FIRE RESCUING ROBOT

COGNITIVE SYSTEMS AND ROBOTICS

APRIL 18, 2023

ASHIKA SRINIVASAN

Table of Contents

Table of Figures

INTRODUCTION

The purpose of fire-rescuing robots is to carry out various tasks, such as looking for survivors and supporting human firefighters by sending information to the receiver about the location of the fire and the number of people trapped inside the commercial building. This information is divided into two groups: those who were less and more severely affected by the fire. Search and rescue missions are one of the main uses for fire rescue robots. Webots allow engineers to simulate, model, and programme a robot's behaviour, including how it will interact with its surroundings.

LITERATURE REVIEW

Robots capable of executing rescue operations from flames have the potential to save lives and reduce the risks firefighters must incur. Researchers have recently focused on improving their ability to traverse scorching terrain, locate victims, and extinguish fires. According to a study report, since convolutional neural network (CNN)-based algorithms have a detection accuracy above 90%, they have been used most frequently in recent years to monitor fires. The latter demonstrated improved classification accuracy compared to the traditional approach and lowered the time cost in particular architectures compared to the backpropagation gradient, which only has a 50% accuracy rate. These models recognise objects and forecast their objectness ratings at each place simultaneously. They have a variety of sensors and instruments at their disposal that enable them to go over dangerous terrain, spot and put out fires, and find and save people and animals. The most current developments in fire-rescuing technology include robotic legs, autonomous robots, and collaborative robots. These robots can navigate rocky terrain and avoid obstacles within burning buildings. Also, it may act as a path guide under normal conditions. Future robots that can spot fires before they spread might significantly lessen the risk of harm to people by aiding firefighters. Robotic firefighters are used to finding the imprisoned people there and leading them to safety by navigating the structure. In general, firefighting robots can completely change our rescue and firefighting operations. These devices can save lives and defend communities by lowering the hazards to human firefighters and enhancing the effectiveness of these tasks.

DESCRIPTION OF IMPLEMENTATION

a) Actual interaction

The confrontation took place in a multi-story structure with a simulated fire. The robot was made to help firefighters rescue those trapped within the structure while reducing their exposure to the dangers of fire and smoke. The Crazyflie robot is a tiny quadcopter built-in Webots as a 3D model that can be used to imitate its behaviour in a virtual setting. It has a quadcopter body with four rotors and a battery and can be controlled using a variety of sensors and actuators. Several factors are adjustable to adapt the robot model to the physical features, such as the mass, drag, and rotor characteristics. Also, it may be flown in various modes, including manual mode, altitude hold mode, and GPS navigation mode.

Once the simulation had started, a Crazyflie robot with a high-altitude node started signalling to conduct the wall following for room locating and finished with the fire detection. It had a small screen displaying the robot's camera feed and controls for its movements. In some circumstances, the robot can even serve as a path guide. It was created to discover a fire with a red colour before it rages out of control. It will identify objects with a green colour as belonging to persons who are less afflicted and objects with a blue tint as belonging to those who are more impacted. Following the separation, it will transmit the number of fires and persons to the receiver as a pedestrian robot to behave like a human pedestrian. Pedestrian robots may be employed for many purposes, including transportation, service robots, and security. I switched the keyboard control inputs to automated ones for this project. A GPS node was further inserted to help locate the fire and the trapped victims. Emitter and receiver can be used in conjunction to build sophisticated interactions and behaviours among various simulation elements.

b) Functionality/Improvements and Results

The content describes the functionality of the Robot class, which controls a drone's sensors and actuators. The class creates nodes for GPS, IMU, Gyro, Camera, and LED and uses PID controllers to determine the velocity of each propeller motor. The setVelocity method sends orders to the motors, and the getImage() and getRecognitionNumberOfObjects() functions collect images and recognise objects. The code first calculates the time since the last measurement (dt) and then reads the range sensor value (range_front). The programme then calculates the robot's velocity in the x-y plane relative to its current orientation with global x-y velocities (vx_global and vy_global), the current yaw angle (actualYaw), and initialises the forward, sideways, and yaw desired velocities to zero. Here, I constructed a distance sensor with the "wb_distance_sensor_get_value" function, which returns the voltage or intensity of the measurement.

