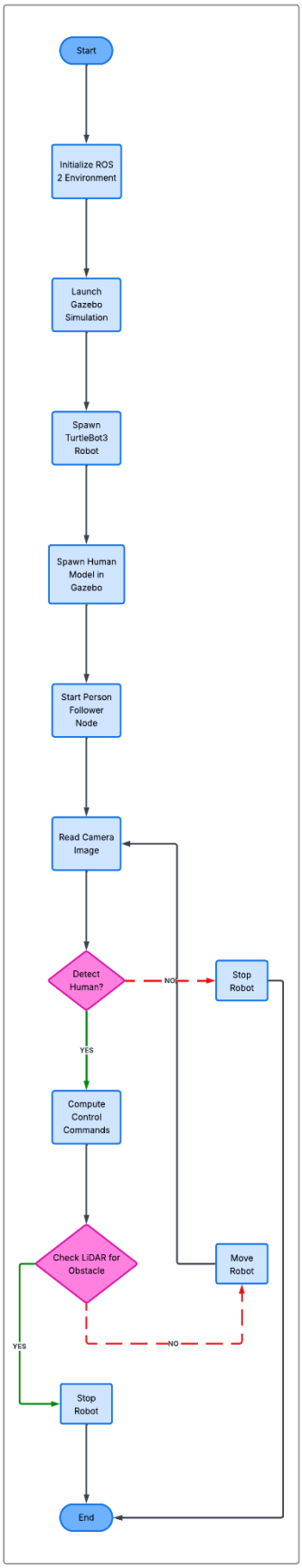
The control strategy employs a proportional controller for angular motion, where the robot's rotational velocity is adjusted based on the horizontal error between the camera center and the detected human position, ensuring smooth alignment and stable tracking. Forward motion is commanded using a constant linear velocity when the path is clear. For safety, LiDAR-based obstacle detection continuously monitors the robot's front region, and if an obstacle is detected within a predefined safety distance, the forward motion is immediately halted. This priority-based control strategy ensures stable and safe person-following behavior.

5.5 Overall system Flowchart

Figure 5.1 illustrates the overall working of the proposed person-following system. The system begins with the initialization of the ROS 2 environment followed by the launch of the Gazebo simulation. The TurtleBot3 robot and the human model are spawned within the simulation environment. The person follower node continuously captures camera images, detects the presence of a human, and computes motion commands. Simultaneously, LiDAR data is monitored to ensure safe operation. The robot follows the human when no obstacle is detected and stops when safety constraints are violated. This entire process operates in a continuous closed-loop manner.

5.6 Person Detection and Following Logic

Figure 5.2 explains the vision-based person detection and following logic. Camera images are processed using a deep learning–based object detection model to detect the presence of a human. When a human is detected, the system computes the angular error between the robot's heading and the human's position in the image frame. A proportional control strategy is applied to generate angular velocity commands, while forward velocity commands enable the robot to follow the target. If no human is detected, the robot remains stationary.



