The University of Sussex
School of Engineering & Informatics

MSc in Advanced Computer Science Candidate number: 260121

Title: Hurdle-avoiding Robot in Coppeliasim

Approximate number of words 4,000 Supervisor(s): Dr Chris Johnson

Candidate's signature: Yadav Priyanshi Yadav

Abstract-

A robot that can avoid hurdles was created, built, and programmed with the possibility of application in both education and research. CoppeliaSim is the greatest choice for simulation. It is open-source and cost-free. Lua is also gaining notoriety for its managed code architecture and simplicity of use in the field of robotics. Lua is used as our robot's controller language. It is an IDE without parallels for programmers. The Lua script is located in the linked functions of the CoppeliaSim fragment. The CoppeliaSim and our application communicate using the Remote API Framework. Whenever the vision sensor detects a signal, the built robot will proceed in that direction while avoiding any hurdles in its way. This work describes the construction, simulation models, and integration of a hurdle-avoiding robot with vision and proximity sensors.

Many successful attempts to design hurdle-avoiding robots have been made. The choice of sensors and the route mapping method used to establish the operational parameters vary amongst these works. The technology performed well in a variety of lighting situations. The versatility of the robot to avoid hurdles is demonstrated by experimental results with a variety of hurdle placements. The sensing element technology is incredibly cheap because it only employs one range-detecting element, which allows the robot to identify victims before it.

Keywords- bubbleRob robot, Coppeliasim, proximity sensor, force sensor, Unmanned aerial vehicles (UAVs), revolute joint.

Introduction-

The usage of robots in both the industrial and home spheres is currently popular. Robotic assistance is available from sunrise to night, either through an intermediary. Many robotic systems must perform the critical duty of hazard avoidance. For fixed base systems, the robot must avoid hurdles in its workspace along with self-collisions. Different robot kinds are used by us for multiple purposes. The crewless vehicle is employed in the industry production pipeline, and mobile robots are utilized in transportation such as food delivery. The term "non-mobile robot" refers to a robotic arm with some degree of adaptability. Typically, a six-degree-of-freedom robot is employed in the business for an array of tasks. It can be utilized for car painting, welding, pick and place, packaging, and many more. An integrated development environment is necessary for every robot research so that researchers may create and evaluate their algorithms.

In recent years, a variety of robots for automation and navigation have been created, including robots that can follow boundaries, corners, humans, and other hurdles. The robot that ignores barriers will avoid them to reach its operational objective. Hurdle-avoiding robots are crucial on the manufacturing floor due to the dependability, accessibility, and cost-effectiveness of deploying mobile robots in industrial and technical applications[15]. Unmanned aerial vehicles (UAVs), on the opposite side, are essential for both military and non-military purposes [14]. Applications for the armed services include telecommunications, observation, and surveillance. Similarly, crisis intervention, remote sensing, vehicle navigation, etc. are examples of civilian applications. Many UAV applications require the capacity to operate in urban settings or uncharted territory with hurdles of various types and dimensions. Autonomous UAVs must be able to identify and avoid impediments in their route as a basic requirement. The concept of the robot base employing proximity and vision sensors was proposed in this work along with an example of a hurdle-avoiding robot. The created robot can serve as a guideline for a variety of applications in academia, industry, and study.

In this study, we offer a method for detecting sudden hurdles that occur in a robot's route and for calculating the object's relative position to the robot. This program's goal is to successfully actualize hurdle detection from the robot with a practical implementation in mind. A comprehensive division of the hurdle is not necessary because the robot does not maneuver around it (which may be time-consuming). The technique is suggested for a robot proximity sensor that employs one wide-angle camera to navigate in a supportive atmosphere. A qualitative study of the situation determines the location of impediments.

Literature review-

Robots require a variety of sensors to gather data about their surroundings. Sensors will aid in determining the object's position, velocity, acceleration, and range inside the workspace of the robot. The range of an object is determined using a variety of sensors. The ultrasonic transducer is one of the most often-used range finders. Moreover, vision systems are used to significantly increase the robot's adaptability, speed, and accuracy for its challenging and complicated duty [15]. Obstacle avoidance technology with a pan-tilt-mounted vision system was developed by [9]. Here, a robot navigates a changing environment by using histograms of images captured by a monocular and monochrome camera.