The following if statements use the sensor data to determine the proper robot velocities. To steer the robot away from an obstacle if it is too close to one, the desired yaw and sideways velocities are set to negative values in the code (i.e., range_front 500). Suppose the robot is too close to an obstacle to its left (i.e., ds_L 70000) and very close to the obstacle in front of it (i.e., range_front 100). In that case, the code sets the desired forward velocity to a negative value and the desired yaw and sideways velocities to positive values to steer the robot to the right and away from the obstacle, and so on. If none of these conditions is met, the code sets the required sideways and forward velocities to positive. To enable the control programme to make decisions based on the data, a Crazyflie robot's emitter node will transmit GPS data to a pedestrian robot's receiver node by setting a channel to 0. Screenshots of the outcomes are attached.

c) Process flow

i. Environmental setup

A furniture and appliance store may be found in the virtual world. It comprises a couch, a cabinet, a 10-by-10-inch floor, a parquet-style appearance light strip with the colour 111 overrides, a caution sign with the dimensions 2 by 0 by 10.0, and a light strip with the colour 111 overrides. Translation, rotation, and size changes are made to the wall, crazyflie robot, and pedestrian robot under the design. A geometry cone was set up with the following dimensions: a modified width of 0.9, a radius of 0.34, and recognition colours of 1, and the same for the blue and green capsules. Robotic pedestrian with an altered shirt, trousers, and shoe colours and translation and rotation of -13.7 11.2 1.37 and 0 0 1 1.31, respectively.

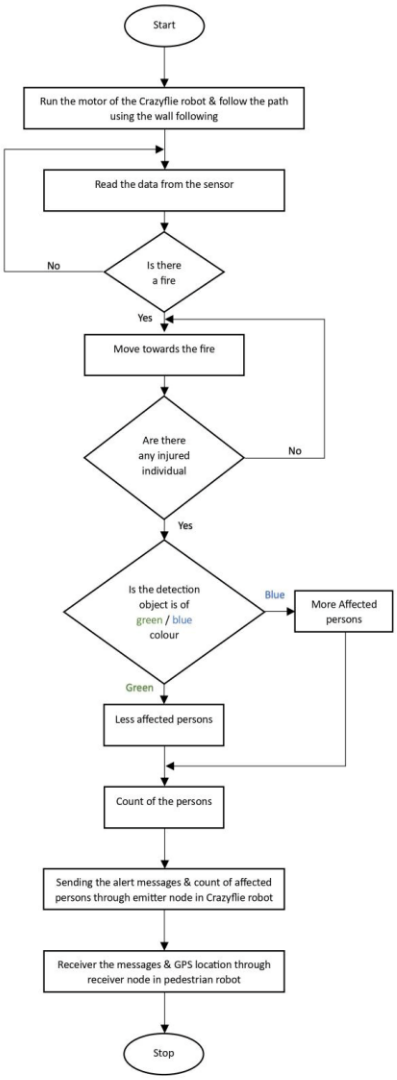

Fig. 1 - Flow chart

ii. Design Ideas and Extra Implementation with AI

Integrating robotics and artificial intelligence using the A* algorithm as the search method. The A* algorithm is preferred because of its heuristic-based search, which allows it to evaluate a map and produce the quickest path. The system will use a binary occupancy map representing the building's basic structural layout, which may be produced through CCTV or drones. The accuracy and completeness of the occupancy map are essential for correct path planning. Overall, the idea is promising and requires accuracy in occupancy map production.

Also, we can use a Random Forest classifier on the training data. The trained model will recognise patterns in sensor data that will be used depending on the specific requirements of the firefighting robot to detect the presence and location of fires.

I experimented with mapping and investigating a building's layout with a Crazyflie. The sensors and mapping methods will determine the code needed for this complex process. However, I attached a Python code snippet that shows how to use a Crazyflie to carry out straightforward mapping tasks using a camera and a distance sensor. Attached the mapping code.

```python

def run_robot(robot):
    print("newcontroller")
    # Get the camera and distance sensor
    camera = robot.getDevice('camera')
    # distance_sensor = robot.getDevice('distance_sensor')
    dis0 = robot.getDevice('ds_F')
    # Enable the sensors
    camera.enable(10)
    dis0.enable(10)
    # Set up the map
    map = []
    # Set the initial position of the robot
    x = 0
    y = 0
    # Fly the robot around the building
    # while robot.step(10) != -1:
         distance = dis0.getValue()
        if distance < 0.5:
           image = camera.getImage()
           map.append((x, y, image))
        # robot.mode(0.1, 0, 0)
        x += 0.1
    # Save the map to a file
    with open('map.txt', 'w') as f:
        for entry in map:
            f.write('x: {}, y: {}\n'.format(entry[0], entry[1]))
            f.write(str(entry[2]) + '\n')
```