5.1 Overall System Flowchart

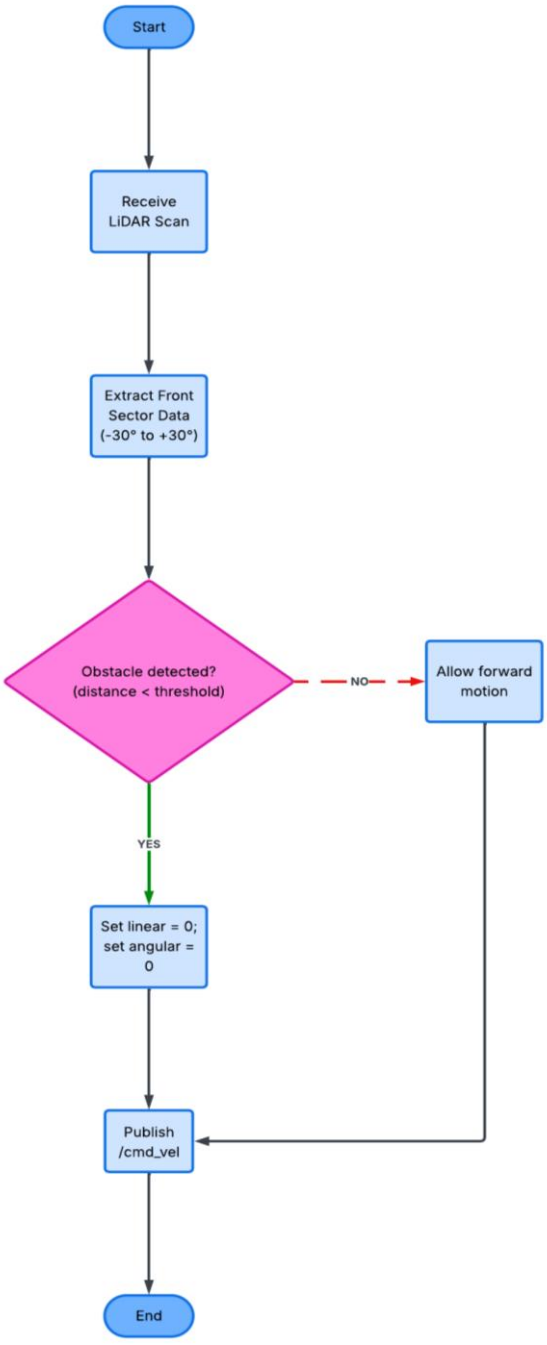


Fig 5.2 Person Detection and Following Logic Flowchart

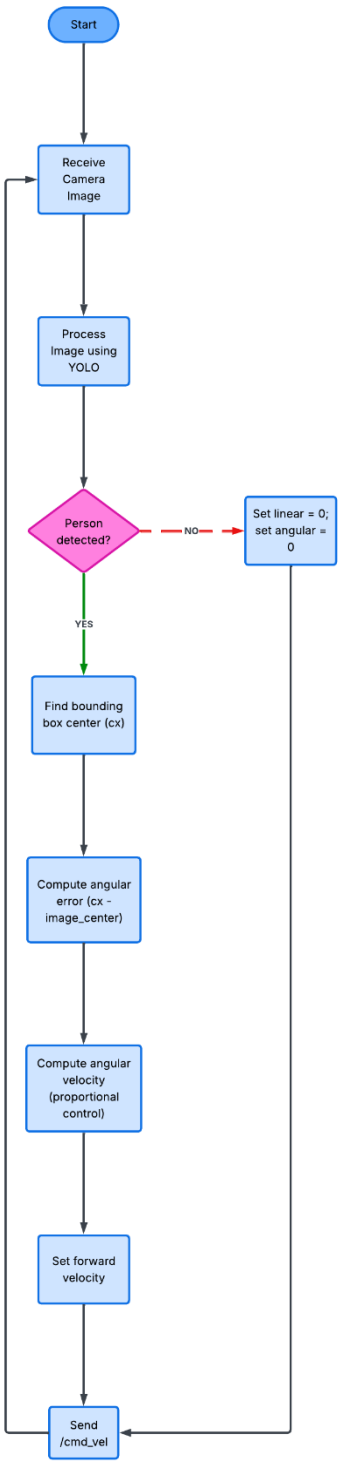


Fig 5.3 LiDAR-Based Safety Mechanism Flowchart

5.7 LiDAR-Based Safety Mechanism

Figure 5.3 describes the LiDAR-based safety mechanism integrated into the person-following system. LiDAR scan data is continuously monitored to detect obstacles in the robot's front sector. If an obstacle is detected within a predefined safety distance, the system immediately overrides the forward motion command and stops the robot. When the path is clear, the robot resumes normal person-following behavior. This safety layer ensures collision-free operation during dynamic interaction.

CHAPTER 6
IMPLEMENTATION

6.1 Gazebo and TurtleBot3 Initialization

The implementation begins with the initialization of the ROS 2 Humble environment, which sets up the required middleware and communication infrastructure. After sourcing ROS 2, the TurtleBot3 waffle_pi model is launched in the Gazebo simulation environment.

As shown in Figure 6.1, the Gazebo server and client are successfully started along with the TurtleBot3 robot. The terminal output confirms that the differential drive controller is active, odometry is being published, and essential sensors such as the camera and LiDAR are initialized correctly. This step ensures that the robot is fully operational before executing higher-level perception and control nodes.