A single camera mounted on a robot [2] demonstrated the identification of impediments (especially walking humans). Block-based motion estimation searches for objects that are present close to the robot. According to [4], blind and visually impaired individuals can employ mobile robot obstacle avoidance systems as a guiding tool. A robot's motor controllers receive electronic signals, and the blind traveller can navigate obstacles using aural signals. Three ultrasonic sensors were used by [3] in his alternate concept for a more affordable and straightforward obstacle avoidance robot. For those with physical disabilities, [5] developed a robot system that is capable of carrying out a variety of duties.

The created robot avoids collision with unforeseen obstacles by using an ultrasonic range finder for sensing and mapping. Self-controlled robots were created by [6] for military applications. It makes strategic adjustments based on the environment utilizing hurdle detection algorithms while employing GPS and a magnetic compass [15]. A system for hurdle detection during rotorcraft low-altitude flight was developed by [8]. Based on various rotorcraft motion limitations, the need for a hurdle detection system for rotorcraft in low-altitude flight is thoroughly examined [7] addressing the use of an IR sensor for smart distance measurement. They discovered from their investigation that one of the main shortcomings of IR-based sensors is their limited range of detection.

Robot Design-

The hurdle detector robot was implemented in CoppeliaSim in several steps. The scripting language used by CoppeliaSim is used to create the program that manages the robot's behavior. The application reads the range sensor values and determines how far away the closest obstruction is. The program stops the robot's motors and spins it until it finds a clear path if the distance is less than a threshold number. The robot resumes moving forward once a clear path has been identified. Firstly, two motors, two wheels, a force sensor, a vision sensor, and a chassis make up the robot. A simple sphere frame makes up the chassis. A slider is used to stabilize the sphere so that the motors and sensor may be mounted on it. The robot can travel in any direction due to the motors, which are connected to the chassis and drive two wheels. The front of the chassis is where the proximity sensor is located, which allows it to identify objects in the robot's route.

Next, the CoppeliaSim programming interface was used to design the robot's behavior. When a hurdle was discovered, the revolute joint was designed to rotate the end effector and navigate around it. The proximity sensors were programmed to identify hurdles in the robot's surroundings. The wheel's movement was also programmed to be controlled by the revolute joint, enabling it to avoid obstacles in its environment. The robot's actions were changed as necessary by the force sensor, which was programmed to measure the force used to manipulate objects.

Finally, the robot's actions were evaluated in the simulation environment to make sure that it could recognize obstacles, avoid them, and manipulate objects. Any problems or mistakes were fixed by doing more.

In this study, we offer a method for detecting sudden hurdles that occur in a robot's route and for calculating the object's relative position to the robot. This program's goal is to successfully actualize hurdle detection from the robot with a practical implementation in mind. A comprehensive division of the hurdle is not necessary because the robot does not manoeuvre around it (which may be time-consuming). The technique is suggested for a robot proximity sensor that employs one wide-angle camera to navigate in a supportive atmosphere. A qualitative study of the situation determines the location of impediments.