Fig. 2 - Mapping

SCREENSHOTS

Fig. 3 - Receiver and environment

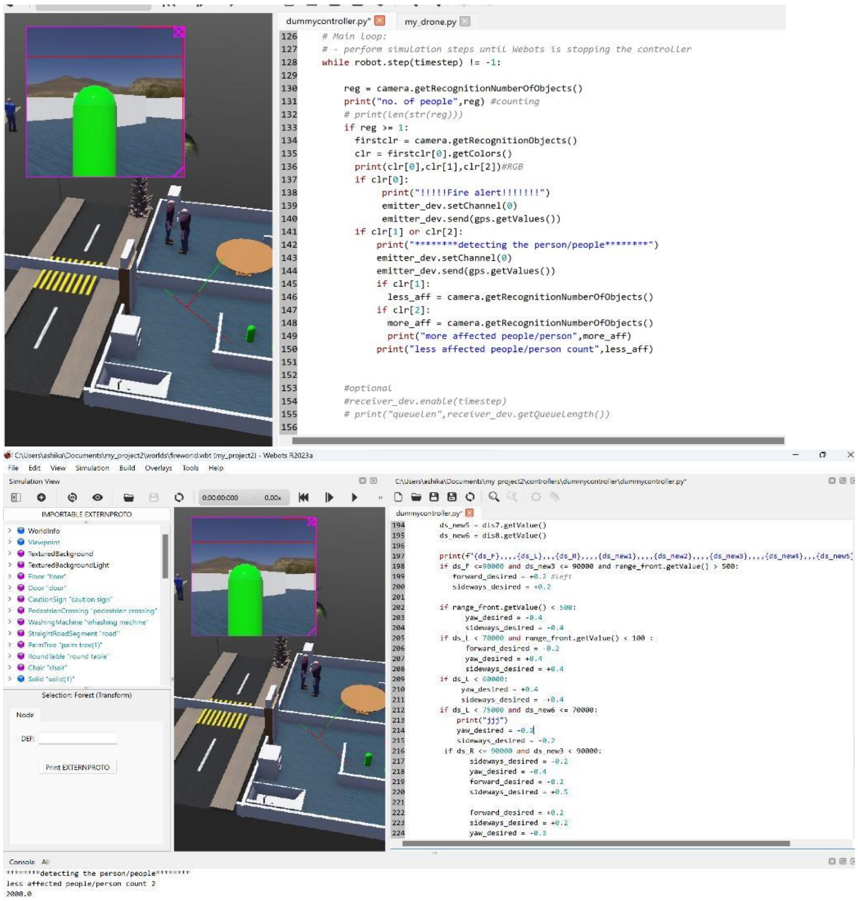

Fig. 4 - Camera recognition and object detection

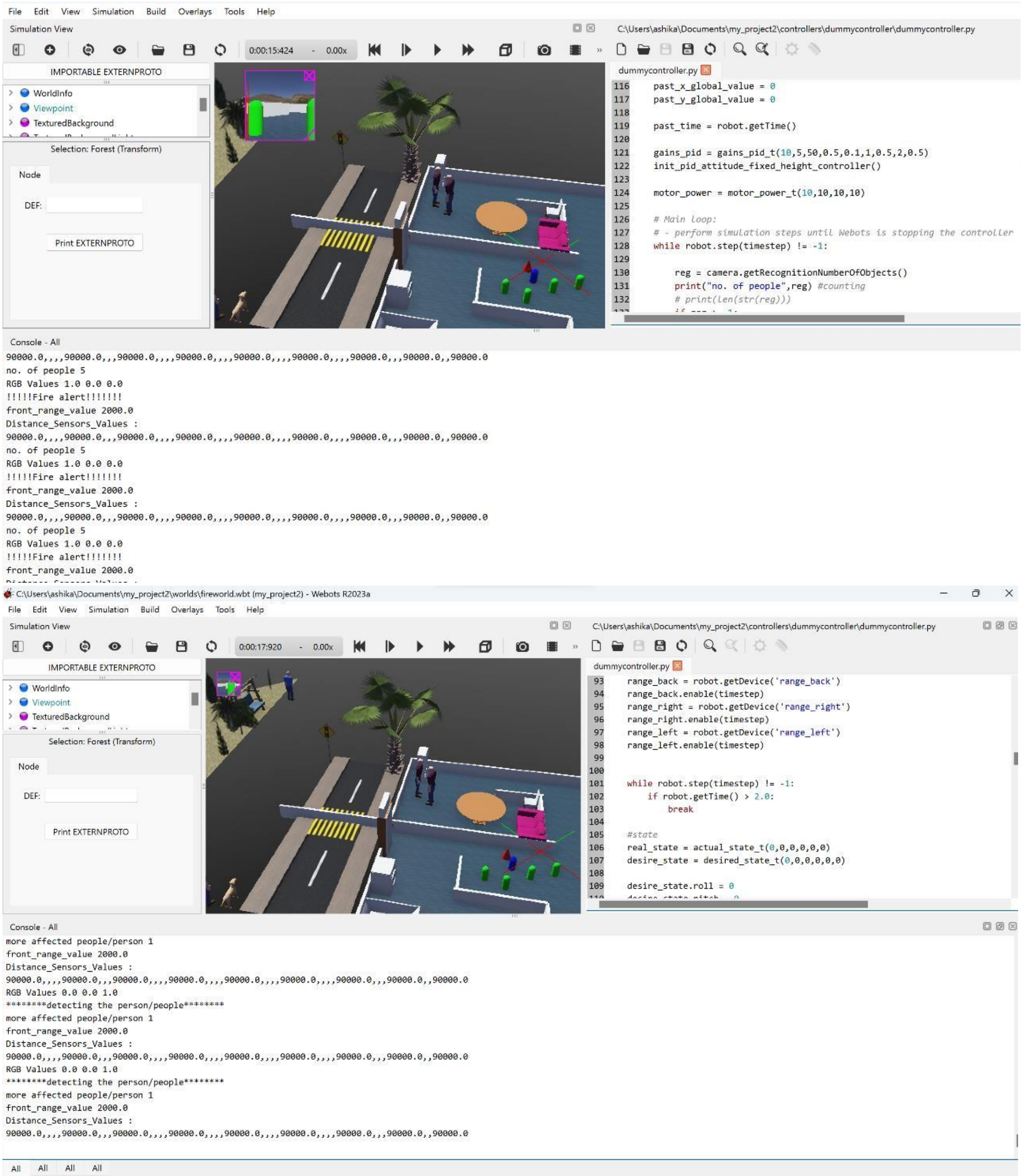

Fig. 5 - Results

CHALLENGES IN VIRTUAL ENVIRONMENT

The programme requires much time learning and comprehending its features and functionalities. Webots are limited to simulating robots and their environments as software simulators. Hardware constraints in the real world are not considered, which might lead to differences between simulation findings and actual outcomes. The documentation could be more sparse and more updated. As a result, finding the information needed to address problems is challenging. And integrate Webots with other software. It supports a wide variety of programming languages and libraries. However, it is necessary to change their code to function as Python could have certain speed and memory management restrictions. Due to Python's less complete support for complex physics simulations and visualization capabilities than C++, it will not work well with particular libraries and functionality. Additionally, several of the Webots API's components were created in C++, which makes utilizing python more difficult.

ADVANTAGES

- With Webots, designing and testing robots in a virtual setting and 3D simulation environment is possible before using them in the real world.
- This user-friendly interface provides a library of pre-built robot models, sensors, actuators, and other easily customizable parts and is used to construct sophisticated robotics applications.
- It is an open-source programme that is an inexpensive and versatile program for robotics research since it is free to use, modify, and share.
- It supports various sensors and actuators used by the Crazyflie robot.
- Functionalities mimic various situations and test various control algorithms using various sensors.
- To build a unique simulation environment for the Crazyflie robot, customizing the Webots environment is adaptable by adding or removing components or nodes.

