

Article

Selected Motion Control Problems for a Robotic Guide Dog

Artur Babiarz , Krzysztof Jaskot  and Maksymilian Grabowy

Department of Automatic Control and Robotics, Silesian University of Technology, 44-100 Gliwice, Poland; krzysztof.jaskot@polsl.pl (K.J.); mg300365@student.polsl.pl (M.G.)

* Correspondence: artur.babiarz@polsl.pl

Abstract

This paper presents the design and implementation of a vision-based system enabling a robot to autonomously follow a human in dynamic environments. The proposed solution was developed using the quadruped walking robot Unitree Go2, which provides high mobility, terrain adaptability, and dynamic stability suitable for indoor and outdoor operation. The perception subsystem is based on a modified vision setup utilizing the ZED2i stereo camera. The camera enables real-time acquisition of RGB images and depth information, allowing for three-dimensional human detection and localization. Depth data are used to estimate the relative position and distance of the tracked person with respect to the robot, ensuring robust target tracking even in partially cluttered environments. The computational layer is implemented on the embedded platform NVIDIA Jetson Orin Nano, which provides sufficient GPU-accelerated processing power for real-time image analysis and decision-making. The control architecture is built upon ROS 2, enabling modular system integration, efficient inter-process communication, and scalability. Dedicated ROS 2 nodes handle image acquisition, human detection, depth processing, trajectory generation, and motion control. The developed system integrates perception and locomotion control to generate smooth and stable motion commands that allow the robot to maintain a safe and consistent distance from the human target. The overall architecture emphasizes real-time performance, modularity, and robustness, making it suitable for applications such as personal assistance, inspection, and collaborative robotics.

Keywords: mobile robots; walking robots; vision system; robot operating system

1. Introduction

Mobility and independence are essential aspects of daily life, yet navigating the world can present significant challenges for people with visual impairments. For decades, guide dogs have played a vital role in helping individuals travel safely and confidently, offering not only physical assistance but also companionship and emotional support. Despite their benefits, guide dogs require extensive training, substantial financial resources, and ongoing care, making them inaccessible to many people who could benefit from their assistance. Recent advances in robotics, artificial intelligence, and sensor technology have led researchers and engineers to explore new solutions for mobility assistance. One of the most promising developments is the robotic guide dog—a sophisticated machine designed to help visually impaired users navigate their surroundings. Equipped with cameras, sensors, GPS systems, and intelligent navigation software, robotic guide dogs can identify obstacles, plan safe routes, and provide real-time guidance to users [1]. Unlike traditional guide dogs, these robotic systems do not require feeding, veterinary care, or years of specialized training,



Academic Editor: Mónica Ballesta Galdeano

Received: 16 June 2026

Revised: 18 July 2026

Accepted: 21 July 2026

Published: 1 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

potentially making mobility assistance more widely available. The emergence of robotic guide dogs represents a significant step forward in assistive technology. By combining the reliability of advanced engineering with the practical functions of guide animals, these devices have the potential to improve accessibility and independence for millions of people worldwide. However, questions remain regarding their affordability, reliability, user acceptance, and ability to operate effectively in complex real-world environments. As technology continues to evolve, robotic guide dogs may transform the way visually impaired individuals interact with and navigate the world around them. This article examines the development of robotic guide dogs and their key features and applications, as well as the opportunities and challenges associated with their adoption.

Robotic technology is rapidly transforming the way humans receive assistance in both everyday life and emergency situations. From helping visually impaired individuals navigate safely to supporting rescue teams in locating missing people, robotic assistance dogs are emerging as innovative tools that combine mobility, intelligence, and real-time decision-making. Inspired by the abilities of traditional service and rescue dogs, these machines use advanced sensors, artificial intelligence, cameras, GPS, and autonomous navigation systems to perform tasks that can improve safety, independence, and efficiency [2].

For people with visual impairments, robotic guide dogs offer a promising alternative to traditional guide dogs, which are often expensive, time-consuming to train, and unavailable to many who need them. These robotic companions can detect obstacles, plan safe routes, and provide navigation assistance without requiring the extensive care and training associated with live animals. By increasing accessibility to mobility support, robotic guide dogs have the potential to enhance independence and quality of life for millions of users worldwide.

Recent advances in robotics [3], artificial intelligence, computer vision, and autonomous navigation have accelerated the development of robotic systems designed to assist visually impaired individuals and support human guidance tasks. Researchers have explored a wide range of solutions, from wheeled guide robots and quadrupedal robotic dogs to intelligent navigation platforms capable of operating in both indoor and outdoor environments. One notable approach involves a hybrid wheel–leg mobility platform developed to assist elderly and visually impaired users during navigation tasks [4]. Unlike conventional guidance robots that rely solely on wheeled or legged locomotion, this system combines both mechanisms to improve versatility. The design incorporates front wheel–leg modules alongside a stable wheel arrangement, enabling efficient movement on flat surfaces while maintaining the ability to negotiate uneven terrain and obstacles. The hybrid architecture seeks to balance energy efficiency, stability, adaptability, and affordability, making it suitable for everyday mobility assistance. The growing integration of artificial intelligence and industrial automation has also stimulated research on robotic dogs and other biomimetic platforms. Since accurate movement control is essential for these systems, researchers have investigated robust navigation and positioning methods based on binocular vision [5]. By employing optical CCD stereo cameras, dynamic tracking algorithms, and kinematic modeling, these systems continuously estimate the robot's position and orientation. Adaptive correction mechanisms allow the robot to compensate for disturbances and maintain precise motion control, resulting in improved tracking accuracy and reliable navigation performance. Inspired by autonomous driving technologies, several studies have proposed robotic guide dogs capable of following designated pedestrian pathways [6]. Traditional mobility aids such as white canes and guide dogs provide valuable assistance but often involve infrastructure limitations or significant maintenance costs. To address these challenges, researchers have developed autonomous trail-following robots that utilize deep convolutional neural networks trained on both simulated and real-world datasets. These systems can recognize navigable paths under

varying lighting conditions and environmental appearances, thereby improving navigation reliability for blind and visually impaired users.

A central and demanding challenge in this application is enabling the robot to track the person it is assisting, so it can maintain a safe and comfortable navigation experience by keeping the user within a designated zone through feedback mechanisms [2,7]. Various sensor configurations for detecting and tracking humans in autonomous robots have been proposed in the recent literature, most commonly RGB(-D) cameras [8,9], 2D LiDARs [10–12], and 3D LiDARs [13]. Relative to RGB(-D) cameras and 3D LiDAR, 2D LiDAR sensors stand out for their low power draw and broad angular coverage, delivering accurate distance readings while requiring lightweight processing, making them well-suited to embedded systems and cost-sensitive applications [14,15].

Indoor navigation has also received considerable attention. One proposed solution combines unmanned ground vehicles with passive ultra-high-frequency RFID technology to guide users toward predefined destinations within public facilities [16]. RFID tags installed at important locations store information that can be retrieved by the robot and converted into spoken instructions. The system further incorporates ultrasonic sensors, grayscale sensors, voice interfaces, and route-recording capabilities, providing an affordable and user-friendly navigation aid for indoor environments. Another line of research focuses on physical human–robot interaction. An indoor guidance robot was developed to escort visually impaired individuals by maintaining direct hand contact through a robotic arm [17]. Human hand detection is achieved using computer vision techniques, while a closed-loop controller regulates the arm’s motion. Safe navigation is ensured through reinforcement-learning-based path planning combined with obstacle avoidance strategies using depth sensing and fuzzy logic. Experimental evaluations demonstrated that the robot could successfully escort users to designated destinations without collisions. Researchers have also sought to replicate the intuitive guidance behavior of biological guide dogs while preserving the consistency and reliability of robotic systems [18]. A physically interactive hybrid guidance framework integrates compliant motion control with user-specific customization features, allowing the robot to interpret user intentions and provide adaptive assistance. Experimental results indicate that this approach can deliver stable guidance while maintaining responsiveness in dynamic environments. Perception capabilities constitute another critical component of robotic guide systems. To enhance environmental awareness, researchers have integrated robotic guide dogs with deep-learning-based object detection frameworks [19]. By combining onboard sensing technologies with the YOLO-GUIDE architecture, these systems can identify, classify, and localize indoor objects in real time, thereby improving situational awareness and navigation safety.

Prior work includes leash-guided systems [7,20] and handle-guided systems [21] designed to navigate indoor spaces and uneven terrain. Researchers have also started tackling the complexities of human–robot interaction [22,23] by building interaction models and force-responsive control mechanisms. Several recent studies have run user trials with blind and visually impaired (BVI) participants to explore how robot dog guides could be designed and what potential they hold. Wang et al. found that quadrupedal robots face usability and trust issues not seen with wheeled robots [24]. Due examined how BVI individuals experience robot guides versus conventional guide dogs, pointing out differences in how they assist with navigation and mobility [25]. The authors of [26] looked into handler–dog guide interactions and found a clear need to personalize robot designs. Hata et al. went further, testing industrial robots as dog guide substitutes and building user-friendly interfaces like voice-controlled apps [27]. Together, these studies provide valuable guidance for designing robot dog guides, with particular emphasis on user experience and effective communication between humans and robots. One proposed

framework allows the robot–human pair to dynamically alter their spatial configuration when traversing narrow passages [28]. Through a hybrid interaction model that considers leash tension and slack states, the system generates safe navigation trajectories while maintaining communication between the robot and the user. Experimental validation using a quadrupedal robotic platform demonstrated the feasibility of this approach in confined spaces. Robotic guidance technologies have also been explored beyond assistive mobility applications. For example, tour-guide robots have been developed to escort visitors through museums and exhibition spaces. These systems employ stretchable arm interfaces to maintain comfortable physical interaction while adapting their speed to the user’s walking pace [29]. Experimental studies showed that maintaining an appropriate distance between the robot and the user improves comfort and allows visitors to focus on surrounding exhibits rather than the robot’s movements. Advances in autonomous locomotion have further improved the capabilities of quadrupedal robotic dogs. Researchers have enhanced navigation performance through the integration of model predictive control, whole-body control, depth-camera-based perception, and advanced trajectory-planning algorithms [30]. Such systems generate detailed three-dimensional representations of their surroundings and can adapt their movement strategies in response to changing terrain conditions, resulting in greater stability and maneuverability. Recognizing the importance of user-centered design, some studies have conducted interviews with guide-dog users, trainers, and visually impaired individuals to identify desirable features for robotic guide systems [31]. These investigations revealed the need for customizable speed control, directional guidance, and collaborative navigation mechanisms. To address these requirements, semantic-aware path-planning approaches have been proposed, enabling robots to consider environmental context, user position, and intended travel direction when generating safe and efficient routes. User acceptance remains a key factor in the adoption of robotic guidance technologies. Evaluations involving blind and visually impaired participants interacting with quadrupedal robots revealed preferences for rigid handles, smooth turning behavior, explainable navigation decisions, and personalized control settings. Participants also emphasized practical concerns such as battery life, maintenance requirements, and operation under varying weather conditions [32]. Among the most comprehensive projects in this field is the blind-assistive autonomous droid device (BUDD-e), which aims to emulate the functionality of a guide dog through a combination of autonomous navigation and a smart tether system. A major challenge addressed by this project is the detection of sudden user stops to prevent excessive pulling forces [33]. Researchers developed a time-domain threshold-based approach that analyzes tether signals and identifies stopping events with high reliability. Experimental results highlighted tether force measurements as the most informative indicator for distinguishing between normal walking and stopping behavior. Subsequent work on the BUDD-e platform focused on the design, implementation, and real-world evaluation of a complete robotic guide prototype [34]. Testing conducted with visually impaired volunteers demonstrated encouraging levels of navigation performance, safety, and user acceptance, indicating the potential of robotic guide systems as viable alternatives or complements to traditional guide dogs. Overall, the existing body of research demonstrates substantial progress toward intelligent robotic guidance systems. Through advances in autonomous navigation, computer vision, human–robot interaction, semantic path planning, and adaptive control, robotic guide dogs are becoming increasingly capable of providing safe, reliable, and user-centered assistance for visually impaired individuals in a wide variety of environments.