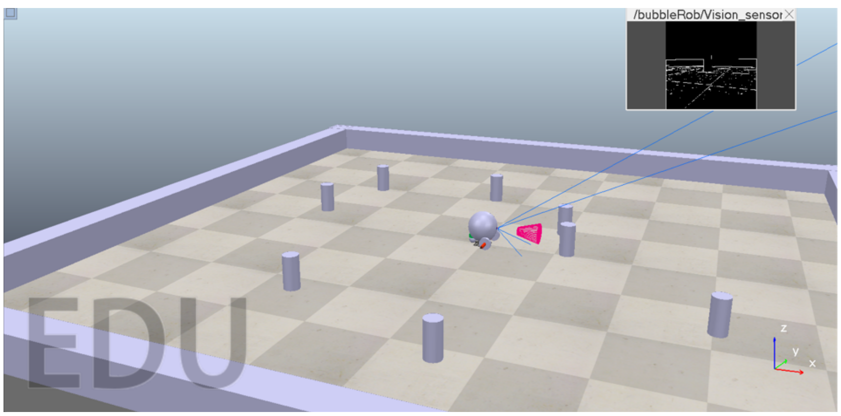

Figure1. bubbleRob robot in Coppeliasim

The ".stl" files can be imported into CoppeliaSim's scene, and all pieces can be oriented by the robot design. Using the level characteristics, all components are transformed into convex shapes and rendered invisible. The robot is configured to be collidable, quantifiable, traceable, and renderable throughout. The side wheels, motors, and slider are given revolute joints since motors are only permitted for the separation wheels. Vision sensors and a cone-shaped proximity sensor are added to the robot's front and base, respectively. A non-threaded child script (software) is connected to the system foundation for the chassis.

After assembly, the items are placed for the simulation in a specific hierarchy (as shown in figure 2).

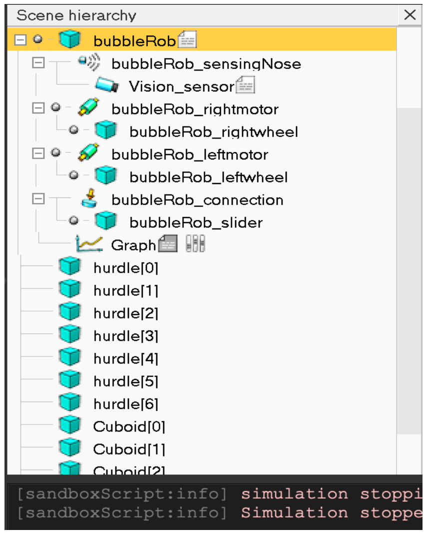

Figure2. Hierarchy scene of hurdle-avoiding bubble Rob robot

A hurdle-avoiding robot with a payload should have a body mass of 4.1894 kg, which is the mass that has been set for the entire robot. In Table 1, the appropriate principle moments of inertia/mass in the x, y, and z- directions are analyzed. A hurdle-avoiding robot benefits from having a lower moment of inertia since it requires less force for a body to conduct the translation-free motion.

Direction	Principle moments of Inertia/mass (m^2)
I_{xx}	4.000e-03
I_{yy}	4.000e-03
I_{zz}	4.000e-03

Table 1. Principle moments of inertia/mass of the bubbleRob

Program Instructions-

The robot may be dynamically assembled using CoppeliaSim, incorporating components like revolute joints, motors, sensors, and even embedded scripts of bubbles and sensors to connect the robot's code to its physical components. Webots, Gazebo, and RobotStudio are a few more robot simulation programs that do not permit engineers to watch robots remotely or double-check their safety. Programming languages supported by CoppeliaSim include C/C++, Python, Java, Lua, and Matlab. The software also enables the integration of robots created by other outside firms [7]. This simulator not only makes it simpler to construct a manufacturing environment and a robot, but it also allows the program to execute in compliance with the model's features. Lua is the programming language used to create the robot's code. Software called CoppeliaSim includes[16]. Lua, a programming paradigm language that is robust, quick, and lightweight. Here, a non-threaded child script is deployed, and it executes in the exact order that the user specifies. There are three key sections of the code:

- Initialization-Here, the items are separately obtained utilizing "sim.getObjectHandle" API and values are placed in variables like the robot's speed. SimUI.create was used to develop the user interface, which in this case included choices to pause and resume the robot as well as to reduce and raise its speed [8].

- Actuation: The different sensors and motors that were initially initialized in this section of the code are executed and put into action. According to the identified path, the wheels' varied speeds are calculated. The code is generated based on the ideas of the hurdle-avoiding robot. The sensors' and motors' variables are called and specified.
- Additional functions: These are essentially UI initialization-related functions.

Sensors-

The most important part of the robot's activity to avoid obstacles is played by its sensors. For this particular simulation, a cone-shaped proximity sensor, a force sensor, and vision sensors are employed. If the path is blocked or otherwise hindered, the proximity sensor enables the robot to avoid a collision. The color of the route is detected and distinguished from the environment using the vision sensors [9].

