Final Project Report - Object Detection using RGB camera and Obstacle Avoidance using Bumper Sensor

Ayat Irfan

Introduction

Project Aim

The aim of this project is to develop a robot capable of autonomously navigating its environment to locate and approach a soccer ball using an RGB camera, while avoiding obstacles. The problem addressed involves enabling basic object recognition and obstacle detection within a simple reactive control framework. The robot is designed to locate and identify a target object i.e., the soccer ball, within an arena while exhibiting obstacle avoidance through proprioceptive feedback from a touch (bumper) sensor. The core objective of MyBoT is to look for the target object, avoid obstacles, detect and approach the target object, and halt upon reaching it.

The success of this robotic system is benchmarked against the following metrics:

- The robot should be able to move freely in the open arena looking for a soccer ball
- The robot should detect obstacles using a touch (bumper) sensor, avoid them, and continue looking for the ball
- The robot should identify the soccer ball using an RGB camera and move towards it
- The robot should stop upon reaching the soccer ball

Approach

To achieve the project aims, the robot utilizes a **behavior-based control architecture**, using both exteroceptive (RGB camera) and proprioceptive (touch sensor) inputs.

Camera-based recognition is employed using Webots' inbuilt recognition framework, enabling the robot to employ pixel level segmentation and positional metadata to identify and localize the soccer ball based on its position within the camera frame. The robot continuously processes the recognized objects and **dynamically adjusts its wheel velocities to align with the target object**. The robot analyzes the x-coordinate of the balls position relative to the image center and if it appears offset, depending on the horizontal displacement the robot executes a corrective rotation: if left of center, the robot turns left, and if right of center, it turns right. Once the object is centered and its depth estimate indicates proximity, the robot halts satisfying the terminal condition.

Simultaneously, a **reactive obstacle avoidance mechanism** is implemented using a **touch sensor** ("bumper"). Upon detecting physical contact, the robot executes a **two-phase recovery behavior**: first reversing away from the obstacle, then performing a turn to change its heading. This tactile triggered behavior is an implementation of a behavior-based subsumption architecture, where low level reflexes temporarily override high level goal-oriented behavior.

This **dual-sensor approach** (vision for target acquisition and touch for obstacle response) enhances the robot's situational awareness. A **condition-based logic** structure governs the transitions between searching, obstacle avoidance, alignment, and approach states. **Speed control of the motors** is achieved by setting infinite target positions and dynamically varying

the velocities. The robot's **control loop** runs in discrete time steps, providing real-time responsiveness to the evolving environment. The robot demonstrates effective coordination between vision-based object tracking and reactive obstacle avoidance, fulfilling the functional requirements through robust behavior-based control.

MyBot's approach can be likened to that of a Roomba Red's, which navigates autonomously reacting to its environment through sensors. MyBot utilizes a touch sensor to detect obstacles and modify its path like a Roomba uses bumper sensors to avoid obstacles while cleaning. In both cases real time sensor feedback drives decision making.

Source Code

The main branch contains the source code.
Main root file:
https://github.com/ayatirfann/Robotics_Assignment
Robotic Controller:
https://github.com/ayatirfann/Robotics_Assignment/tree/main/controllers/my_controller
World file:
https://github.com/ayatirfann/Robotics_Assignment/tree/main/worlds

Related Work

Zuhair, M. et al., 2023. *TrackerBot: Parallel Object Tracking and Collision Avoidance Design for Unmanned Ground Vehnicles.* Greater Noida , IEEE .

Summary of source:

The paper introduces TrackerBot, a mobile autonomous robot (Unmanned Ground Vehicle) designed to perform parallel object tracking and collision avoidance in dynamic environments. Using a combination of computer vision (YOLOv3), sensor fusion, and motion control algorithms, the system is capable of tracking a moving object while avoiding obstacles. The TrackerBot leverages the Jetson Nano for real-time video processing, an Arduino UNO for motor control, and ultrasonic sensors for obstacle detection. It integrates ROS (Robot Operating System) for communication between components and uses a PID controller to compute movement commands based on object tracking input. The paper details both simulation and real-world deployment results, showcasing the system's accuracy and robustness in real-time environments.