At the same time, similar robotic technologies are being developed for search-and-rescue operations [35–37]. In disasters such as earthquakes, floods, or building collapses, robotic assistance dogs can enter dangerous environments that may be unsafe for human rescuers.

Equipped with thermal imaging, sound detection, and advanced mapping systems, they can search for trapped or missing people, relay critical information to rescue teams, and help speed up life-saving operations [38]. Their ability to operate in hazardous conditions makes them valuable partners in emergency response efforts.

The rapid advancement in robotics and artificial intelligence has created new opportunities for developing intelligent assistance systems that can enhance human mobility, safety, and independence. Robotic assistance dogs represent a promising application of these technologies, combining autonomous navigation, environmental perception, and human–robot interaction to support users in a variety of scenarios. From guiding visually impaired individuals through complex urban environments to assisting rescue teams in locating victims during emergency operations, these robotic platforms aim to extend human capabilities and provide reliable support where it is most needed.

Despite the promise this technology holds, the research community has yet to thoroughly explore how a robotic guide dog should be designed, and many factors remain largely unaddressed. A straightforward approach would be to replicate an animal guide’s behavior as closely as possible; however, identical design choices can produce very different user experiences, since the underlying sensing and actuation mechanisms differ fundamentally from those of a living animal. Furthermore, any single design decision typically carries multiple trade-offs at once, which makes it hard for a designer to weigh one option against another. Take gait controllers as an example: model-based optimization and learning-based methods—two widely used approaches—differ in foot clearance, step length, noise output, and how they recover balance, each with its own strengths and weaknesses [39,40]. Robots can also be given capabilities that go beyond what any animal could offer [41–43], such as language-based communication powered by large language models (LLMs) [44,45]. This not only distinguishes them from animal guides but also creates room for entirely new kinds of innovation. A notable illustration of how these methods keep evolving, along with the innovations being built on top of them, is a study [46] introducing a Perception–Awareness–Decision (PAD) framework. Rather than simply stacking components together, it systematically integrates Simultaneous Localization and Mapping (SLAM) with Vision–Language Models (VLMs), speech recognition, and large language models (LLMs). The resulting system builds a situational map that helps a robot moving through crowds decide whether to keep navigating, pause, or initiate a conversation.

2. Problem Statement

Based on the information and research results reported in the literature, the authors of this paper have concluded that the solutions presented for the use of dog-type robots in collaboration with humans are most often oriented toward a single, precisely defined application [1]. For this reason, the main goal of the authors is to build their own control system for a selected mobile platform in the form of a four-legged (dog-type) robot that would ensure versatility of application for the three most popular tasks:

1. Assistive robot.
2. Rescue robot.
3. Guide-type robot.

These are very similar tasks, but they require operation under highly specific environmental conditions.

In this article, the authors present the first research results regarding specific assumptions that will allow for the implementation of basic activities related to the first two tasks mentioned above. The main assumptions regarding the entire system are as follows:

- Universality, consisting in using readily programmable, commercially available embedded systems to build the control system.

- Movement on four legs, which guarantees ease of overcoming uneven terrain.
- The use of appropriate sensors that will ensure recognition of a specific shape and following it.
- Openness of the hardware solution, which will enable rapid future expansion according to specific needs arising from the tasks set.

Our contributions are as follows:

- Developing the concept and construction of a proprietary control system based on commercially available components.
- A simple control scheme based on visual information, enabling the tracking of and following a human at a fixed distance and tolerance (sensitivity zone).
- The identification of diverse and complex problems, based on the conducted tests, that need to be solved in order to accomplish task 3.

It is important to distinguish that the current implementation demonstrates a baseline person-following capability utilizing a distinct visual marker. Full robotic guide-dog functionality necessitates advanced user guidance interfaces, autonomous obstacle aware semantic navigation, rigorous safety validation and specialized physical interaction mechanisms for visually impaired users remains outside the scope of this preliminary study. The present system serves as a perception and kinematic basis for these future functionalities.

Additionally, presented project focuses on exploring the potential of the Unitree GO2 Edu quadruped robot as a foundation for developing advanced robotic assistance systems. The primary objective was to investigate the robot's capabilities and implement a solution that enables autonomous human-following behavior, a fundamental requirement for future guidance and assistance applications. The development process utilized the ROS 2 Humble robotics framework together with the NVIDIA Jetson Orin Nano computing platform, providing the necessary infrastructure for perception, control, and autonomous navigation.

3. Concept and Implementation of a Navigation and Location System

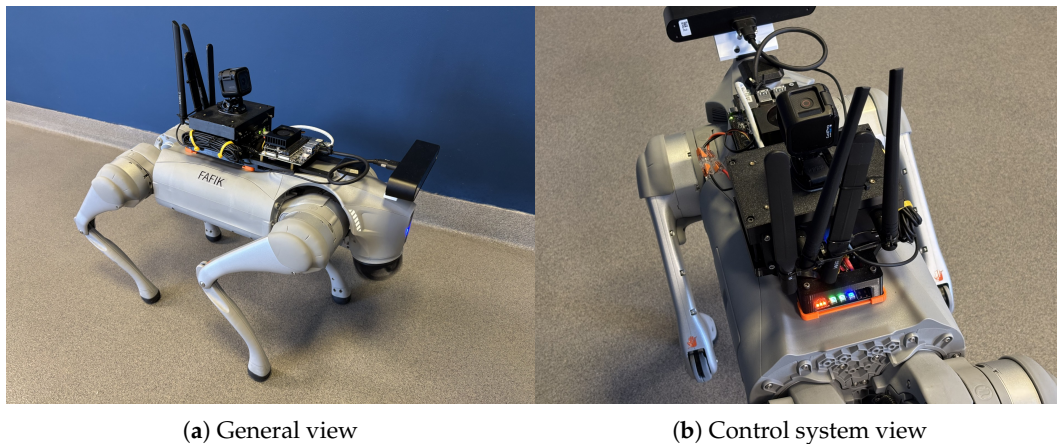
The main research objectives were achieved using the Unitree Go2 Edu quadruped robot [47], shown in Figure 1. It is equipped with numerous sensors as standard, including the following:

- 2D camera—mounted on the front of the robot,
- Unitree 3D Lidar sensor—mounted on the underside of the robot,
- strain gauges mounted in each foot of the robot, allowing for the detection of ground contact and the force exerted by each foot,
- Intel RealSense D435i stereo camera—allowing the robot to detect depth and color,
- IMU (Inertial Measurement Unit) sensor

An additional component of the robot is a computing module, the so-called Expansion Dock, equipped with an NVIDIA Orin Nano 8GB computing unit running Ubuntu 20.04 and ROS 2 Foxy. Unfortunately, this solution has a drawback that prevents proper use of the robot's capabilities. During tests with the basic equipment, problems with data acquisition, which can be used for localization and navigation, were encountered.

Due to emerging issues with the software and communication with the default control system, the decision was made to modify several components of this system.

As part of the design, the entire computing unit was modified using a proprietary solution, also based on NVIDIA Orin, but running the newer Ubuntu 22.04 operating system and the Humble version of ROS 2. Additionally, the RealSense camera was replaced with a Stereolabs ZED2i camera.



(a) General view

(b) Control system view

Figure 1. The Unitree Go2 Edu robot used in the project was modified by the authors.

4. Hardware Stack

Figure 2 shows a general diagram of the connection of the main robot control elements and their mutual communication.

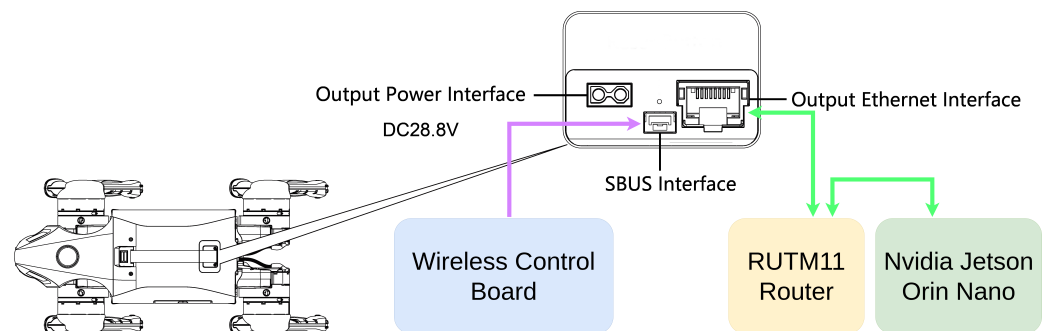


Figure 2. System connection schematic.

The most important elements of the proposed and implemented control system shown in Figure 1 are described in the following sections of the article.

4.1. Control Board

To integrate Jetson Orin Nano, industrial router (RUTM11) and resolve power distribution constraints, a custom Power Management and Interface Board (PMIB) was designed and fabricated (Figure 3). This PCB acts as the central hub for the system, fulfilling three primary functions: power regulation, wireless communication bridging, and visual feedback for the operator.

- **Power Distribution:** The robot's primary battery pack (operating between 21.6 V and 33.6 V) serves as the main power input. The PMIB features a step-down converter to supply a stable 12 V rail, which is needed for powering the NVIDIA Jetson Orin Nano and RUTM11 router.
- **Wireless-to-SBUS Bridge:** The core logic is driven by a Seeed Studio XIAO ESP32S3 microcontroller. It serves as a dedicated wireless bridge, receiving remote control commands via Wi-Fi or Bluetooth and translating them into the SBUS protocol. This SBUS signal is injected directly into the Unitree Go2's low-level controller, enabling a quick, manual override and safety fallback independent of the main ROS 2 system.
- **Visual Feedback:** For immediate system diagnostics without external displays, the board is equipped with dedicated power indicator LEDs for the 3.3 V, 5 V, and 12 V

rails, as well as a string of WS2812B addressable LEDs driven by the microcontroller to indicate the current state of the SBUS bridge and manual control.



Figure 3. The custom power management and interface board.

4.2. NVIDIA Jetson Orin Nano

An NVIDIA Jetson Orin Nano acts as a high-level controller in the system and is utilized as the primary edge computing module (Figure 4). Jetson fulfills the main functions of data acquisition, data processing and autonomous decision-making.

- **Data Acquisition:** The module serves as the central hub for high-bandwidth sensory input. It directly interfaces with the Stereolabs ZED2i camera via a USB 3.0 connection and robot built-in lidar via ethernet interface.
- **Data Processing:** Utilizing its CUDA-accelerated GPU, the Jetson handles computationally intensive tasks in real time. This involves executing the YOLO neural network pipeline to detect the targeted object and extracting precise distance metrics by analyzing the depth map region corresponding to the generated bounding box.
- **Autonomous Decision-Making:** Based on the processed spatial coordinates, the module evaluates the target's relative position against the robot. It subsequently executes the control logic to calculate the required linear and angular velocities, publishing the final trajectory commands to the ROS 2 topic read by the robot ROS 2 command interface.

