Artificial Intelligence Power Line Maintenance System

By

Oluwafemi Fabiyi

Xu Gao

Kejia Ran

Pujing Yang

Supervisor: Professor Yuri Balasanov

A Capstone Project

Submitted to the University of Chicago in partial fulfilment

of the requirements for the degree of

Master of Science in Analytics

Division of Physical Sciences

December 2021

Abstract

High voltage transmission lines usually cover vast distances to transport electricity from power plants. The maintenance work of high-voltage power lines is performed by human workers, who are often called power linemen. It is an extremely dangerous job since linemen have to ride helicopters to reach the power lines and crawl down the energized wire for inspection and repair. Fortunately, it is feasible to replace human workers with autonomous robots. This AI maintenance system aims to integrate the robots and use them to perform step-by-step linemen's work. The system is built based on the simulation program, which is part of the industry standard.

Keywords: real-time system, computer vision, robot localization, motion planning, robot simulation

Executive Summary

Working on thousands of voltages of electricity while being suspended in the air, powerline workers rank among the top 10 most dangerous jobs in the United States (BLS, 2020). According to The Bureau of Labor Statistics, this position has a fatality rate of 19.2 per 100,000 workers. Besides the heightened need for more rigorous safety standards and precautionary measures, drones and autonomous mobile robots can be put in use to perform inspections and repairs to eliminate risks to human safety and business assets.

Starting with patrolling auto-drones, OpenCV was used to analyze the information extracted from the vision sensor. When target objects, which represented the damaged parts, were found, the target's location was calculated and sent to the robot repairing units. The system also integrated the improved Vector Field Histogram (VHF+) algorithm for robot localization and path planning to guide the robot to the destination. To manipulate the robot arm, the Python remote API was utilized to scan, locate, pick up and drop target objects, which simulated the process of resolving powerline faults with autonomous robots.

The system demonstrated collaboration between two autonomous agents: a drone in flight for inspections and a mobile robot for groundwork. It went further to showcase coordinated communication of the drone with a ground-based robot that navigated to a target position, having avoided all obstacles along its path to perform inspections and repairs.

Table of Contents

List of Figures

List of Tables

Introduction

R3 Robotics uses artificial intelligence (AI) and machine learning (ML) in robotics to solve real-world problems in industries, such as warehouse security, construction, pipeline inspections, wind turbine inspections, search and rescue, maintenance, and wildlife preservation. The R3 Robotics research labs in Texas and Illinois reflect centers of excellence that focus on robot simulations, robot design, physics, math and AI/ML robot software development. Powerline maintenance activities have lagged in the context of incorporating robotics technologies. R3 robotics aims to help the industry efficiently transit into a new era with a more intelligent and safer powerline inspection and maintenance system.

Problem Statement

The problem of injuries and death resulting from powerline maintenance has a 19.2% fatality related work rate per every 100,000 workers. A good starting point in addressing this would be to limit the direct involvement of humans such that at least 25 yearly reported total injuries could be averted. With the help of R3 robotics, an AI powerline maintenance system was designed to help replace human workers with autonomous robots. By implementing the system, one could save $60,000-$70,000 per headcount. More importantly, it reduces fatal injury cases by nearly 100%.

Research Goals

This solution to replace human workers is a robotics system driven by artificial intelligence and machine learning concepts. The system uses autonomous agents equipped with relevant peripherals and devices. With appropriate assumptions made, the goal is to simulate within a controlled environment and test the feasibility of the simulated repair process flow.

Scope

The system design was based entirely on a simulation environment. Thus, the inspection and maintenance tasks were simplified and represented with a blue target object. The system is designed to detect and remove the target object from the powerline. For the purpose of this research, a vision sensor and proximity sensor were used in the system. However, additional equipment such as thermo-graphic cameras and electric field measurement tools would likely be required in a real-life environment.

Background

The primary hardware used to build this system is autonomous robots. Autonomous robots are widely used today in various industries, including agriculture businesses, warehouses, hotels, etc. The main feature of autonomous robots is that they perceive the surroundings and navigate through unknown terrains independently. In terms of software, the design of this system relies heavily on computer vision, robot localization, path planning programs.