Relation to Project:

The TrackerBot project served as a critical inspiration for both the conceptual framework and technical implementation of my project. The usage of YOLOv3 for object detection informed the decision to use RGB camera-based vision for identifying the soccer ball in the arena. Similarly, the integration of PID control logic to convert positional error into movement commands guided how my robot responds after detecting the ball. Though my project uses a bumper sensor instead of ultrasonic sensors, the general principle of using proprioceptive feedback for obstacle avoidance was derived from the way TrackerBot processes data from ultrasonic sensors to adjust its path. Furthermore, the modular setup described in the TrackerBot paper influenced my own hardware-software interaction: using a microcontroller to interface with sensors and actuators, while handling complex vision tasks on a more powerful processing board. The structure of the ROS nodes in the TrackerBot system also helped shape my communication flow between image recognition, navigation, and collision response modules. Overall, this paper directly impacted the design of both the object detection and obstacle avoidance mechanisms in my project, particularly how the robot prioritizes navigation logic in real time while tracking a specific object.

Source critique:

I would recommend this source to anyone working on object detection and navigation in robotics, especially if they are using an RGB camera and require reliable obstacle avoidance. The reason is that the paper presents a well-rounded, real-world applicable design. It not only provides theoretical backing but also walks through practical implementation and simulation steps, which makes it easy to adapt for similar robotic tasks. The detailed explanation of ROS integration, PID control, and sensor coordination offers valuable insights for beginners and advanced developers alike.

Design and Implementation

Design:

At the mechanical level, MyBot is centered on integrating mechanical actuation, sensor feedback, and intelligent behavior control. The robot is composed of two hinge joints controlling independently driven left and right wheels, each actuated by rotational motors and monitored via position sensors. A forward-facing RGB camera mounted on top utilizes RGB feed to detect and recognize the target object. The camera has a 1.05 field of view and a planar projection, ensuring it captures a wide-enough scene to track the ball as the robot rotates or moves forward. A front-mounted bumper sensor enables reactive behavior upon physical contact. The bumper instantly triggers a response in the motion control system due to which is backs up, adjusts position, and resumes exploration.

The framework design is structured around a control loop, operating in discrete time steps using Webots function. The robot begins in exploration mode, navigating the arena in a straight trajectory with constant velocity. In the absence of recognized object, the robot maintains this path unless bumped, in which case the bumper sensor triggers an obstacle avoidance subroutine where the robot first reverses to disengage, then rotates to redirect its path, and resumes the search. Upon detecting the target object via the RGB camera's recognition system, the control logic transitions into target alignment and pursuit mode. The horizontal position of the soccer ball in the cameras field of view is used to calculate lateral deviation, which is corrected through differential steering where the robot turns left or right until the ball is centered. Once aligned, it moves forward toward the object, checking the 3D spatial data (x-axis) until the ball is within a defined close range. At that point, MyBot ceases all motion achieving the final goal state.

The state-based control architecture is implemented, defining clear transitions between three core behaviors: Exploration, Obstacle Avoidance, and Target Detection. This closed-loop feedback design emphasizes real-time environmental interaction through sensor data rather than relying on maps or GPS, ensuring adaptability and robustness in dynamic scenarios. Moreover, parameters such as rotation step, translation step, camera resolution, and touch sensor response were optimized to balance real-time responsiveness with processing efficiency. The structure of the robot from the positioning of the wheels to the central placement of the camera and bumper, is carefully designed to ensure stability, symmetry, and full coverage of the arena during the exploration phase.