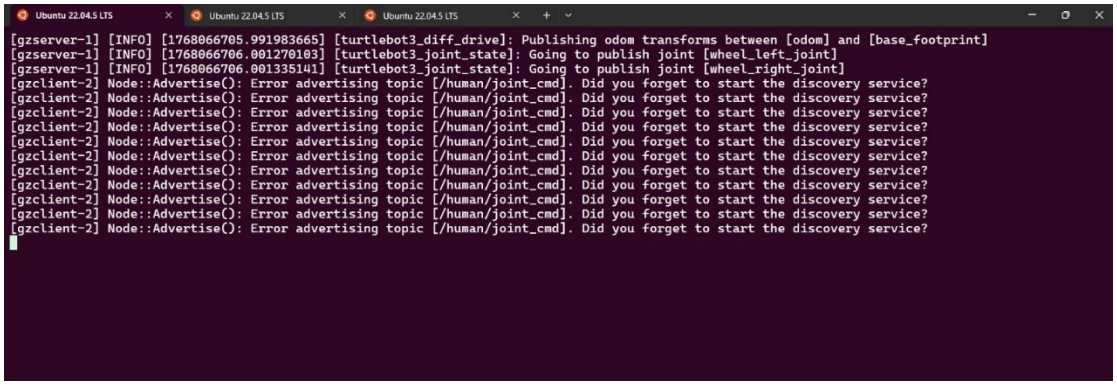


Fig 6.1 TurtleBot3 robot launched in Gazebo with active controllers and sensor interfaces in ROS2

6.2 TurtleBot3 Motion and Sensor Configuration

The TurtleBot3 robot acts as the autonomous mobile platform in the simulation. It subscribes to velocity commands on the /cmd_vel topic and converts them into wheel motions using a differential drive mechanism. Odometry data is continuously published, allowing the robot's motion to be tracked accurately in the simulation.

The onboard RGB camera provides real-time visual input for human detection, while the 2D LiDAR sensor publishes laser scan data used for obstacle detection. These sensors collectively enable perception-driven navigation and safe motion control.

6.3 Human Model Spawning in Gazebo

A human is represented using a **Gazebo human model**, which provides a realistic visual target for the robot's perception system. The human model is dynamically inserted into the simulation environment using Gazebo's spawn_entity.py mechanism. **As illustrated in Figure 6.2**, the terminal output confirms the successful spawning of the human model at a specified location within the Gazebo world. This dynamic spawning allows the human model to be added at runtime without restarting the simulation.

Movement of the human model is performed manually within the Gazebo environment using the interactive translation arrows. By repositioning the human model during simulation runtime, dynamic human motion is simulated, enabling effective testing of the robot's tracking and following behavior.

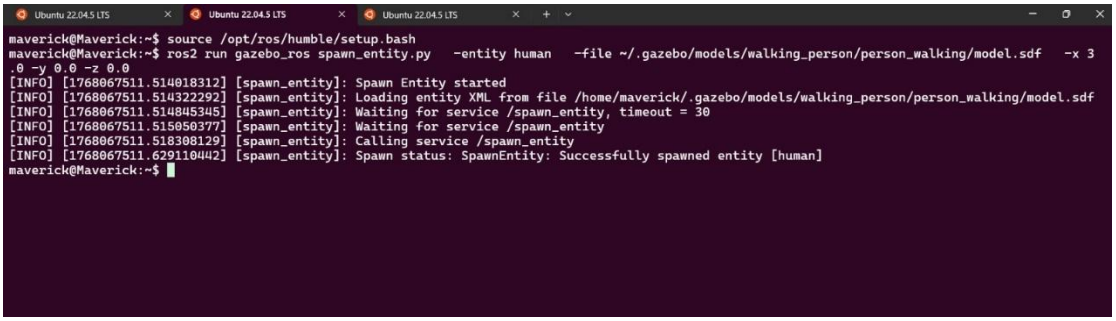


Figure 6.2: Spawning of the Gazebo human model using ROS 2 spawn_entity utility in ROS2

6.4 Execution of the Person Follower Node

The person-following functionality is implemented in the Person Follower Node, which is executed after the robot and human models are initialized. This node subscribes to the compressed camera image topic published by the TurtleBot3 camera.