Additionally, the Jetson processes LiDAR data to generate 3D point clouds. While these data are captured for postanalysis and the future development of 3D mapping and active collision avoidance algorithms, the current motion control loop relies solely on visual perception.

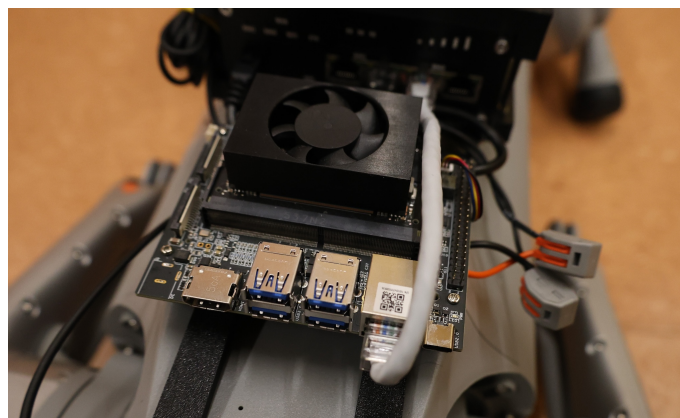


Figure 4. The NVIDIA Jetson Orin Nano computing module mounted on the robot.

4.3. Teltonika RUTM11

To establish a robust communication and external connectivity, a Teltonika RUTM11 industrial router is integrated into the system (Figure 5). This device manages internal data exchange and provides additional spatial features, fulfilling three functions:

- **Local Communication:** It provides a reliable Ethernet LAN connecting the Jetson Orin Nano and the Unitree Go2, guaranteeing low-latency data transmission for critical ROS 2 middleware protocols.
- **Dual-Mode Connectivity:** The router operates simultaneously as a Wi-Fi Access Point (AP) and a network client. As an AP, it allows operators to remotely stream telemetry without physical tethering. As a client, it enables the robot to connect to existing external infrastructure or internet networks.
- **GNSS Localization:** The device features an integrated Global Navigation Satellite System module, supplying absolute positioning coordinates that can be utilized for outdoor navigation and advanced sensor fusion.



Figure 5. The Teltonika RUTM11 industrial router providing GNSS and dual-mode networking.

4.4. Stereolabs ZED2i Camera

To provide accurate environment perception, a Stereolabs ZED2i stereo camera is integrated into the hardware stack (Figure 6). Designed for tough industrial and outdoor conditions, the ZED2i acts as the robot's primary vision sensor. It provides the essential sensory input for the perception node, fulfilling three key functions:

- **Visual Data Acquisition:** The camera captures high-definition RGB video and continuously streams it to the Jetson Orin Nano via USB 3.0. These frames provide the essential visual context required for the YOLO detection model.
- **Real-Time Depth Mapping:** Utilizing stereo vision algorithms, the ZED2i generates accurate depth maps alongside the RGB stream. By matching the 2D bounding boxes detected by YOLO with this depth data, the system extracts the exact 3D spatial coordinates (x, y, z) of the targeted person.
- **Seamless ROS 2 Integration:** The camera integrates smoothly into the software pipeline thanks to the official `zed-ros2-wrapper`. This compatibility provides immediate access to synchronized data streams, bypassing the need to write custom low-level drivers and significantly accelerating the development process.



Figure 6. The Stereolabs ZED2i stereo camera utilized for target detection and 3D localization.

5. Control Pipeline

The entire control architecture is implemented within the ROS 2 framework, utilizing the edge computing capabilities of the NVIDIA Jetson Orin Nano.

The software pipeline operates as a continuous closed-loop system, which can be divided into three stages: visual perception and target detection, spatial kinematics and motion generation. This section details the data flow across these interconnected nodes—from the acquisition of raw images to the publication of final velocity commands that control the robot’s movements.

ROS 2 Node Topology and Data Flow

The system relies on a decentralized node topology where specific computational tasks are isolated into dedicated processes. As illustrated in the block diagram (Figure 7), the data flows sequentially through three main stages:

1. **Perception Layer:** The pipeline originates with the Stereolabs ZED2i camera receiving information about its environment. The camera captures raw visual data and depth information, transmitting it via a USB 3.0 connection directly to the edge computing module.
2. **Jetson Orin Nano:** This module runs the core ROS 2 nodes for perception and control, splitting the workload into three sequential steps:
 - **YOLO Object Detection and Localization:** The video stream is processed by a YOLO neural network to detect the target person. By matching the resulting 2D bounding box with the camera’s depth map, the system extracts the 3D position (x, y, z) of the target. This spatial data is then published to the `/obj_det/objects` and `/depth/depth_registered` topics.
 - **Motion Controller:** A custom C++ node subscribes to these topics to drive the robot. As a reliable baseline approach, it relies on a straightforward, rule-based control logic rather than complex kinematics. It evaluates the 3D lateral angle error (e_t) of the target relative to the camera’s optical axis to steer left or right, and uses the metric depth (X_t) to adjust the forward or backward speed.
 - **Control Translator:** Because the robot’s native hardware does not directly accept standard `Twist` commands, an additional bridging node is utilized. It subscribes to `/cmd_vel` and translates velocity vectors into native `/wirelesscontroller` messages. Effectively, this node simulates a virtual remote controller, injecting steering commands into the robot’s low-level system.

3. Unitree GO2: The quadruped robot acts as the physical execution layer. Receiving the simulated /wirelesscontroller messages over the local network, the robot treats these inputs exactly as if they were coming from a human operator's remote control. Its built-in low-level hardware takes over to handle the physical movement required to follow the target.

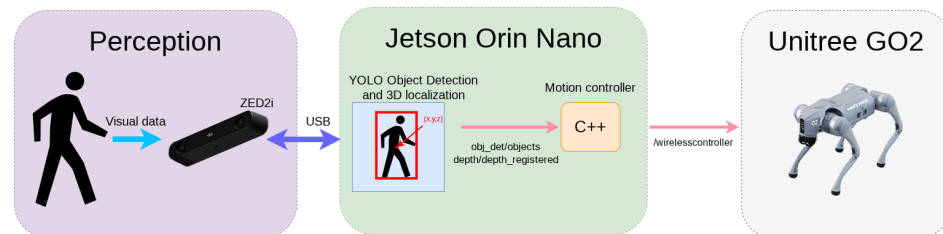


Figure 7. Diagram illustrating the data flow from the ZED2i camera to the Unitree GO2 actuation layer.

6. Visual Perception and Custom Object Detection

A common challenge in autonomous following systems is target ambiguity in crowded environments. When multiple people enter the camera's field of view, a standard human-detection model often struggles to determine which individual to follow. To overcome this and ensure reliable tracking, the system focuses on a visually distinct marker. While a high-visibility safety vest was used for this implementation, the approach is highly adaptable and can easily be trained to recognize other specific professional clothing, such as a firefighter's uniform.

6.1. Dataset and Local Training

The custom detection model was trained using a pre-annotated dataset of individuals wearing high-visibility vests. We chose the YOLO26n (nano) architecture because it delivers the high-speed inference required for edge computing without compromising detection accuracy. The network was trained locally for 100 epochs to recognize the vests across various lighting conditions and angles.

Training metrics indicate a highly successful learning process without overfitting, as both training and validation loss curves converged steadily throughout the entire 100-epoch run (Figure 8). The final model achieved mean average precision (mAP@0.5) of approximately 0.99, along with a robust mAP@0.5:0.95 nearing 0.90. Sample validation results in Figure 9 confirm the model's ability to accurately detect and bound the safety vests across different distances and real-world environments.

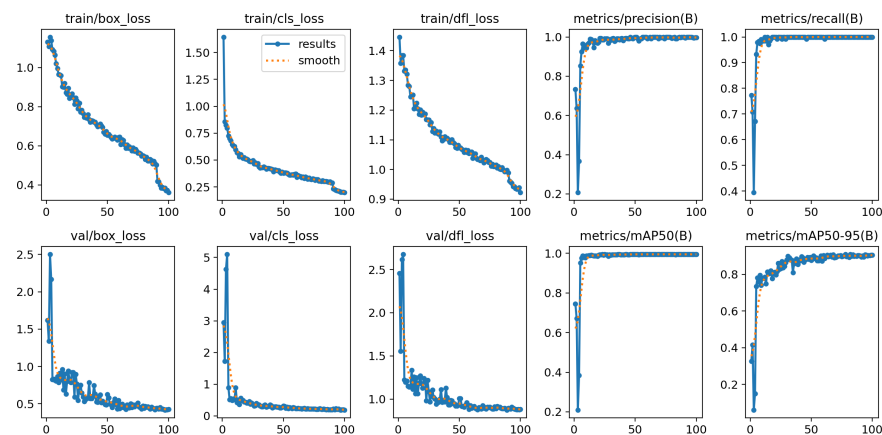


Figure 8. Training and validation metrics over 100 epochs, showing steady loss reduction and near-perfect mAP@0.5 scores.



Figure 9. Sample inference results from the validation dataset demonstrating accurate bounding box placement.

6.2. ONNX Conversion and ZED2i SDK Integration

Traditional vision pipelines often require a standalone ROS 2 node to run neural network inference and manually correlate the resulting 2D bounding boxes with a depth map. To reduce software complexity and minimize latency, this project directly utilizes the native Custom Object Detection module built into the Stereolabs ZED2i SDK.

To achieve this hardware-level integration, the locally trained YOLO26n model was exported to the ONNX (Open Neural Network Exchange) format using the standard Ultralytics framework:

```
1 !yolo export model=yolo26n.pt format=onnx simplify=True dynamic=False
   imgsiz=600
```

This conversion optimizes the model's architecture, making it fully compatible with the TensorRT inference engine utilized internally by the ZED2i driver.

6.3. Direct 3D Localization and Wrapper Configuration

To deploy the custom model, the `zed-ros2-wrapper` configuration parameters were modified to enable the Custom Object Detection module. This process requires adjustments in two separate configuration files. First, the object detection pipeline is explicitly activated and set to expect custom input within the `common_stereo.yaml` file:

```
1 object_detection:
2   od_enabled: true
3   enable_tracking: false
```

```

4  detection_model: 'CUSTOM_YOLOLIKE_BOX_OBJECTS'
5  max_range: 40.0
6  filtering_mode: 'NMS3D'
7  prediction_timeout: 2.0
8  allow_reduced_precision_inference: false

```

Subsequently, the `custom_object_detection.yaml` configuration is adjusted to point directly to the exported ONNX file path and define the required detection parameters (Listing 1):

Listing 1. Configuration snippet from `custom_object_detection.yaml` defining the path to the ONNX weights.

```

1  ros__parameters:
2  object_detection:
3  custom_onnx_file: 'yolo26n.onnx'
4  custom_onnx_input_size: 600
5  custom_class_count: 1
6  class_000:
7  label: 'yellow-vest'
8  model_class_id: 0
9  enabled: true
10 is_grounded: true
11 is_static: false
12 tracking_timeout: -1.0
13 tracking_max_dist: -1.0
14 max_box_width_normalized: -1.0
15 min_box_width_normalized: -1.0
16 max_box_height_normalized: -1.0
17 min_box_height_normalized: -1.0
18 max_box_width_meters: -1.0
19 min_box_width_meters: -1.0
20 max_box_height_meters: -1.0
21 min_box_height_meters: -1.0
22 object_acceleration_preset: 'DEFAULT'
23 max_allowed_acceleration: 100000.0
24 velocity_smoothing_factor: 0.5
25 min_velocity_threshold: 0.2
26 prediction_timeout_s: 0.5
27 min_confirmation_time_s: 0.05

```

With the configuration files prepared, the perception node is initialized using the ROS 2 launch system. The following command launches the wrapper, with modified `common_stereo` and `custom_object_detection` files automatically applying the specified parameters for the ZED2i camera (Listing 2):

Listing 2. Command to launch the ZED2i ROS 2 wrapper with the customized configuration.

```

1  ros2 launch zed_wrapper zed_camera.launch.py camera_model:=zed2i
    ros_params_override_path:=<path to>/common_stereo.yaml
    custom_object_detection_config_path:=<path to>/custom_object_detection.
    yaml object_detection_config_path:=<path to>/object_detection.yaml

```

During runtime, the wrapper automatically initializes the model and loads the custom weights into the camera's driver. Consequently, the ZED2i SDK internally handles the neural network inference and automatically fuses the 2D bounding boxes with its hardware-accelerated stereoscopic depth data. As a result, the camera node directly publishes the absolute 3D spatial coordinates (x, y, z) of the detected operator to the `obj_det/objects` topic. This seamless integration offloads processing overhead and provides the motion controller with ready-to-use spatial data at a high frequency.