Literature Review

In “About the future of power line robotics,” Montambault and Pouliot shared their vision on maximizing the impact on power line inspection and Maintenance practices. They think robotics is part of the solution for an intelligent power line maintenance system considering the dangers involved in this process. The author of this paper also designed a prototype system using the mobile robot to move along the wire to perform inspection in 2007. However, the system is limited with repairing, which is not enough to complete the entire maintenance task. To successfully implement robotics with the inspection activities, computer vision technology is essential.

Computer vision deals with how computers interpret images or videos like human visual systems and perform automated tasks based on their understanding of the visual data. Some computer vision applications include face detection, image searching, artistic image conversion, etc. (Dadhich, 2018, p.7). Despite the many use cases, the application of computer vision in systems can be tricky. Some even say, "Computer vision is a passive technology" and requires "'clever' algorithms to interpret the images." (Yoshida, 2011 p.187) In 2014, researchers successfully designed a vision-based broken strand detection method for a power-Line Maintenance Robot. The system applied multi-classifier which consists of two support vector machines to classify the wires into normal wire, broken strand malfunction." (Song, 2014 p.1)

Robot localization and path planning are building blocks for robots to be autonomous. Such process is called SLAM (Simultaneous Localization and Mapping) by some researchers in this area. The robot starts by extracting features from the environment, for example, to know if it is in front of a door or a wall. The features serve the robot as a landmark and eventually turn into a map (Correll, 2020, p.144). Kalman filter and piratical filter estimator is often used for localization. Despite its robustness, the problem with the Kalman filter is that it assumes linearity throughout the model, which may not be true in real life. A more enhanced algorithm could be the VFH (Vector Field Histogram). This method "deals with situations that are problematic for purely local obstacle avoidance algorithms (Ulrich, 2000 p.1). After locating itself, it then plans its path to navigate through the environment. The two most common techniques used in this process are visual odometry (VO) and laser odometry (LO). VO is pretty self-explanatory, while LO "assumes the use of laser ranger-finder scanners instead of visual camera sensors. This may allow for faster reactions and more precise observations in lowlight workspaces, but the measurements are monochrome and may be applied only in cases where the environment representation only requires precise shapes" (Kudriashov 2020 p.3).

Hardware & Data

For the purposes of this experiment, data was collected by multiple simulation equipments, processed and transmitted between different robot objects in the simulated environment. Specifically, the drone flew over the power grid, captured the object location, and then transmitted the object coordinates to the mobile robot for path planning. At last, the robot arm collected the precise location data of the target object to perform object pickup.

Robotics Hardware

There were three types of sensors that the system utilized in this experiment. The system selected FastHokuyo URG 04LX LASER sensor for path planning data, Photoelectric distance sensor OID200 for distance data and Blackfly S USB3 for image data.

An essential component for Robot Path planning was the FastHokuyo URG 04LX sensor (*UST-10LX*, n.d.), which was considered an optimal laser scanning solution for obstacle detection and localization. It's an advanced tool that returns max scan distance and angle to help detect and locate objects along the path.

Although the model of proximity sensor was not specified for this experiment, Photoelectric distance sensor OID200 (*OID200 - Photoelectric Distance Sensor - Ifm Electronic*, n.d.) was selected as the demonstration example. This proximity sensor has a very long range due to time-of-flight measurement, reliable background suppression and color-independent detection. It also has adjustment dial function and IO-Link function that allows easy parameter setting and user-friendly communication.

Vision sensor model was also specified for this experiment. Blackfly S USB3 (*Blackfly S USB3 | Teledyne FLIR*, n.d.) was the final chosen model as the vision sensor example. The Blackfly®

S is packed with powerful features, enabling its users to easily produce exact images and accelerate application developments. This includes both automatic and precise manual control over image capture and on-camera preprocessing.