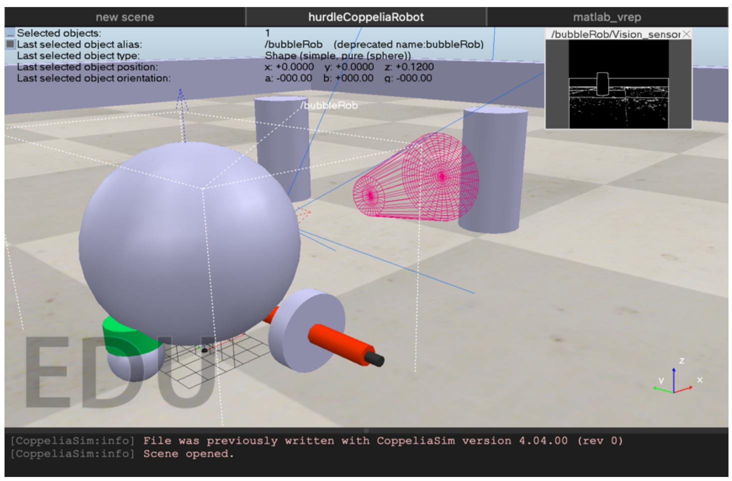

Figure3. Proximity sensor detecting hurdle

For the simulation, the proximity sensor is configured to be hidden. The cone measures 0.5 meters in radius, 1 meter in range, and 90 degrees in tilt. All items must be set to observable for the collision simulation for them to be comparable to this sensor. When this sensor detects a hurdle, both wheels drive backward or in any direction where they can avoid the hurdle at varying speeds.

By producing ultrasonic waves, as a proximity sensor, an ultrasonic sensor can be utilized to control the robot's range from a barrier.. It transforms the reflected audio frequency to electrical signal, which is subsequently used to determine how far away the robot is from the target[15]. The transmitter and receiver are the two basic components of the sensor [9]. The sound wave is emitted by the transmitter and returns to it at the receiver. For optimal safety in the production setting, these sensors can quickly identify the surface of an obstruction and carry out collision avoidance algorithms [15].

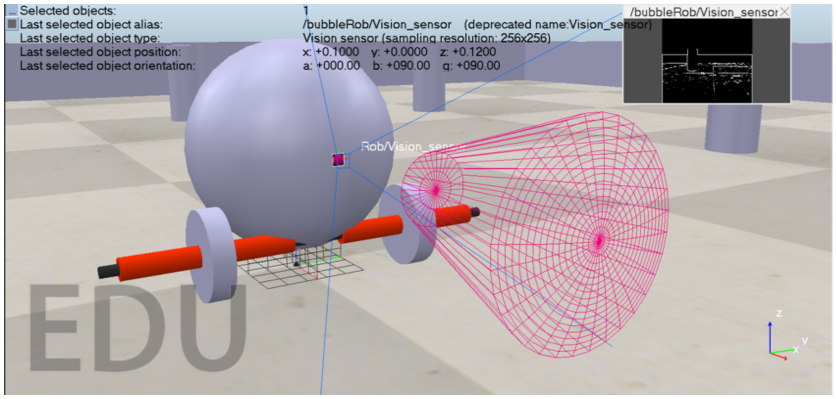

Figure4. Vision sensor in the bubbleRob

The most crucial elements of the robot architecture are the vision sensors, which are depicted in figure 7. These sensors aid in route recognition, which enables the robot to advance. The perspective style and 256x256 resolution make it easier to see the color underneath. The distant clipping plane, which corresponds to the code directing this robot to travel straight, turn right, or turn left appropriately, is 0.010 meters for near and 10 meters for a far plane.

The force sensor can be used to simulate the force between two hurdles or to monitor the force exerted on a robot's end effector. Also, since sensors are often lightweight and dependable, the robot's weight, inertia, and torque are unaffected. In this scenario, 0.050 meters size of the force sensor is attached to the bubbleRob_slider. Here force sensor works by simulating the interaction between the bubbleRob and the bubbleRob_slider.