Figure 10 illustrates the live output of this integrated pipeline, demonstrating the system successfully identifying the operator in the safety vest and instantly extracting their spatial coordinates.

To further reduce false detections and improve tracking reliability, a preliminary calibration procedure was implemented using the configuration parameters defined in the ZED2i wrapper. Specifically, tracking timeout thresholds, velocity-smoothing factors, and dynamic confirmation times (e.g., `min_confirmation_time_s: 0.05`) were empirically fine-tuned. This calibration ensures that only stable, confident 3D spatial coordinates are published to the motion controller, effectively filtering out transient background artifacts and false positive detections.

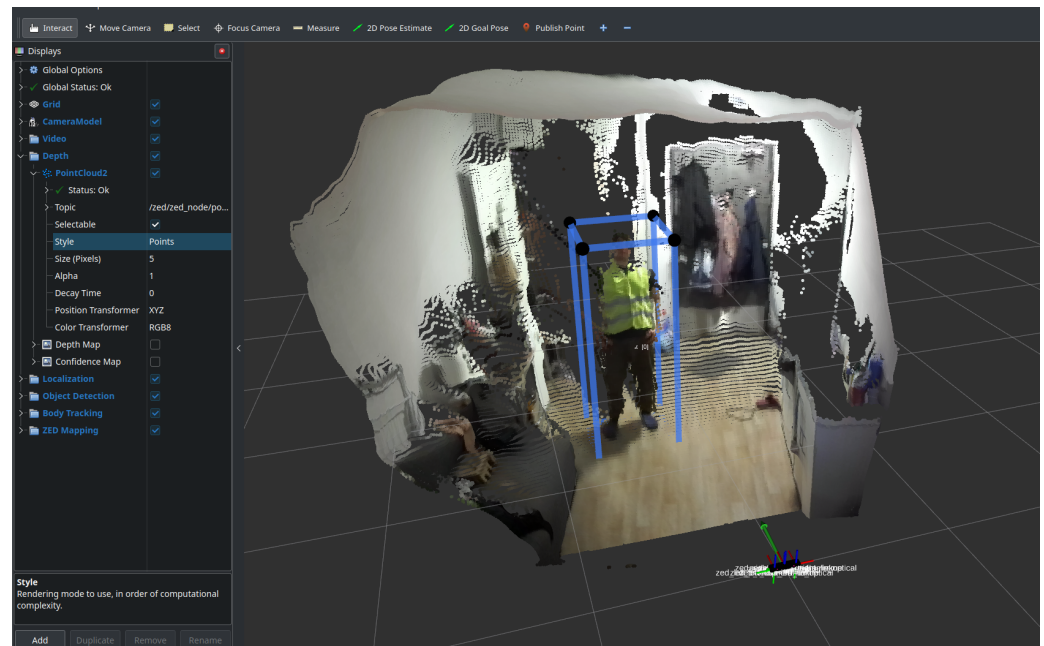


Figure 10. Live 3D object detection output from the zed wrapper, showing successful identification of the high-visibility vest and real-time spatial localization of the operator, visualized in RViz2.

7. Motion Generation and Control Strategy

Once the perception pipeline extracts the target's 3D coordinates, the system must translate this spatial data into actual movement. To ensure predictable and responsive tracking, the motion controller employs a straightforward, purely kinematic proportional control strategy.

7.1. Error-Driven Proportional Control Logic

A custom C++ node processes the spatial data published by the `zed_wrapper`. The target's position is defined as (X_t, Y_t, Z_t) within the `left_camera_frame`. Because this data complies with the standard ROS 2 REP-103 convention (where the X-axis points forward and the Y-axis points left), X_t directly corresponds to the depth distance to the target, while Y_t represents the lateral offset.

The primary goal of the controller is to keep the operator at a safe reference distance (D_{ref}) while keeping them centered in the robot's field of view ($Y_{ref} = 0$).

7.1.1. Linear Velocity Control

The robot's forward and backward speed, v_x , is driven by the depth error $e_d = X_t - D_{ref}$. To stop the robot from continuously oscillating or jittering when the operator stands still,

a distance deadband ($\pm\delta_d$) is applied. If the target is within this tolerance zone, the robot stops. Otherwise, it scales its speed proportionally:

$$v_x = \begin{cases} K_p \cdot (e_d - \delta_d) & \text{if } e_d > \delta_d \quad (\text{target is moving away}) \\ K_p \cdot (e_d + \delta_d) & \text{if } e_d < -\delta_d \quad (\text{target is too close}) \\ 0 & \text{if } |e_d| \leq \delta_d \quad (\text{target is in the optimal zone}) \end{cases} \quad (1)$$

where K_p is the experimentally tuned proportional gain for translation.

7.1.2. Angular Velocity Control

Steering is handled in a similar manner. The angular velocity ω_z depends on the angle error e_θ . A rotational deadband ($\pm\delta_\theta$) ensures that the robot maintains a stable heading as long as the target remains reasonably close to the center of the frame:

$$\omega_z = \begin{cases} K_\omega \cdot (e_\theta - \delta_\theta) & \text{if } e_\theta > \delta_\theta \quad (\text{target is to the left}) \\ K_\omega \cdot (e_\theta + \delta_\theta) & \text{if } e_\theta < -\delta_\theta \quad (\text{target is to the right}) \\ 0 & \text{if } |e_\theta| \leq \delta_\theta \quad (\text{target is centered}) \end{cases} \quad (2)$$

where K_ω is the proportional gain for rotation.

7.2. Actuation Translation

Finally, the calculated velocities (v_x, ω_z) are packaged as standard ROS 2 `geometry_msgs/Twist` messages and published to the `/cmd_vel` topic.

However, since the Unitree Go2 does not natively support standard `Twist` inputs, a dedicated translation node bridges this gap. It intercepts the velocity commands and translates them into the robot's native `/wirelesscontroller` format. Functioning as a “virtual joystick”, this node injects the autonomous commands directly into the Go2's low-level motion controller, allowing for smooth, real-time physical tracking of the operator.

8. Kinematic Foundations and Spatial Tracking Zones

To effectively implement the error-driven proportional control strategy described above, it is necessary to map the spatial target coordinates to the robot's physical constraints. This involves determining the kinematics that describe the transformations between the geometric center of the four-legged platform and the camera mounted on it [48]. The following sections detail the kinematic offsets and define the robot's operational zones based on the calculated spatial errors.

8.1. Coordinate System Transformation

Figure 11 shows the assumed location of the pivot point near the camera placed in the front part of the body. In compliance with ROS 2 conventions, the X-axis points forward, and the Y-axis points left. Assuming that the center of rotation near the camera minimizes the error resulting from rotations around the body's geometric center; the tracking point remains relatively stable in the field of view during rapid turns.

Let B denote the frame associated with the kinematic center of the robot and C denote the frame associated with the camera. The velocity of the robot's base is

$$\mathbf{v}_B = \begin{bmatrix} v_{bx} \\ v_{by} \\ \omega_b \end{bmatrix} \quad (3)$$

For a rigid body, the kinematic relationship is

$$\mathbf{v}_C = \mathbf{v}_B + \boldsymbol{\omega}_b \times \mathbf{r} \quad (4)$$

where

- $\mathbf{v}_C$ —camera point velocity vector,
- $\boldsymbol{\omega}_b = [0, 0, \omega_b]^T$ —angular velocity around the Z-axis,
- $\mathbf{r} = [r_x, r_y, 0]^T$ —camera offset relative to the center of the robot.

The cross product yields

$$\boldsymbol{\omega}_b \times \mathbf{r} = \begin{bmatrix} 0 \\ 0 \\ \omega_b \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ 0 \end{bmatrix} = \begin{bmatrix} -\omega_b r_y \\ \omega_b r_x \\ 0 \end{bmatrix} \quad (5)$$

Consequently,

$$\begin{bmatrix} v_{cx} \\ v_{cy} \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_{bx} - \omega_b r_y \\ v_{by} + \omega_b r_x \\ \omega_b \end{bmatrix} \quad (6)$$

For this specific quadruped, the camera is mounted centrally along the lateral axis, meaning $r_y = 0$. Assuming negligible holonomic lateral sliding ($v_{by} \approx 0$), the expression simplifies to

$$\begin{bmatrix} v_{cx} \\ v_{cy} \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_{bx} \\ \omega_b r_x \\ \omega_b \end{bmatrix} \quad (7)$$

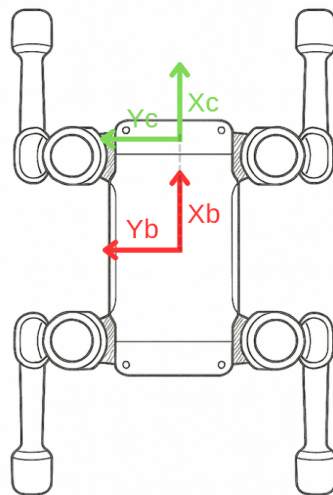


Figure 11. Shifted and adopted axes X–Y reflecting the kinematic offset.

8.2. Robot Operation Zones Based on Depth Error

The proportional linear velocity v_x defined in the previous section intrinsically creates three distinct physical operating zones for the robot, depending on the current depth X_t relative to the reference distance D_{ref} and the deadband δ_d . Figure 12 illustrates these states.

- **Zone a (Repulsion Zone):** Occurs when $X_t < D_{ref} - \delta_d$. The target has breached the minimum safe distance. The depth error e_d becomes significantly negative, prompting the controller to generate a negative v_x . The robot dynamically reverses to regain the optimal distance.

- Zone *b* (Translational Deadband): Occurs when $|X_t - D_{ref}| \leq \delta_d$. The target is in the optimal resting distance. Forward/backward translation is halted ($v_x = 0$), preventing jitter. In this zone, the robot only executes angular corrections (ω_z) to keep the target centered.
- Zone *c* (Dynamic Following Zone): Occurs when $X_t > D_{ref} + \delta_d$. The target is moving away. The positive error e_d generates a forward velocity v_x proportional to the distance, ensuring that the robot accelerates smoothly to catch up with the operator.