The node continuously processes incoming camera frames to detect the presence of a human. When a human is detected, the horizontal position of the detected person in the

image frame is calculated. Based on this information, angular velocity commands are generated to align the robot toward the human, and forward velocity commands are issued to enable the robot to follow the target.

If no human is detected in the camera frame, the robot remains stationary by publishing zero velocity commands.

6.5 LiDAR-Based Safety Mechanism

To ensure safe operation, a LiDAR-based safety mechanism is integrated into the person follower node. The node subscribes to the /scan topic and continuously monitors LiDAR range data in the robot's front sector.

As shown in Figure 6.3, the terminal logs display real-time distance measurements obtained from the LiDAR sensor. When an obstacle is detected within the predefined safety distance, the system immediately overrides forward motion commands and stops the robot. This confirms the correct functioning of the LiDAR-based safety logic during execution.

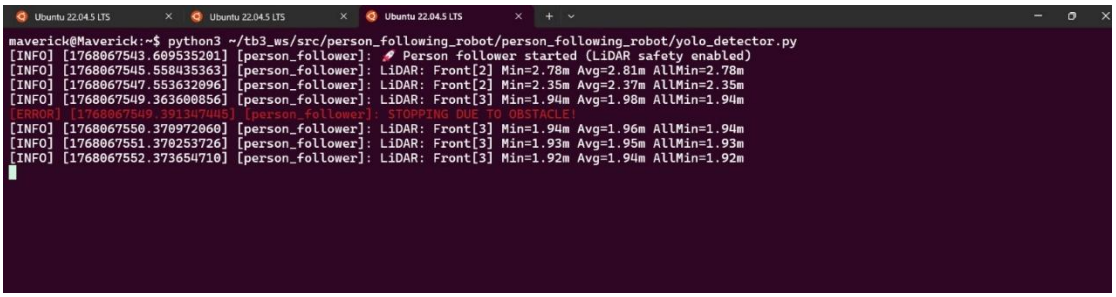


Figure 6.3: Execution of the person follower node showing LiDAR-based obstacle detection and safety stop in ROS2

CHAPTER 7

RESULTS AND DISCUSSION

7.1 Simulation Results

The simulation results demonstrate the successful operation of the proposed person-following robot system in a ROS 2 Humble and Gazebo environment. The TurtleBot3 robot is able to detect, track, and follow a moving human model in real time using camera-based perception and velocity control.

As shown in Fig. 7.1, the robot successfully detects the human using the onboard camera, indicated by a bounding box around the person and the "FOLLOWING ENABLED" status. The robot continuously adjusts its angular velocity to align itself with the human's position while maintaining forward motion when the path is clear.

Fig. 7.1Camera-based person detection and following enabled in Gazebo environment

When the human moves closer to the robot or enters a predefined safety region, the LiDAR-based safety mechanism is activated. This behaviour is illustrated in Fig. 7.2, where the system detects an obstacle within the safety distance and immediately stops the robot, as indicated by the “OBSTACLE DETECTED – STOPPED” status. This confirms the effectiveness of the integrated LiDAR-based collision avoidance mechanism.