DISADVANTAGES

- Webots have limitations in customising the simulation environment beyond what is already provided in the software.
- The graphics quality could be better than other simulation software and can be slow, especially when simulating complex robots or environments.
- It is not easy to precisely control the robot's movement in the simulation environment, making it challenging to test and develop robot algorithms.
- It is relatively small compared to other popular robotics software, making it challenging to find support or resources to help solve problems.
- The inability to mimic battery life poses a serious obstacle to testing drone control algorithms that depend on it.
- Since wind impacts are not simulated, simulations may not evaluate drone control algorithms in windy conditions.

- It offers a realistic simulation environment, but it cannot replicate every possible contextual condition that could impact how well the Crazyflie robot performs in the real world. The simulated results and the actual implementation may differ as a result of this constraint.

USE OF ROBOTS AND SIMULATION IN THE REAL WORLD

The potential uses of the Crazyflie robot in various fields. It can search for missing or trapped individuals in disaster areas and inspect hard-to-reach infrastructure using sensors and a camera. In agriculture, the robot can monitor crops, identify issues, and collect information using cameras and sensors. The robot's small size and maneuverability make it an ideal tool for these applications. The benefits of using fire rescue simulators to train firefighters and emergency responders may be applied to a real-world challenge. These simulators provide a safe environment to practice firefighting methods, decision-making, and communication skills. Also, allow for the testing of new firefighting tools and technology. Using these simulators can enhance the readiness of emergency responders and improve their abilities to handle fire situations, potentially leading to faster response times.