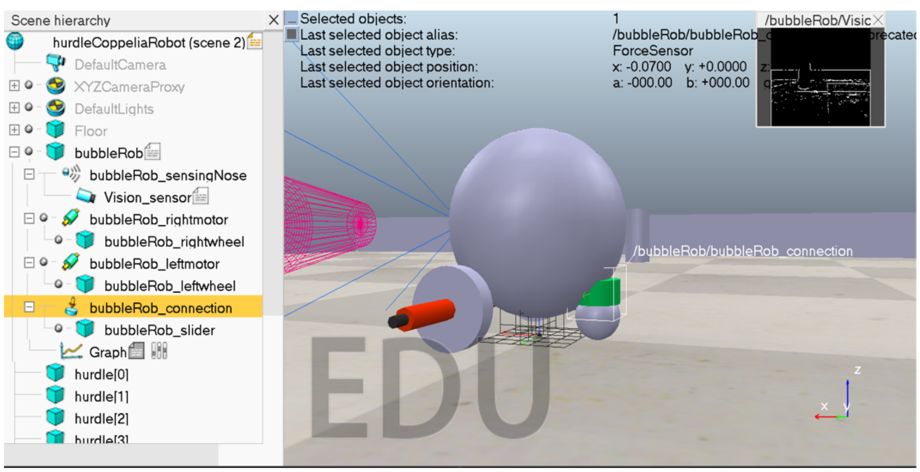

Figure5. Force sensor on the top of bubbleRob_slider

In this scenario, similar to a camera, vision sensors are used to capture images of the ground beneath the robot. Three of these sensors must be used in a set to segment the line to be followed from the factory floor. The IR (infrared) photodiode operates on the basis of reflecting infrared light from the incident surface, according to Planck's radiation equation. These detectors respond to infrared radiation, as their title suggests[15]. When a white surface is recognized, IR sensors reflect, providing the essential inputs for the hurdle-avoiding operation to occur.

Simulation and Results-

With CoppeliaSim, this robot's simulation is run. The built-in physics engine of CoppeliaSim was used to model the robot's behavior. The simulation's setting was a straightforward maze with various hazards scattered about. The robot was put at the beginning of the maze and instructed to make its way to the end without running into any hurdles. We may use the program to examine the robot's dynamic characteristics and give each component a unique actuation code. The fundamental ideas that guided this simulation were:

- Both wheels of the bubbleRob will advance at the same speed if the middle vision sensor detects a barrier.
- The left wheel of the bubbleRob travels with a slower (0.09v) motion when the left sensor detects a hurdle, guiding the bubbleRob to move to the left side.
- The right wheel of the bubbleRob travels with a slower (0.09v) motion when the right sensor detects a hurdle, guiding the bubbleRob to move to the right side.

As a result, the bubbleRob is now able to avoid obstacles, turn to the left or right, or even in other directions. The robot can move by identifying obstacles and its environment with the help of the vision sensors. A horizontal section with moves to the left and the right results in a closed path. To verify the image is recognized, moreover, a floating view is included that is able to show the front view of bubbleRob in a screen, and connected to the robot's center visual sensor.

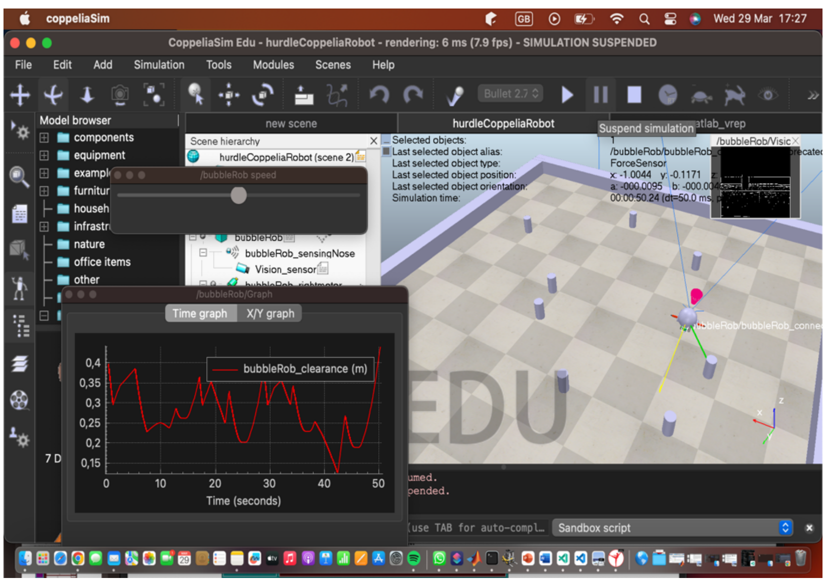

Figure6. Top view of simulation of hurdle-avoiding Robot in Coppeliasim

In Figure 6 the yellow and the green line are showing that robot can detect the distance between itself and the hurdle. Due to the bends and turns of the manufacturing floor-designed track, the robot travels along it at varied speeds. The path being followed affects the linear and angular velocities. The robot can navigate itself and move around the surroundings effectively thanks to the simultaneous operation of all sensors and motors.