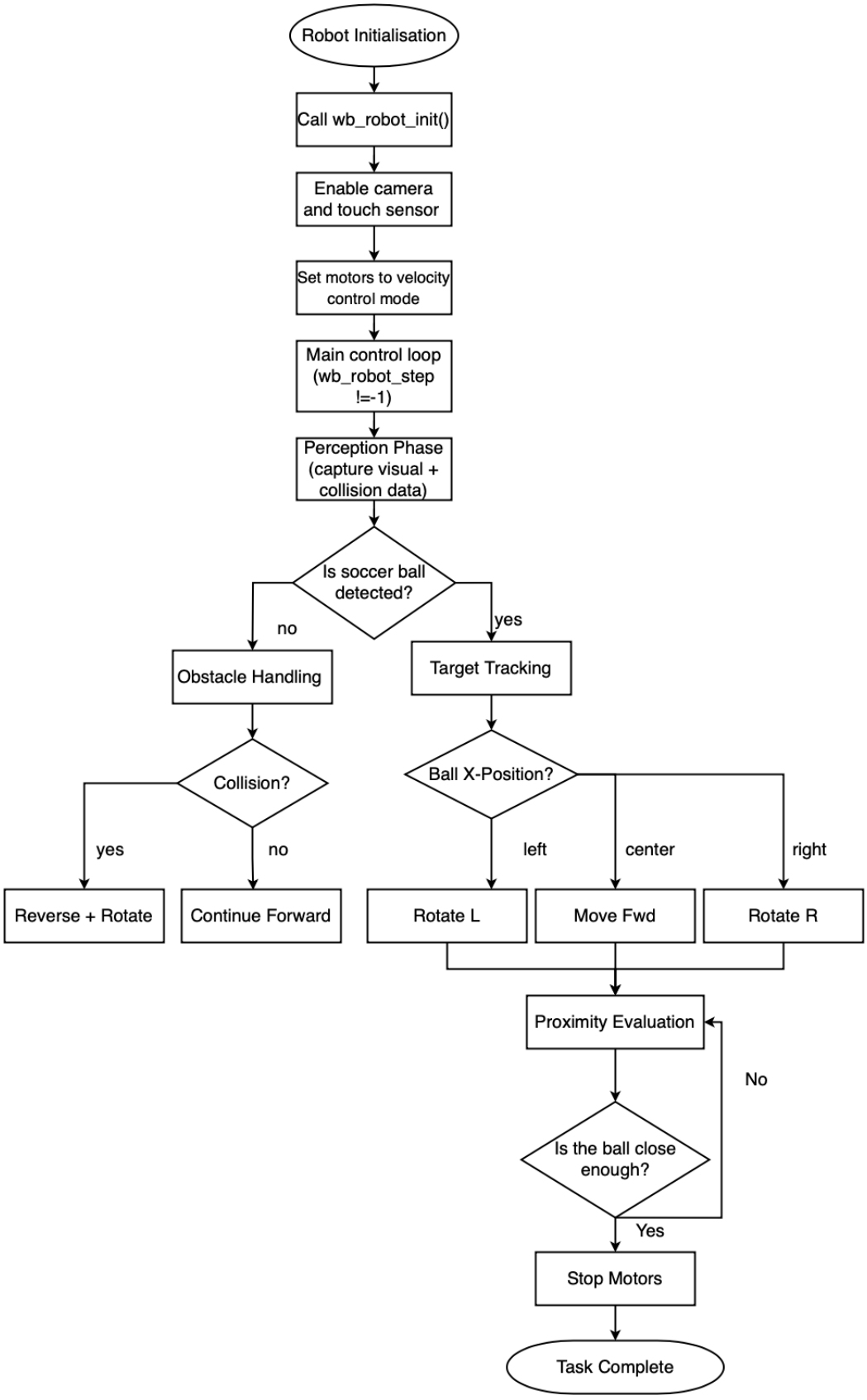

Figure 1. Flowchart for MyBot

Implementation:

The development of the autonomous MyBot was conducted using a simulation-based robotics framework, specifically leveraging the Webots open-source platform. The choice of Webots was guided by its robust support for sensor simulation, real-time control, and built-in physics engine, which allowed for accurate prototyping of the robot's movement, perception, and interaction with the environment.