The Crazyflie robot can improve productivity and reduce expenses by performing tasks more precisely than humans. It can be used to check dangerous areas and collect data that may lead to new inventions and understandings. Fire rescue simulators can improve the abilities of first responders, resulting in faster reaction times, more successful rescues, and fewer fatalities and damages. It can also be used to test and enhance firefighting technology. Simulated fire rescue operations can foster collaboration and knowledge exchange among emergency personnel, strengthening alliances and increasing cooperation during emergencies.

The Crazyflie robot needs complex control systems, sensors, and algorithms to operate safely and efficiently in agriculture. Robotics experts are required to design the robot's hardware, software, and control systems, while sensor technology experts are needed to integrate sensors that provide reliable data. Data processing and analysis professionals are required to develop algorithms that can analyse the data gathered by the robot to optimise agricultural processes and reduce water consumption. Together, these experts can create and operate the robot to improve agricultural yields and resource efficiency. The development of realistic fire simulations requires multidisciplinary support from various fields, such as computer graphics, physics, and chemistry. Biomedical engineering is necessary to develop better equipment for firefighters, while data science and statistics can help evaluate the effectiveness of different rescue strategies. A multidisciplinary approach is required to comprehend and handle the complexities of fire rescue operations.

CHALLENGES IN IMPLEMENTING SOLUTIONS BASED ON AI & ROBOTICS IN REAL PROBLEMS

Based on the information gathered from its sensors, the robot must act quickly and decisively. Implementing an AI-based system that can analyse the data and make judgments in real-time takes work. To undertake coordinated tasks, robots, and human operators must instantly communicate with each other. Integrating numerous sensors, such as thermal imaging cameras, gas detectors, and lidar sensors, can be not easy. It is essential to ensure the security of the robot and the humans it saves. High heat, smoke, and other dangers must be considered when designing the robot. Additionally, it must be designed to avoid doing anything that can endanger others, such as unintentionally knocking someone over. Another major obstacle is ensuring the robot's price is reasonable and fair.

ETHICAL IMPLICATIONS

Using cameras and sensors for fire rescue operations carries privacy risks. Policies and procedures must be established to protect customer privacy. The public should know the technology's intended usage, advantages, and risks. To prevent discrimination, the robot must be designed and used unbiasedly. Concerns about trust between the retail establishment and its customers may arise from employing technology in fire-rescuing applications. The use of technology and the safeguards put in place to ensure the security and privacy of customers must be made transparent and clearly understood. Using technology in fire rescue applications can increase safety. However, it's crucial to ensure the technology is dependable and efficient in spotting potential fire hazards and warning customers and personnel of the threat.

CONCLUSION

In conclusion, creating a robot that can save people from a fire is a difficult endeavour that necessitates careful planning and considering various elements. The robot must be built with the ability to move through a hazardous area, locate the fire and potential victims, and locate and free any trapped people. This concept employs an autonomous firefighting robot capable of spotting flames and moving forward, left, and right. After the project was built successfully, the simulation was performed, and the intended result was reached. An autonomous firefighting robot has been developed to complete the project's goals effectively.

REFERENCES

Darminto, H. W. A., & Sandjaya, A. (2011). Conjugate Gradient Backpropagation Implemented On Obstacle Avoidance Wheel Robot In Webots.

Ghali, R., Jmal, M., Souidene Mseddi, W., & Attia, R. (2020). Recent advances in fire detection and monitoring systems: A review. In *Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18), Vol. 1* (pp. 332-340). Springer International Publishing.

Hasimi, M. S. M., Zuha, W. H. W., Shafie, S., & Marhaban, M. H. (2010). Fire Fighting Robot. In *Asia Pacific Symposium on Applied Electromagnetics and Mechanics (APSAEM 2010)*.

Jijesh, J. J., Palle, S. S., Bolla, D. R., Penna, M., Sruthi, V. P., & Alla, G. (2020, November). Design and Implementation of Automated Fire Fighting and Rescuing Robot. In *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)* (pp. 320-323). IEEE.

Prince, A. T. Z., Kabir, K. S., Nabil, M., Abdullah, R., SM, Z. A., Rafat, S. S., ... & Raza, K. Fire Fighting Robot.

Reinhart, J., Khandwala, V., Shah, T., Young, J., Ross, L., Delica, R., ... & Pollard, R. (2003). Design and Implementation of Fully Autonomous Fire-Fighting Robot for a Small House Model. In *16th Florida conference on Recent advances in Robotics*.

Riches, L., & Judge, M. Firefighter Assistance Robot.