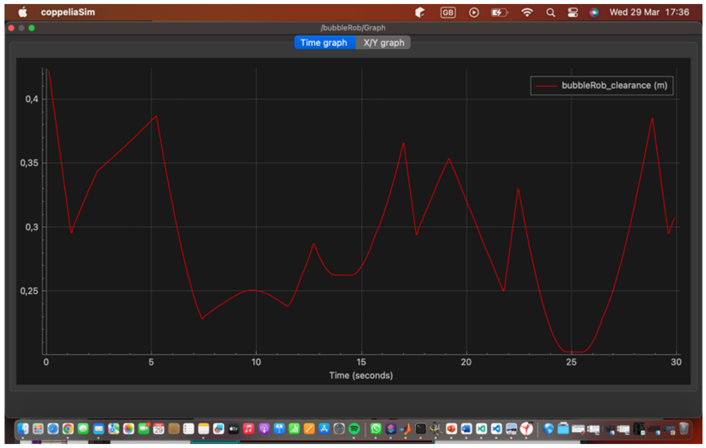

Figure7. hurdle avoidance simulation graph in Coppeliasim

This graph shows the progression of the robot's distance from environmental hurdles over time. Peaks on the graph show instances where the robot encountered obstacles and had to alter its course to avoid them. The valleys stand in for instances when the robot could proceed with certainty. The built-in physics engine of CoppeliaSim was used to model the robot's behavior. The simulation's setting was a straightforward maze with various hazards scattered about. The robot was put at the beginning of the maze and instructed to make its way to the end without running into any hurdles.

Speed of motor (Manually)-

We can add some programmes to determine the motor's speed. But we can manually determine the robot's speed because all of its characteristics have previously been determined.
The wheel's diameter is 7 cm in coppeliasim, allowing for the robot to be moved with the greatest degree of accuracy. The motor's rotational speed is 210 revolutions per minute.

- d represents the distance travelled by the motorized wheel.
- P represents the circumference of the circle (squared)
- N represents the number of rotations the motor makes
- t shows the Time required for the robot to travel a distance of 'd';
- speed of motor, v = d (cm)/t (sec)

Calculation-

- Wheel radius (R) = 8 cm/2
  = 4 cm
- The circle's circumference= (2ʌr)= 2*3.14*4
  =25.12cm
- Rotation per Minute = (1/60) Rotation per Second
- So, Rotation occurred by motor per Second = 240/60
  =4 RPS
- The length travelled by the wheel mounted to the engine per second= 25.12*4
  =100.48cm.

So, the speed of the motor is 100.48cm/s.

Conclusion and future work-

The obstacle-detecting robot created and simulated in CoppeliaSim, in conclusion, shows the viability of using range sensors and autonomous programming to navigate through challenging settings. Many applications, including search and rescue, industrial automation, and military operations, may benefit from the robot's ability to avoid obstacles. More sensors and programming to enable more complicated behaviors could be added when the robot is developed further.

We have described a system in this research that can identify sudden impediments that cross a robot's route. The technology has been proposed for navigation in surrounding heritage-like hallways and is made to detect hurdles and calculate the robot's moment of impact.

These robotic systems may be a crucial component in streamlining a factory's operations and raising the degree of manufacturing quality. Nonetheless, safety requirements in this regard are crucial and should be assessed while taking all limitations into account. There were just 38 robot-related incidents in the US between 1987 and 2016 according to the US Department of Labor [1]. Despite the minimal likelihood of these mishaps, safety precautions should be taken to protect the workers and the affected apparatus. Laser scanners can also be used to ensure that robots and manufacturing floor employees don't collide. Each robot should be evaluated separately, and any potential risks should be verified frequently.