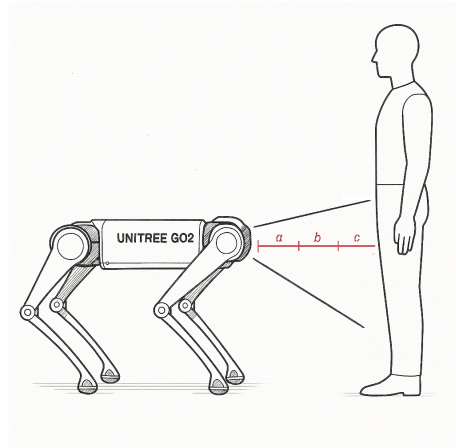


Figure 12. Three operational distance zones mapping the depth error e_d to robot behavior.

8.3. Translating Spatial Lateral Offset to Angular Velocity

Instead of relying on 2D pixel bounding boxes, this system uses 3D spatial data from the `zed_wrapper` (Figure 13). The robot's rotation is controlled directly by the target's deviation from the *X* axis of the robot (*Z* axis of ZED2i camera). To stop the robot from twitching in response to minor body sway from the operator, the rotational deadband δ_θ acts as a spatial buffer (e.g., ± 0.1 radian from the *X* axis).

$$\omega_z = 0 \quad \text{when } |e_t| \leq \delta_\theta \quad (8)$$

when the operator moves beyond this buffer ($|e_t| > \delta_\theta$), the proportional controller calculates the angular velocity $\omega_z = K_\omega \cdot (|e_t| - \delta_\theta) \cdot \text{sgn}(e_t)$ to smoothly turn the robot back toward the target.

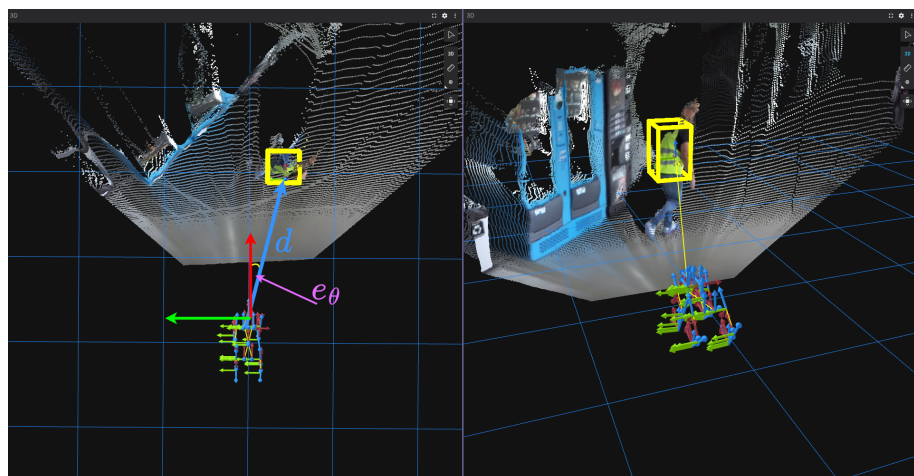


Figure 13. Visualization of the spatial tracking parameters in Foxglove.

9. Boundedness Analysis of the Human-Following Controller

This section investigates the boundedness properties of the proposed human-following controller implemented on the Unitree GO2 quadruped robot.

The controller utilizes the YOLO object detector to estimate the position of the human target in the image frame and generates velocity commands based on the distance and angular errors between the robot and the person.

9.1. Control Law

Let $d(t)$ denote the measured distance between the robot and the human, and d_r be the desired following distance.

The distance tracking error is defined as

$$e_d(t) = d(t) - d_r. \quad (9)$$

Similarly, let $\theta(t)$ represent the angular deviation of the target from the center of the camera image. The angular tracking error is

$$e_\theta(t) = \theta(t). \quad (10)$$

The commanded linear and angular velocities are given by

$$v(t) = K_d e_d(t), \quad (11)$$

and

$$\omega(t) = K_\theta e_\theta(t), \quad (12)$$

where $K_d > 0$, $K_\theta > 0$ are controller gains.

9.2. Assumptions

The following assumptions are adopted:

1. The human walking velocity is bounded, i.e.,

$$\|v_h(t)\| \leq V_{\max}. \quad (13)$$

2. The YOLO detection error is bounded,

$$|e_y(t)| \leq \varepsilon_d, \quad (14)$$

and

$$|e_\theta^y(t)| \leq \varepsilon_\theta. \quad (15)$$

3. The robot velocity commands satisfy

$$|v(t)| \leq v_{\max}, \quad |\omega(t)| \leq \omega_{\max}. \quad (16)$$

9.3. Distance Error Dynamics

Assuming bounded human motion, the distance error dynamics can be written as

$$\dot{e}_d(t) = v_h(t) - K_d e_d(t) - e_y(t). \quad (17)$$

Similarly, the angular error dynamics are

$$\dot{e}_\theta(t) = \omega_h(t) - K_\theta e_\theta(t) - e_\theta^y(t), \quad (18)$$

where $|\omega_h(t)| \leq \Omega_{\max}$ denotes the bounded angular motion of the human.

Theorem 1. Suppose that the assumptions stated above hold and that $K_d > 0, K_\theta > 0$.

Then, the distance and angular tracking errors are uniformly ultimately bounded. In particular,

$$|e_d(t)| \leq |e_d(0)|e^{-K_d t} + \frac{V_{\max} + \varepsilon_d}{K_d}, \quad (19)$$

and

$$|e_\theta(t)| \leq |e_\theta(0)|e^{-K_\theta t} + \frac{\omega_{\max} + \varepsilon_\theta}{K_\theta}. \quad (20)$$

Proof. Consider the Lyapunov function

$$V_d(e_d) = \frac{1}{2}e_d^2. \quad (21)$$

Its derivative along the trajectories of (17) is

$$\dot{V}_d = e_d \dot{e}_d = e_d(v_h - K_d e_d - e_y). \quad (22)$$

Using the bounds on the human velocity and measurement errors yields

$$\dot{V}_d \leq -K_d e_d^2 + (V_{\max} + \varepsilon_d)|e_d|. \quad (23)$$

Therefore, $\dot{V}_d < 0$ whenever

$$|e_d| > \frac{V_{\max} + \varepsilon_d}{K_d}. \quad (24)$$

Consequently, the distance error is uniformly ultimately bounded and satisfies (19).

The proof for the angular error follows analogously by considering

$$V_\theta(e_\theta) = \frac{1}{2}e_\theta^2.$$

This establishes (20).

□

The obtained results indicate that the proposed YOLO-based following controller guarantees bounded tracking performance provided that the human motion and perception errors remain bounded. The ultimate tracking accuracy depends on the controller gains and on the quality of the object detection algorithm. Increasing the proportional gains reduces the steady-state tracking error but may increase sensitivity to measurement noise and actuator limitations.

While these theoretical results indicate stable human-following behavior under idealized conditions, practical deployments on the Unitree GO2 robot must account for real-world, unmodeled dynamics. Factors such as ROS 2 inter-node communication latency, proprietary low-level actuation delays, hardware saturation, and transient perception noise can influence tracking accuracy. Consequently, while the controller provides a stable kinematic foundation, achieving optimal real world performance requires tuning of the controller gains to mitigate these hardware and middleware constraints.

10. Results

In the case of research on the quality of a robot dog following a selected person, the best way to present the results is the actual operation of the system in natural conditions.

Figure 14 shows several photos from the experiment of moving the “wanted” person with the robot. The place used to search and follow the selected person was the building of the Faculty of Automation, Electronics and Computer Science of the Silesian University of Technology. The photos show the calibration of the control system and several shots of the robot moving behind a moving human.

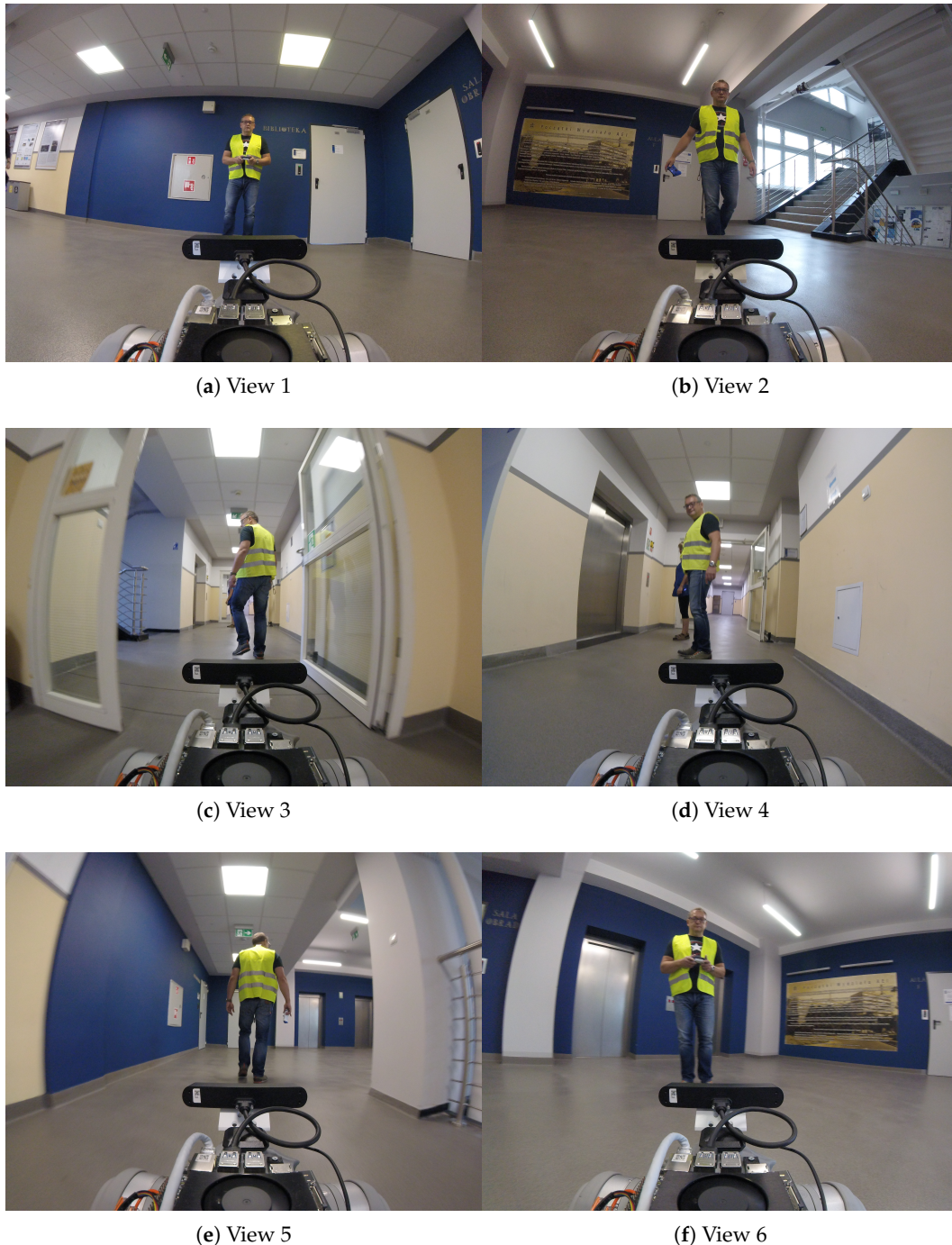


Figure 14. Sample photos showing the robot dog following a selected person.