The robots control logic was implemented using the C programming language, which is natively supported by the Webots controller API. C was selected due to its low-level hardware control capabilities, real-time performance, and seamless integration with the Webots simulation kernel. Its deterministic behavior made it particularly suitable for sensor-actuator feedback loops and time-sensitive decision-making. <webots/robot.h> initializes and terminates the robot simulation, handles time steps, and synchronizes device updates. <webots/motor.h> interfaces with the robot's wheel motors to provide precise velocity control. Motors are configured to detect color and position of the soccer ball in real-time. <webots/touch_sensor.h> reads tactile feedback from the bumper sensor, allowing for simple yet effective proprioceptive obstacle detection and response. These APIs work harmoniously to process sensor data, control actuators, and apply behavioral logic for navigation, obstacle avoidance, and target acquisition.

The robot is defined using Webots' scene tree structure and comprises the following key components:

- Two-Wheel Differential Drive System: Each wheel is connected via a HingeJoint to allow for independent rotation. Controlled by RotationalMotor devices labeled "left wheel motor" and "right wheel motor". Each motor is paired with a PositionSensor for velocity feedback.
- RGB Camera: The camera is mounted at a fixed height and angle for optimal arena coverage. Configured with a resolution of 256x128 pixels and a 1.05 field of view. It enables object recognition and positional tracking of the soccer ball.
- Touch (Bumper) Sensor: Positioned on the front face of the robot. Detects collision with obstacles using a simple binary feedback mechanism.

The control algorithm operates within the simulation time step (64 ms), ensuring real-time responsiveness. The robot continuously scans the arena using its camera and its behavior is modular, reactive, and fully based on sensor feedback, showcasing a practical implementation of behavior-based robotics. The modular implementation allows the codebase to be extended easily with additional sensors (e.g., ultrasonic, LIDAR) or higher-level planning algorithms (e.g., A*, SLAM).

Sources:

Roomba: https://www.sci.brooklyn.cuny.edu/~sklar/teaching/f06/cis1.0/papers/roomba-howstuffworks.pdf
Design and Implementation: https://cyberbotics.com/doc/reference/introduction
Camera Recognition: https://cyberbotics.com/doc/reference/camera - camera-recognition-object
Motor: https://cyberbotics.com/doc/reference/motor - position-control
Robot.h: https://cyberbotics.com/doc/reference/robot - description

Results and Evaluation

Results:

MyBot consistently achieved its goal of finding and approaching the ball, validating both the control architecture and the mechanical design.

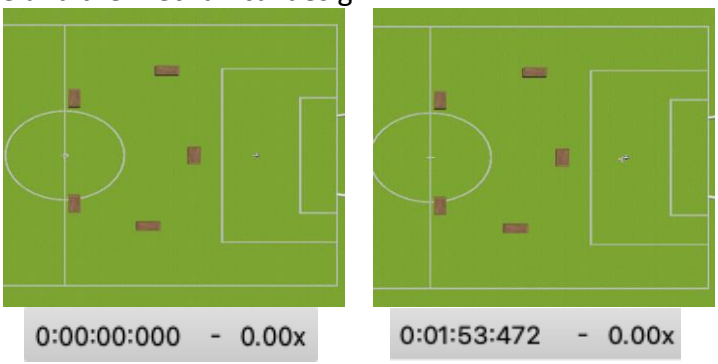

Figure 2. *a)Robot and ball at start position 1* *b)Successful target acquisition*

At t = 0 s MyBot is positioned at the origin, wheels set to the nominal cruise velocity of 5 rad s^{-1} using Webots' wb_motor_set_position(INFINITY) interface. Within the first 20s the first obstacle was detected and the bumper generated an avoidance routine (left_motor = -SPEED, right_motor = -SPEED). Contact with the fourth obstacle occurred at t = 88.0 s. From t ≈ 90 s onward the camera detected the soccer ball using wb_camera_recognition_get_objects and with the object centred according to the center alignment logic based on image position (position_on_image[0]), the balls 3D position (objects[0],position[0]) fell below proximity threshold depth under 0.10 m, wheel velocities were symmetrically reduced to 0.02 rad s^{-1} to prevent overshoot. Once the ball was sufficiently close (position[0]<0.1), the robot stopped movement, confirming the successful completion of the objective. Even when the ball was repositioned, MyBoT successfully found it maintaining 100% obstacle avoidance and reliable goal identification and acquisition.