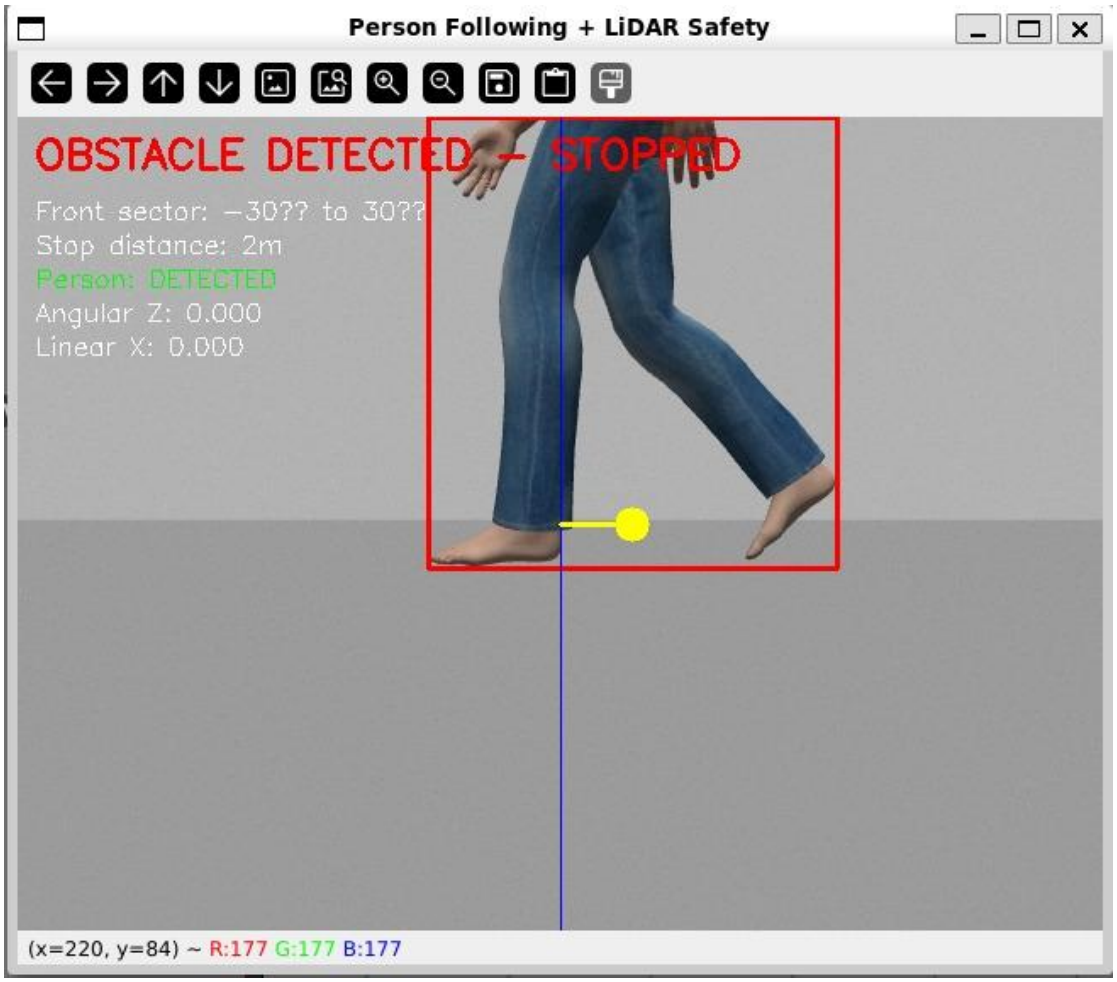


Fig. 7.2 Robot stopped due to LiDAR obstacle detection in Gazebo environment

The overall interaction between the robot and the human model within the Gazebo simulation environment is shown in Fig. 7.3. The figure highlights the robot following the human while continuously monitoring its surroundings using LiDAR. The robot maintains a safe following distance and exhibits stable behavior throughout the simulation