In order to best present the results of the designed assistance robot control system, three links to a video file are attached in the Appendix, in which the robot moves in a rather crowded labyrinth of corridors of the building of the Silesian University of Technology and outdoor tracking.

11. Experiments

The following section presents the system outputs recorded directly from the ROS 2 environment. In contrast to the results presented in the Results Section, which were processed and selected for the evaluation of the proposed algorithm, the data presented here consist of raw messages published by the individual ROS 2 nodes. The data were recorded using the *ros2 bag* utility with the *ros2 bag record* command for the selected topics. This approach makes it possible to replay the entire experiment and perform a detailed analysis of the system operation based on the messages exchanged between its individual components.

The recorded data provide insight into the complete person-tracking pipeline, starting from person detection using the ZED2i stereo camera, and finally distance and position estimation based on measurements. Simultaneous analysis of data acquired from multiple sensors enables the evaluation of sensor synchronization, algorithm stability, and potential delays introduced during data processing and message publication.

Figure 15 presents an example of data recorded with *rosbag* at the 8th and 38th seconds of the experiment. Figure 15a–f show the person detection result, the detection associated with the ZED2i stereo camera, and the corresponding LiDAR measurements, respectively. This combination enables a direct comparison of the information provided by the individual sensors and facilitates the assessment of the sensor fusion process. The plots shown in Figure 15g–i present the evolution of the selected system parameters over a 30 s interval. These plots allow the stability of the measurements, the smoothness of the tracking process, and variations caused by the movement of the tracked person to be analyzed.

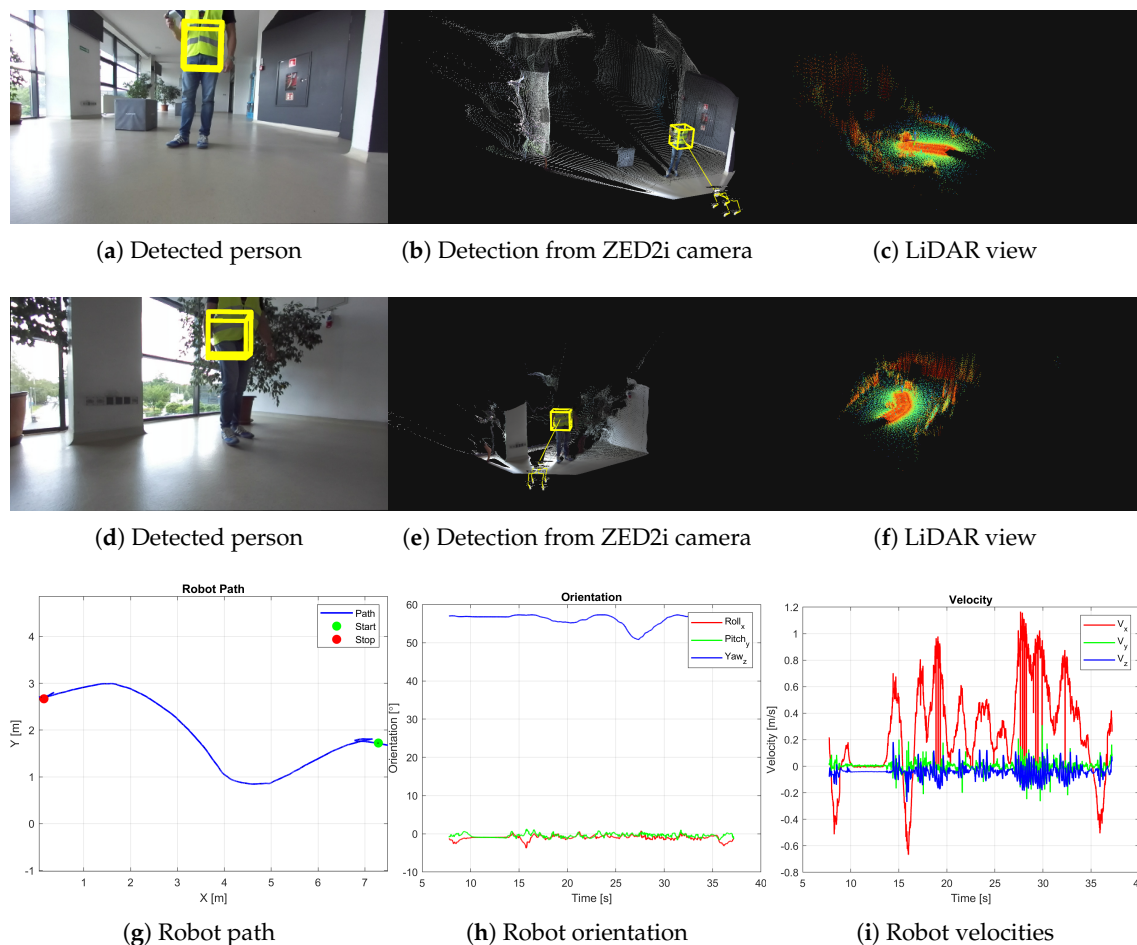


Figure 15. Sample data recorded directly from ROS 2 at 8 s and 38 s.

Figure 16 presents another example of data recorded at the 37th and 54th seconds of the experiment. Similar to the previous example, Figure 16a–f illustrate the person detection results, the corresponding ZED2i stereo camera data, and the LiDAR measurements. The plots shown in Figure 16g–i present the recorded system parameters over a 16 s interval. A comparison of both experiments demonstrates that the proposed system maintains stable operation under different tracking conditions, while the measurements remain consistent despite changes in the position of the tracked person relative to the mobile robot.

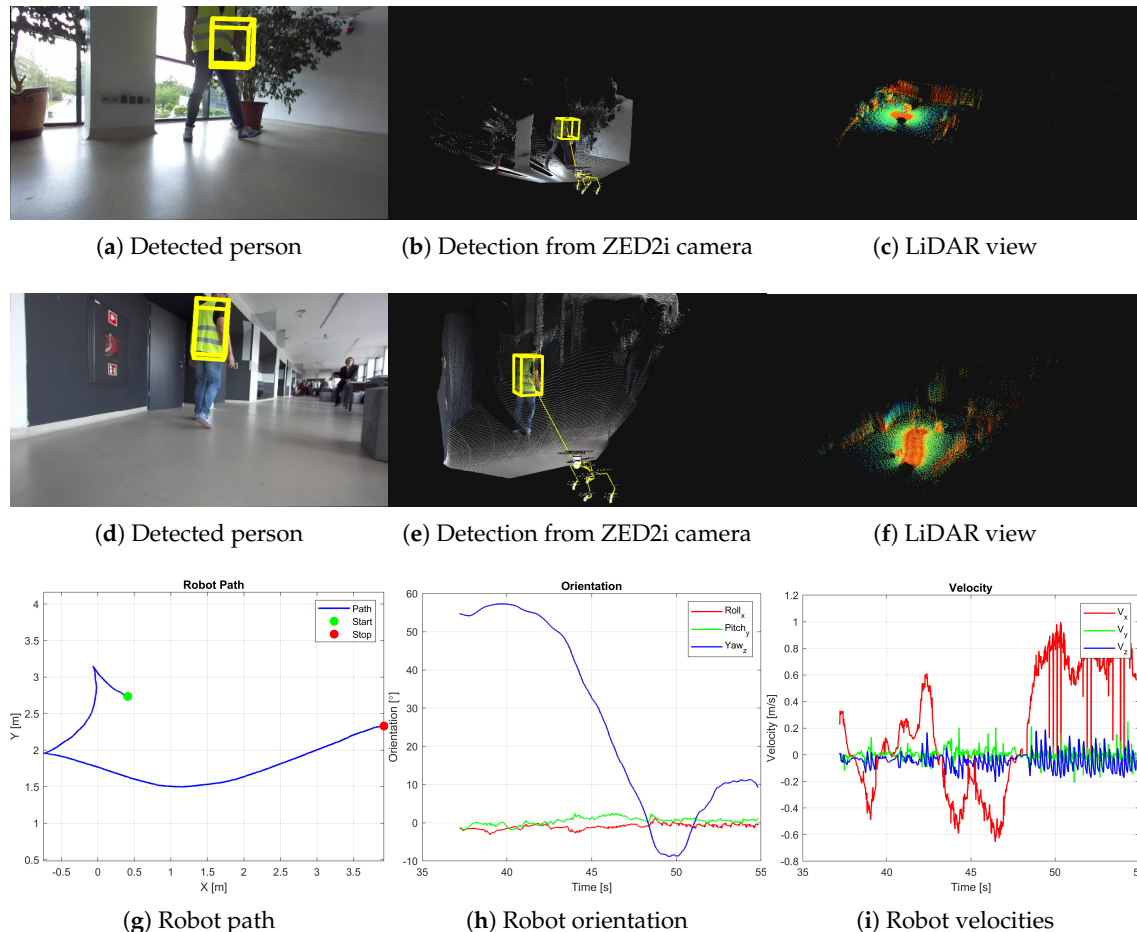


Figure 16. Sample data recorded directly from ROS 2 at 37 s and 54 s.

A particularly challenging experiment for evaluating the tracking accuracy and robustness of the proposed system is the scenario in which the tracked person suddenly accelerates and then comes to an abrupt stop. Such a situation represents a demanding test case because it requires the detection, tracking, and sensor fusion algorithms to rapidly update the estimated position of the target while maintaining stable and reliable estimates. The results of this experiment are presented in Figure 17. The recorded signals exhibit short-term variations caused by the rapid motion of the tracked person; however, the system successfully maintains continuous target tracking throughout the experiment.

An incorrectly positioned *textbounding box* can be observed in Figure 17b. It should be emphasized that this effect is not caused by an error in the object detection algorithm, but rather by delays introduced during data recording by ROS 2 and the *rosbag* utility. When messages from multiple topics are recorded simultaneously, small temporal offsets may occur between the captured image and the corresponding detection or sensor data. Consequently, the stored image may not correspond exactly to the same instant at which the detection data were recorded, resulting in an apparently misplaced bounding box. This

phenomenon is solely related to the data-recording process and does not affect the real-time performance or accuracy of the proposed tracking system.