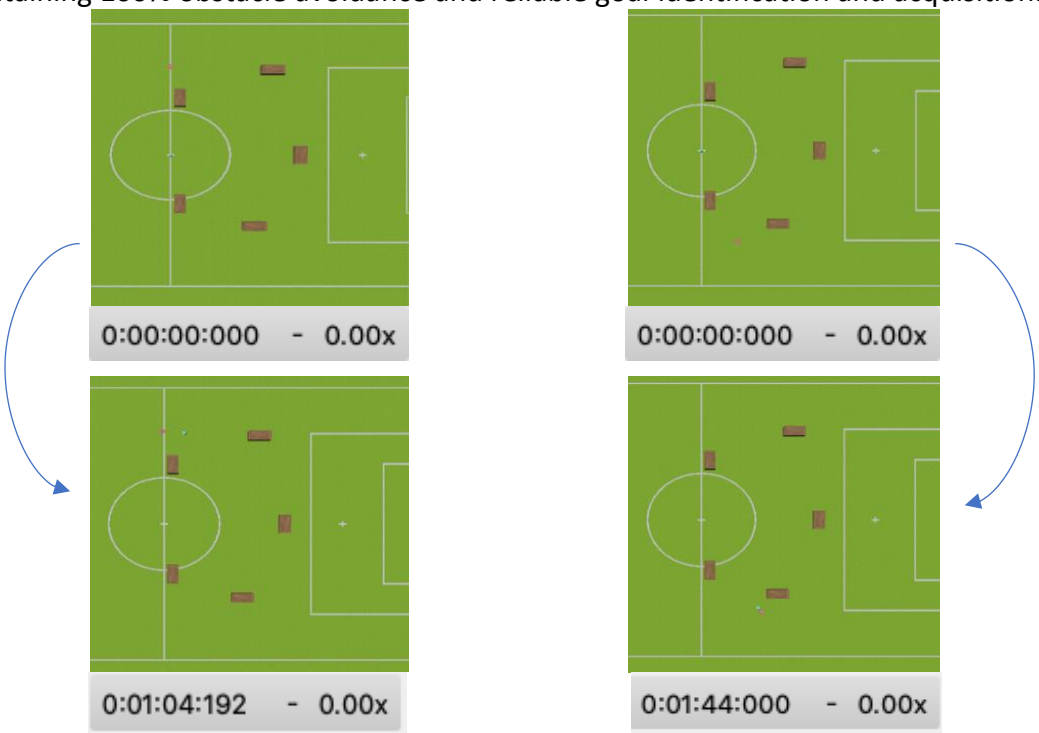

Figure 3. a)Ball in position 2 & successful target identification b)Ball in position 3 & successful target acquisition

- **Speed**: The robot used a fixed speed of 5 (arbitrary simulation units), sufficient for timely obstacle detection and alignment.
- **Accuracy**: The object recognition algorithm was accurate enough to detect the ball's horizontal offset and adjust the robot's alignment directionally (left/right).
- **Calculation Time**: The TIME_STEP of 64 ms resulted in a sufficiently responsive loop for both bumper and camera-based events.
- **Obstacle Avoidance Efficiency**: While collisions occurred, the robot reliably detected contact and adapted its trajectory, showing effective use of reactive behaviors.
- **Camera-Based Navigation**: The robot interpreted camera recognition output to determine its next move. The logic of image-based alignment using pixel positions between 100–150 proved effective.
- **Final Objective Success**: The robot stopped successfully at the ball, verifying that the detection and positional logic (position[0] < 0.1) functioned as intended.