To verify the robot's effectiveness in carrying out the operation, the sensors and actuators must undergo numerous tests and revisions [8]. The study of simulation makes sure that the following robot, which has castor joints, proximity sensors, force sensors, and vision sensors, can move along a globalized path in a manufacturing setting. When the speed is changed, it is discovered that the robot can go far in a few seconds when taking into account the necessary twists and path adjustments.

Reference -

1. N. Mizuno et al., “Enhanced path smoothing based on conjugate gradient descent for firefighting robots in petrochemical complexes,” Advanced Robotics, vol. 33, no. 14, pp. 687–698, Jul. 2019, doi: 10.1080/01691864.2019.1632221.
2. K. Vichova, M. Hromada, J. Valasek, and F. Paulus, “Comparison Analysis the Use of Modern Technologies by Fire Rescue Service,”2020, pp. 0535–0541. doi: 10.2507/31st.daaam.proceedings.074.
3. S. Kirubakaran, S. P. Rithanyaa, S. P. Thanavarsheni, and E. Vigneshkumar, “Arduino based firefighting Robot,” Journal of Physics: Conference Series, vol. 1916, no. 1, p. 012204, May 2021, doi: 10.1088/1742-6596/1916/1/012204.
4. A. A. Sharafutdinov and A. Y. Timasheva, “Structural and intelligent scheme of navigation system of a ground-based mobile robot for forming a traffic route,” IOP Conference Series: Materials Science and Engineering, vol. 860, p. 012019, Jul. 2020, doi:10.1088/1757-899X/860/1/012019.
5. E. O. Kargapolova, V. v Kuleshov, and P. Yu Scuba, “Assessment of the Use of Robotic Equipment for Extinguishing Fires at Oil Refining Enterprises,” IOP Conference Series: Earth and Environmental Science, vol. 720, no. 1, p. 012086, Apr. 2021, doi:10.1088/1755-1315/720/1/012086.
6. H. Fan, V. Hernandez Bennetts, E. Schaffernicht, and A. Lilienthal, “Towards Gas Discrimination and Mapping in Emergency Response Scenarios Using a Mobile Robot with an Electronic Nose,” Sensors, vol. 19, no. 3, p. 685, Feb. 2019, doi:10.3390/s19030685.
7. A. Viseras, M. Meissner, and J. Marchal, “Wildfire Front Monitoring with Multiple UAVs using Deep Q-Learning,” IEEE Access, pp.1–1, 2021, doi: 10.1109/ACCESS.2021.3055651.
8. M. N. Kirubakaran, S. Arun Kumar, S. Sasikala, S. Gohithmugilan, and M. Muralidhar, “Towards Building Intelligent Robotic Systems to Enhance the Safety of Firefighters,” Journal of Physics: Conference Series, vol. 1997, no. 1, p. 012040, Aug. 2021, doi:10.1088/1742- 6596/1997/1/012040.
9. T. Tong, F. Guo, X. Wu, H. Dong, L. Ou, and L. Yu, “Global Path Planning for Fire-Fighting Robot Based on Advanced Bi- RRT Algorithm,” in 2021 IEEE 16th Conference on Industrial electronics and application.
10. Jagruti Chaudhari, Asmita Desai, S Gavarskar, MGM’s Jawaharlal Nehru Engineering College, “Line Following Robot Using Arduino for Hospitals”, 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), September 2019.
11. “Sensors”, Fierce Electronics (https://www.fierceelectronics.com/sensors/what-ultrasonic-(sensor)
12. Yitao Ding and Ulrike Thomas “Collision Avoidance with Proximity Servoing for Redundant Serial Robot Manipulators” IEEE Systems Journal PP(99): 1-11, September 2018 (15) conclusion literature
13. https://iopscience.iop.org/article/10.1088/1757-899X/1012/1/012008
14. 14“The Importance of Robot Safety on Your Factory Floor”, Genesis Systems, IPG Photonics (https://www.genesis-systems.com/blog/importance-robot-safety-factory-floor)
15. https://iopscience.iop.org/article/10.1088/1757-899X/152/1/012064
16. https://iopscience.iop.org/article/10.1088/1757-899X/1012/1/012008