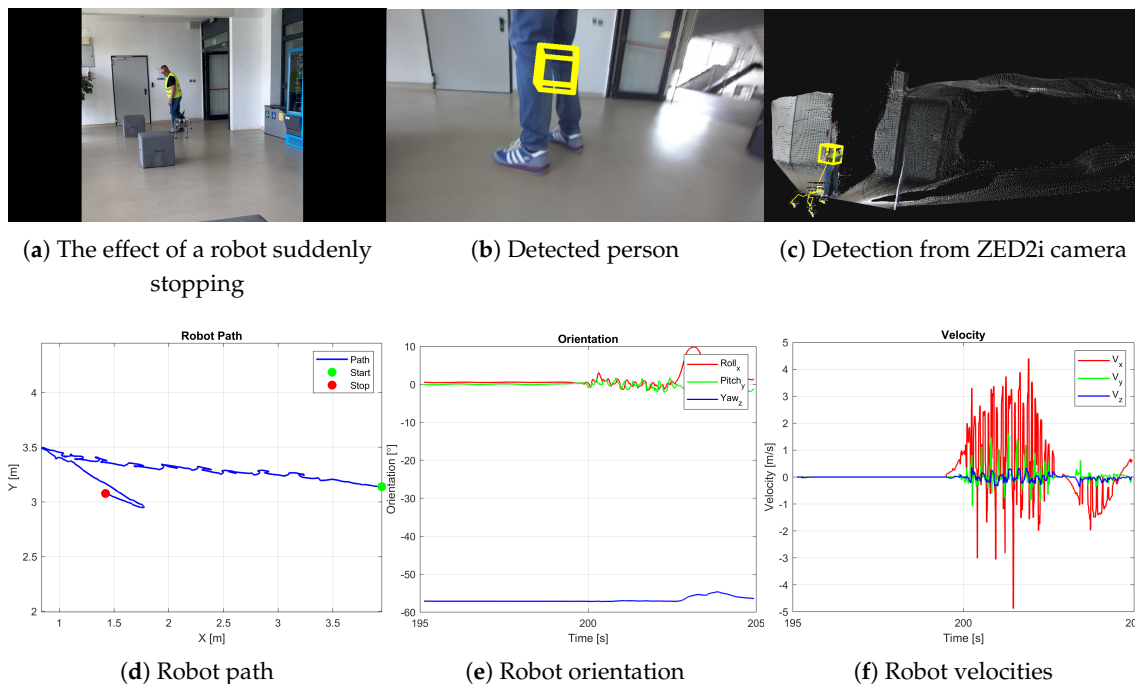


Figure 17. Sudden acceleration and abrupt stop of the tracked person.

Figure 18 presents three examples of partial occlusion of the tracked person, in which the ZED2i camera is unable to capture the entire body due to obstacles within its field of view. In the first two cases (Figure 18a,b), the person is partially occluded by a structural pillar, while in the third case (Figure 18c), the occlusion is caused by a box located between the camera and the tracked person. In each of these scenarios, the system has access only to partial visual information about the observed object, creating a challenging scenario for detection and tracking algorithms.

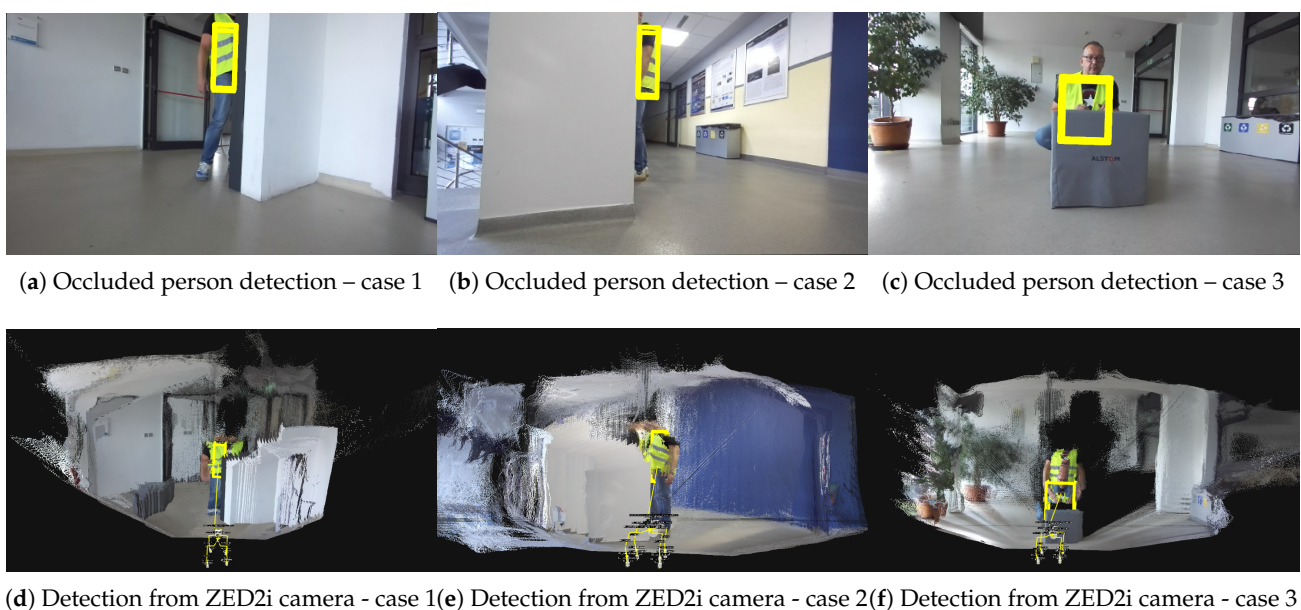


Figure 18. Examples of partial occlusion of the tracked person: (a,b) occlusion by a structural pillar; (c) occlusion by a box.

Despite the partial occlusion of the person's body, the system successfully detected the reflective safety vest worn by the tracked person in all analyzed cases. The results indicate that the proposed method maintains high detection performance even under conditions of partial object visibility, provided that the distinctive features of the reflective vest remain visible in the image. The obtained results confirm that using a reflective safety vest as a distinctive identification feature improves the robustness of person detection and tracking, even when the observed person is only partially visible.

12. Identified Limitations of the Presented Solution

During the experiments, several issues were identified that need to be addressed in further research, and which, in their current form, introduce certain limitations. The first such issue is the execution of movement along guaranteed collision-free trajectories. In the current solution, active collision avoidance is not yet implemented within the motion control loop, while the robot is equipped with a front-mounted LiDAR sensor that provides the necessary spatial data to enable forward-moving obstacle avoidance in the future, a fundamental hardware limitation remains. The LiDAR's current placement prevents scanning the area directly behind the robot, creating a sensory blind spot. Therefore, even when collision avoidance algorithms are fully integrated, reversing the robot will remain challenging, as a collision could still occur if an obstacle is located precisely within this unmapped rear zone.

Another limitation is the very fast movement required to keep up with the human (see Figure 17a) and sudden braking. Figure 17a shows a situation in which the robot, trying to avoid a collision, suddenly stops and stands on its front paws. In such cases, an algorithm for slow robot braking should be developed. The final limitation is the robot's response to a potential collision resulting from a dynamic obstacle striking the robot's body in a direction perpendicular to the robot's motion. In this case, a single LIDAR is insufficient.

13. Discussion

The tests performed showed that after appropriate camera calibration, the robot was able to follow the operator in a stable and consistent manner in various environmental conditions. The device responded correctly to both smooth and sudden changes in the operator's direction of movement and adjusted its speed to the pace of their walking.

The control algorithm proved to be fast and efficient enough to ensure smooth operation of the entire system in real time, without noticeable delays in response to changes in the operator's position. Visual data processing was performed continuously, and the generated robot trajectories were free from discontinuous zones caused by data processing delays and from sudden jumps and oscillations. Importantly, the system maintained a high level of performance even in situations when there were other people or objects in the camera's field of view, confirming its immunity to background noise.

The results obtained indicate that the developed solution meets the assumed reliability requirements and can constitute a solid basis for further expansion of the robot's functionality.

14. Conclusions

Further development of the project may include the implementation of more advanced image-processing methods, including deep neural networks for human silhouette detection and tracking, as well as obstacle classification within the environment. This would increase the system's robustness to varying lighting conditions and improve detection quality in environments with high object density and significant background motion. The integration of such methods into the existing architecture could be achieved using the hardware

accelerators available on the Jetson Orin Nano platform, enabling real-time operation without a significant increase in computational resource requirements.

Another important direction for future development is the integration of the system with additional sensors, such as LiDAR, GPS, and others. Data acquired from multiple sources could be combined within a sensor-fusion module, enabling the creation of more accurate environmental maps and improving the robot's collision-avoidance capabilities. As a result, reliable operation would also be possible in environments with limited visibility or under conditions where the vision signal is degraded.

It is also worth considering the development of a human motion prediction module that could estimate the future position of the operator based on trajectory history and motion dynamics. Such a solution would improve the smoothness of the following behavior and enable the system to react proactively, which is particularly important in the case of sudden changes in direction or movement speed. Additionally, reinforcement learning methods could be employed for adaptive tuning of control parameters, allowing the system to learn and improve its performance during operation.

From a practical perspective, the project has the potential for application not only in laboratory environments but also in internal logistics, warehouses, manufacturing facilities, and distribution centers, where autonomous robots can assist workers in transporting goods and materials. Expanding the system with multi-robot collaboration capabilities, enabling robots to exchange sensory data and coordinate trajectory planning, could pave the way for entire fleets of robots operating in an organized and safe manner.

An equally important area of development is the optimization of energy consumption and the miniaturization of components, which would extend battery-powered operating time and increase the overall mobility of the system. In the longer term, the advancement of such solutions may contribute to the wider adoption of autonomous legged robots in applications requiring mobility in challenging terrain, such as search-and-rescue operations, infrastructure inspection, and support in transporting equipment to hard-to-reach locations.

Author Contributions: Conceptualization, A.B., K.J. and M.G.; methodology, M.G.; software, M.G.; validation, K.J. and M.G.; formal analysis, A.B.; investigation, A.B., K.J. and M.G.; resources, M.G.; writing—original draft preparation, A.B., K.J. and M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Links to Video Materials

Below are three links to videos that demonstrate the actual operation of a dog robot.

- Robot operation in a crowded corridor
<https://1drv.ms/v/c/05213ec0af8c3e0a/IQDpbqF5GJvLQK9WkeuJHofeAVRscKSkmP93UIspIuTAIVA?e=HGmfEU> (accessed on 17 July 2026).
- Bipedal robot tracking
https://1drv.ms/v/c/05213ec0af8c3e0a/IQASvYv9HOaRSbsTwmTI3ygBAeNty6IsFgC1_c1RLaQIBlg?e=9rTkuL (accessed on 17 July 2026).
- Outdoor tracking

<https://1drv.ms/v/c/05213ec0af8c3e0a/IQArpLDj3sYaRaLQWI7Lkq5xAcZw-P-Yd2ktQSXKQZiqe44?e=Z5kJqP> (accessed on 17 July 2026).