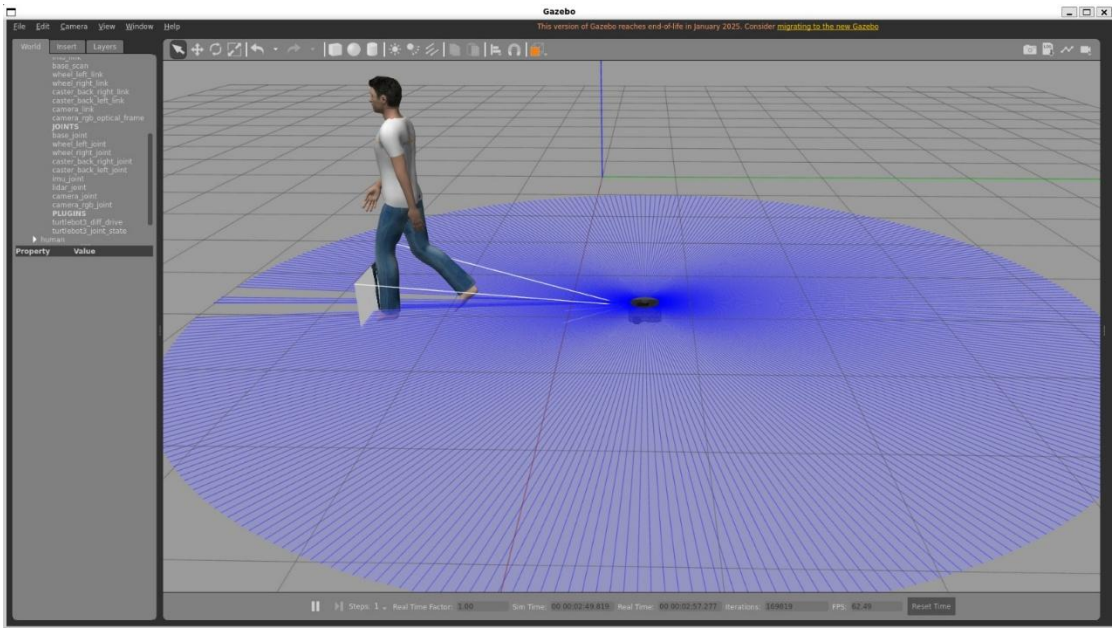


Fig. 7.3 TurtleBot3 following human model in Gazebo environment

CHAPTER 8
CONCLUSION AND FUTURE WORK

8.1 Conclusion

This project successfully demonstrates the design and simulation of a person-following mobile robot using ROS 2 Humble in a Gazebo simulation environment. A TurtleBot3 robot equipped with a camera sensor was used to detect, track, and follow a moving human model in real time. Vision-based perception enabled the robot to estimate the relative position of the human, while proportional control logic was applied to generate appropriate linear and angular velocity commands.

In addition, a LiDAR-based safety mechanism was integrated to ensure collision-free operation. The robot reliably stopped when an obstacle was detected within a predefined safety distance and resumed motion when the path was clear. The modular ROS 2 node-based architecture enabled efficient communication between perception, control, and actuation components, simplifying system development and debugging.

The simulation results confirm that the proposed system performs reliably in a controlled environment and highlight the effectiveness of ROS 2 as a flexible and robust platform for developing autonomous mobile robot behaviours.

8.2 Future Enhancements

Several enhancements can be incorporated to improve the system's capability and robustness. Advanced deep learning–based human detection and tracking models can be integrated to improve performance under varying lighting conditions, partial occlusions, and complex environments. Depth estimation techniques or sensor fusion using RGB-D cameras can be used to achieve more accurate distance control.

The system can be extended to include full obstacle avoidance and path planning to enable safe navigation in cluttered and dynamic environments. Support for multi-person tracking would allow more complex human–robot interaction scenarios. Finally, deploying the developed system on real TurtleBot3 hardware would enable validation under real-world sensing noise, environmental uncertainties, and physical constraints.

REFERENCES

[1] R. K. Megalingam, R. Anandu, H. D. Teja Anirudh Babu, S. Ghali and V. S. Y. Avvari, “Implementation of a Person Following Robot in ROS-Gazebo platform,” in *Proc. 2022 Int. Conf. for Advancement in Technology (ICONAT)*, Goa, India, Jan. 2022, pp. 1–5, doi: 10.1109/ICONAT53423.2022.9726010.

[2] D. Della Monica “Robot Making Human-Like Actions for Social Interaction: Going Deeper into ROS2 Multiprocessing and Kinematics,” in *Advances in Intelligent Systems and Computing*, Springer, 2024, pp.

[3] A. Mayellaro, “Person-Aware Autonomous Navigation for an Indoor Sanitizing Robot in ROS2,” M.S. thesis, Politecnico di Torino, 2022.

[4] D.-W. Lim and J. Noh, “ROS 2 Application to the Safety Monitoring of Collaborative Robots,” in *Proc. 2024 4th Int. Conf. on Robotics, Automation and Artificial Intelligence*, Dec. 2024, pp. 42–46.

[5] I. K. E. Purnama, M. A. Pradana and Muhtadin, "Implementation of Object Following Method on Robot Service," 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 2018, pp. 172-175, doi: 10.1109/CENIM.2018.8710819. keywords: {Cameras;Robot kinematics;Robot vision systems;Senior citizens;Turning;HOG;depth;Turtlebot;Object Following;human detecting},