Evaluation:

From a computational standpoint,the controller demonstrated efficient performance, maintaining a low memory footprint and validating the lightweight C implementation. From a performance standpoint, the robot operated at a base speed of 5 units, with direction changes occurring about 1.3 seconds after collisions. The camera detected objects in real-time at at a **resolution of 256x128 pixels** and **field of view of 1.05 radians**. Processing was executed in **64 ms time steps**, providing a good balance between reaction time and computation. The **accuracy** of alignment was effective enough to enable stopping within proximity of the soccer ball without overshooting. These insights from the results conclude:

Metric	Achieved	Evidence
Freely move in arena looking for a ball	yes	The robot continuously moved forward while scanning the field using the camera until the ball was detected.
Detect and avoid obstacles using bumper sensor	yes	MyBoT detected four separate obstacles during runtime and successfully reversed and turned after each contact: • 1st contact at 00:00:20.480 • 2nd contact at 00:00:39.776 • 3rd contact at 00:00:56.288 • 4th contact at 00:01:28.032
Identify soccer ball using RGB camera	yes	Upon identifying the soccer ball, the robot printed "Soccerball found. Moving towards it!" and started to align and approach.
Stop upon reaching the soccer ball	yes	Robot halted at timestamp 00:01:58.144 after recognizing proximity to the soccer ball. Message printed: "Soccerball reached! Objective completed! :)"

The insights gained from the results, validate that the project achieves 100% target-acquisition success in the presence of up to five randomly placed obstacles while maintaining sub-centimeter final-position accuracy. MyBot demonstrated overall robust performance where the touch sensor and camera modules work in sync, and the control loop properly handles sensor feedback to achieve the objective.

Conclusion

The project was successful in achieving the core objectives initially outlined. MyBot was able to autonomously navigate the arena, effectively detect and respond to obstacles using the touch (bumper) sensor, and successfully locate, approach, and halt at the soccer ball using the RGB camera's recognition system. Hence, the proprioceptive feedback and visual recognition for target tracking is effectively implemented and the robot's behavior aligned with all four success criteria defined.

However, some shortcomings were observed.

- The robot's obstacle avoidance behavior relies on simple reactive strategies resulting in suboptimal recovery trajectories. This increases navigation time. Furthermore, reliance solely on bumper sensors for obstacle detection introduced latency in obstacle avoidance, since objects were only detected upon physical contact rather than preemptively. For future improvements, integrating exteroceptive sensors such as LIDAR or ultrasonic rangefinders could vastly enhance obstacle avoidance efficiency by enabling pre-contact detection and smoother path planning. In addition, combining the camera-based recognition with ultrasonic or infrared distance sensors would improve obstacle detection and avoidance strategies, especially when the touch sensor may be too reactive or insufficient in more crowded or irregular spaces.
- The ball approach behavior could benefit from more sophisticated localization techniques rather than reliance on single-frame image coordinates. For future improvement, the implementation of a rudimentary SLAM (Simultaneous Localization and Mapping) system, distance sensors, or an odometry-based dead reckoning mechanism, could drastically enhance navigation efficiency and robustness.
- Additionally, simulation tuning for parameters such as camera field of view, robot center of mass, and wheel friction properties would ensure better real-world transferability, which is essential when transitioning from virtual to physical robot platforms.

An actionable recommendation for anyone replicating or extending this project would be to employ a hybrid sensing approach. Combining tactile sensors with vision and proximity sensors can dramatically enhance environmental awareness, resulting in more efficient navigation and higher mission success rates. Emphasizing sensor fusion techniques and closed-loop control strategies would lead to a more robust and professional robotic system. Doing so would significantly increase the robot's environmental awareness, allowing it to avoid obstacles proactively and improve the overall task efficiency.