References

1. Qin, C.; Ruwanpathirana, G.; Thilakarathna, S.; Wen, H.; Ji, Y.; Yue, J.; Baduge, S. A comprehensive review of quadruped robots: Vision perception, motion control, applications and challenges. *J. Autom. Intell.* **2026**, *in press*. [\[CrossRef\]](#)
2. Hong, B.; Lin, Z.; Chen, X.; Hou, J.; Lv, S.; Gao, Z. Development and application of key technologies for Guide Dog Robot: A systematic literature review. *Robot. Auton. Syst.* **2022**, *154*, 104104. [\[CrossRef\]](#)
3. Malfi, P.; Nicolella, A.; Spirito, M.; Cosenza, C.; Niola, V.; Savino, S. Motion Sensing Study on a Mobile Robot Through Simulation Model and Experimental Tests. *WSEAS Trans. Appl. Theor. Mech.* **2022**, *17*, 79–85. [\[CrossRef\]](#)
4. Garrosa, M.; Ceccarelli, M.; Russo, M.; Yang, B. Design of a Mobile Assisting Robot for Blind and Elderly People. *Appl. Sci.* **2025**, *15*, 10474. [\[CrossRef\]](#)
5. Li, Z.; Xu, N.; Zhang, X.; Peng, X.; Song, Y. Motion Control Method of Bionic Robot Dog Based on Vision and Navigation Information. *Appl. Sci.* **2023**, *13*, 3664. [\[CrossRef\]](#)
6. Chuang, T.K.; Lin, N.C.; Chen, J.S.; Hung, C.H.; Huang, Y.W.; Teng, C.; Huang, H.; Yu, L.F.; Giarré, L.; Wang, H.C. Deep Trail-Following Robotic Guide Dog in Pedestrian Environments for People who are Blind and Visually Impaired—Learning from Virtual and Real Worlds. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 5849–5855. [\[CrossRef\]](#)
7. Chen, Y.; Xu, Z.; Jian, Z.; Tang, G.; Yang, L.; Xiao, A.; Wang, X.; Liang, B. Quadruped Guidance Robot for the Visually Impaired: A Comfort-Based Approach. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 12078–12084. [\[CrossRef\]](#)
8. Pun, N.S.; Sonbhadra, S.K.; Agarwal, S.; Rai, G. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. *arXiv* **2021**, arXiv:2005.0138.
9. Xu, Z.; Zhan, X.; Xiu, Y.; Suzuki, C.; Shimada, K. Onboard Dynamic-Object Detection and Tracking for Autonomous Robot Navigation With RGB-D Camera. *IEEE Robot. Autom. Lett.* **2024**, *9*, 651–658. [\[CrossRef\]](#)
10. Jia, D.; Hermans, A.; Leibe, B. DR-SPAAM: A Spatial-Attention and Auto-regressive Model for Person Detection in 2D Range Data. In Proceedings of the 2020 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 10270–10277. [\[CrossRef\]](#)
11. Beyer, L.; Hermans, A.; Linder, T.; Arras, K.O.; Leibe, B. Deep Person Detection in 2D Range Data. *arXiv* **2018**, arXiv:1804.02463.
12. Jung, E.J.; Lee, J.H.; Yi, B.J.; Park, J.; Yuta, S.; Noh, S.T. Development of a Laser-Range-Finder-Based Human Tracking and Control Algorithm for a Marathoner Service Robot. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 1963–1976. [\[CrossRef\]](#)
13. Gómez, J.; Aycard, O.; Baber, J. Efficient Detection and Tracking of Human Using 3D LiDAR Sensor. *Sensors* **2023**, *23*, 4720. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Hasan, M.; Hanawa, J.; Goto, R.; Suzuki, R.; Fukuda, H.; Kuno, Y.; Kobayashi, Y. LiDAR-based detection, tracking, and property estimation: A contemporary review. *Neurocomputing* **2022**, *506*, 393–405. [\[CrossRef\]](#)
15. Jia, D.; Hermans, A.; Leibe, B. 2D vs. 3D LiDAR-based Person Detection on Mobile Robots. In Proceedings of the 2022 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 3604–3611. [\[CrossRef\]](#)
16. Kovacs, I.F.; Karolyi, A.C.; Stângaciu, C.S.; Stângaciu, V.; Nimară, S.; Curiac, D.I. An RFID-Based Indoor Guiding System for Visually Impaired People. *Information* **2025**, *16*, 220. [\[CrossRef\]](#)
17. Yang, C.H.; Juang, J.G. Guide Robot Based on Image Processing and Path Planning. *Machines* **2025**, *13*, 560. [\[CrossRef\]](#)
18. Kim, J.; Lee, D.; Choi, Y.; Lee, I. Motion Control Framework for Interactive Robotic Guide Dogs: A Systems Perspective. *IEEE Robot. Autom. Lett.* **2026**, *11*, 3422–3429. [\[CrossRef\]](#)
19. Rees, N.; Thiyagarajan, K.; Kodagoda, S. Robotic Guide Dog for Real-time Indoor Object Detection and Classification with Localization. In Proceedings of the 2024 IEEE Applied Sensing Conference (APSCON), Goa, India, 22–24 January 2024; pp. 1–4. [\[CrossRef\]](#)
20. Morlando, V.; Lippiello, V.; Ruggiero, F. Tethering a Human with a Quadruped Robot: A Guide Dog to Help Visually Impaired People. In Proceedings of the 2023 31st Mediterranean Conference on Control and Automation (MED), Limassol, Cyprus, 26–29 June 2023; pp. 547–553. [\[CrossRef\]](#)
21. Kim, J.T.; Yu, W.; Kothari, Y.; Tan, J.; Turk, G.; Ha, S. Transforming a Quadruped into a Guide Robot for the Visually Impaired: Formalizing Wayfinding, Interaction Modeling, and Safety Mechanism. *arXiv* **2023**, arXiv:2306.14055.
22. Kim, J.T.; Yu, W.; Tan, J.; Turk, G.; Ha, S. How to Train Your Guide Dog: Wayfinding and Safe Navigation with Human-Robot Modeling. In Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction, HRI Companion '23; Association for Computing Machinery: New York, NY, USA, 2023; pp. 221–225. [\[CrossRef\]](#)

23. DeFazio, D.; Hirota, E.; Zhang, S. Seeing-Eye Quadruped Navigation with Force Responsive Locomotion Control. *arXiv* **2023**, arXiv:2309.04370.
24. Wang, L.; Chen, Q.; Zhang, Y.; Li, Z.; Yan, T.; Wang, F.; Zhou, G.; Gong, J. Can Quadruped Navigation Robots be Used as Guide Dogs? *arXiv* **2023**, arXiv:2210.08727.
25. Due, B.L. Guide dog versus robot dog: Assembling visually impaired people with non-human agents and achieving assisted mobility through distributed co-constructed perception. *Mobilities* **2023**, *18*, 148–166. [\[CrossRef\]](#)
26. Hwang, H.; Jung, H.T.; Giudice, N.A.; Biswas, J.; Lee, S.I.; Kim, D. Towards Robotic Companions: Understanding Handler-Guide Dog Interactions for Informed Guide Dog Robot Design. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, CHI '24*; Association for Computing Machinery: New York, NY, USA, 2024. [\[CrossRef\]](#)
27. Hata, R.; Trikasemsak, N.; Giudice, A.; Doore, S.A. See Spot Guide: Accessible Interfaces for an Assistive Quadruped Robot. *arXiv* **2024**, arXiv:2402.11125.
28. Xiao, A.; Tong, W.; Yang, L.; Zeng, J.; Li, Z.; Sreenath, K. Robotic Guide Dog: Leading a Human with Leash-Guided Hybrid Physical Interaction. In *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 30 May–5 June 2021; pp. 11470–11476. [\[CrossRef\]](#)
29. Wakabayashi, H.; Hiroi, Y.; Miyawaki, K.; Ito, A. Development of a Personal Guide Robot That Leads a Guest Hand-in-Hand While Keeping a Distance. *Sensors* **2024**, *24*, 2345. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Chen, T.; Huangfu, Y.; Srigrarom, S.; Khoo, B.C. Path Planning and Motion Control of Robot Dog Through Rough Terrain Based on Vision Navigation. *Sensors* **2024**, *24*, 7306. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Hwang, H.; Xia, T.; Keita, I.; Suzuki, K.; Biswas, J.; Lee, S.I.; Kim, D. System Configuration and Navigation of a Guide Dog Robot: Toward Animal Guide Dog-Level Guiding Work. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 29 May–2 June 2023; pp. 9778–9784. [\[CrossRef\]](#)
32. Kim, J.T.; Byrd, M.; Crandell, J.L.; Walker, B.N.; Turk, G.; Ha, S. Understanding Expectations for a Robotic Guide Dog for Visually Impaired People. In *Proceedings of the 2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Melbourne, Australia, 4–6 March 2025; pp. 262–271. [\[CrossRef\]](#)
33. Li, J.; Corno, M.; Farina, M. A method to detect the sudden stopping in an assistive robot for the visually impaired people. *IFAC-PapersOnLine* **2024**, *58*, 624–629. [\[CrossRef\]](#)
34. Li, J.; Farina, M.; Mozzarelli, L.; Cattaneo, L.; Rattamasanaprapai, P.; Tagarelli, E.A.; Corno, M.; Perego, P.; Andreoni, G.; Lettieri, E. BUDD-e: An autonomous robotic guide for visually impaired users. *Control Eng. Pract.* **2026**, *172*, 106854. [\[CrossRef\]](#)
35. Chitiken, H.; Sanfilippo, F.; Ma, S. Robotics in Search and Rescue (SAR) Operations: An Ethical and Design Perspective Framework for Response Phase. *Appl. Sci.* **2023**, *13*, 1800. [\[CrossRef\]](#)
36. Jamwal, H.D.S.; Singh, S. Comprehensive Survey on Autonomous Disaster Reconnaissance: A Comparative Analysis of UAVs and UGVs. *Telecom* **2026**, *7*, 31. [\[CrossRef\]](#)
37. Bakzadeh, R.; Joao, K.M.; Androulakis, V.; Khaniani, H.; Shao, S.; Hassanalian, M.; Roghanchi, P. Enhancing Emergency Response: The Critical Role of Interface Design in Mining Emergency Robots. *Robotics* **2025**, *14*, 148. [\[CrossRef\]](#)
38. Sharma, K.; Doriya, R.; Pandey, S.K.; Kumar, A.; Sinha, G.R.; Dadheech, P. Real-Time Survivor Detection System in SaR Missions Using Robots. *Drones* **2022**, *6*, 219. [\[CrossRef\]](#)
39. Ji, G.; Mun, J.; Kim, H.; Hwangbo, J. Concurrent Training of a Control Policy and a State Estimator for Dynamic and Robust Legged Locomotion. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4630–4637. [\[CrossRef\]](#)
40. Di Carlo, J.; Wensing, P.M.; Katz, B.; Bledt, G.; Kim, S. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 1–5 October 2018; pp. 1–9. [\[CrossRef\]](#)
41. Setchi, R.; Dehkordi, M.B.; Khan, J.S. Explainable Robotics in Human-Robot Interactions. *Procedia Comput. Sci.* **2020**, *176*, 3057–3066. [\[CrossRef\]](#)
42. Dehkordi, M.B.; Mansy, R.; Zaraki, A.; Singh, A.; Setchi, R. Explainability in Human-Robot Teaming. *Procedia Comput. Sci.* **2021**, *192*, 3487–3496. [\[CrossRef\]](#)
43. Anjomshoe, S.; Najjar, A.; Calvaresi, D.; Främling, K. Explainable Agents and Robots: Results from a Systematic Literature Review. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19*, Montreal, QC, Canada, 13–17 May 2019; pp. 1078–1088. [\[CrossRef\]](#)
44. Padmanabha, A.; Yuan, J.; Gupta, J.; Karachiwalla, Z.; Majidi, C.; Admoni, H.; Erickson, Z. VoicePilot: Harnessing LLMs as Speech Interfaces for Physically Assistive Robots. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology, UIST '24*; Association for Computing Machinery: New York, NY, USA, 2024. [\[CrossRef\]](#)
45. Kim, C.Y.; Lee, C.P.; Mutlu, B. Understanding Large-Language Model (LLM)-powered Human-Robot Interaction. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, HRI '24*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 371–380. [\[CrossRef\]](#)

46. Bejarano Sepulveda, E.J.; Bo, V.; Sanfeliu, A.; Garrell, A. Perception–Awareness–Decision: Socially-Aware Robot Navigation and Interaction. In *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction, HRI Companion '26*; Association for Computing Machinery: New York, NY, USA, 2026; pp. 258–262. [[CrossRef](#)]
47. Unitree Go2 Description. Available online: <https://www.unitree.com/go2> (accessed on 10 June 2026).
48. Atique, M.M.U.; Sarker, M.R.I.; Ahad, M.A.R. Development of an 8DOF quadruped robot and implementation of Inverse Kinematics using Denavit-Hartenberg convention. *Heliyon* **2018**, *4*, e01053. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.