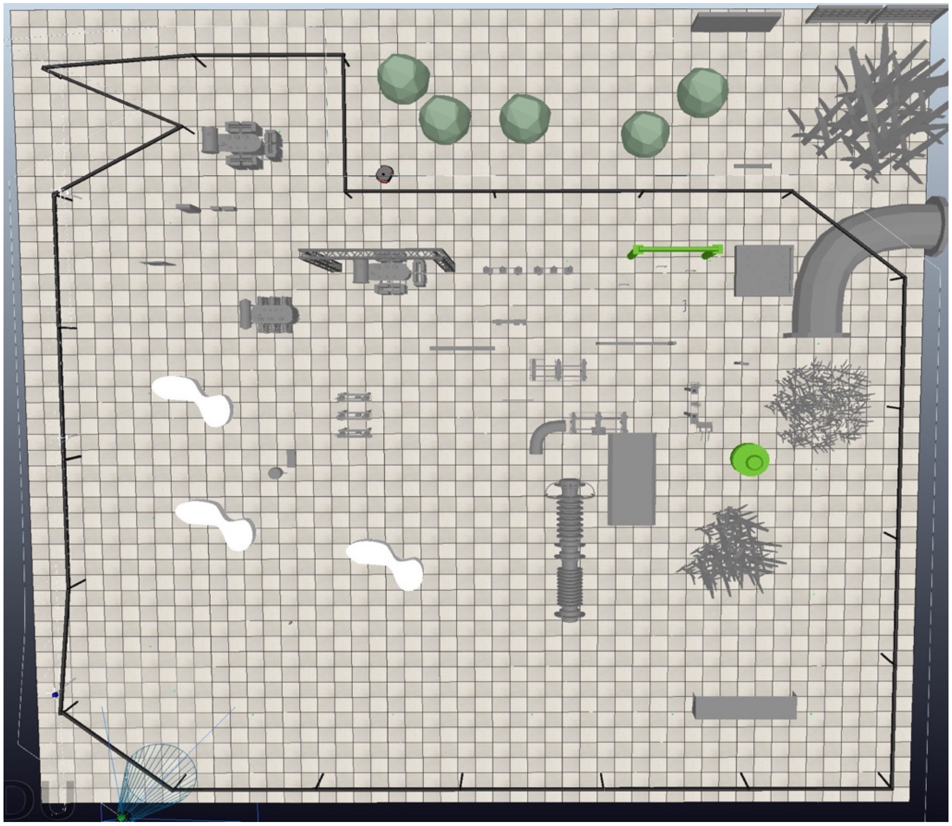
Data Collection

The drone flew over a preset route and captured computer vision data of the powerline every five seconds through its vision sensor. Image data were collected in a matrix form of pixels. Each pixel contained three values, which corresponded to three basic color channels: red, green and blue.

Proximity sensor data were collected by the sensors on the drone as well as the mobile robot. When the drone detected the object from the camera, the proximity sensor started to gather the relative coordinates of the target of the object. Combining the Global Positioning System (GPS) location of the drone with the relative coordinates of the target, the system was able to retrieve the estimated coordinates of the target object for the mobile robot.

The mobile robot used the proximity sensor to obtain the precise location of the target object. When the mobile robot reached the target, the robot arm scanned horizontally and calculated the final coordinates for object pickup. The proximity data were processed in a similar fashion to obtain the image data.

Figure 1. *Simulation Environment*



Coordinate data was shared through GPS of the drone in (X, Y, Z) and transferred to the mobile robot for motion planning. A summary of the data collected by the sensors in the simulated environment is shown in Table 1 and Table 2.

Table 1. *Independent Variable Summary*

Mother Device	Variable Source	Independent Variables
Drone	Vision Sensor	RGB Image Data
Drone	Proximity Sensor	Vertical Distance Data
Mobile Robot	FastHokuyo	Angle & Distance
Robot Arm	Drone	Rough Target Location
Robot Arm	Proximity Sensor	Horizontal Distance Data

Table 2. *Dependent Variable Summary*

Input Variables	**Dependent Variables**
RGB Image Data	Proximity Sensor Trigger
Vertical Distance Data	Rough Target Location
Angle & Distance	Speed
Angle & Distance	Yaw Angle
Horizontal Distance Data	Precise Target Location
Precise Target Location	Arm Yaw Angle

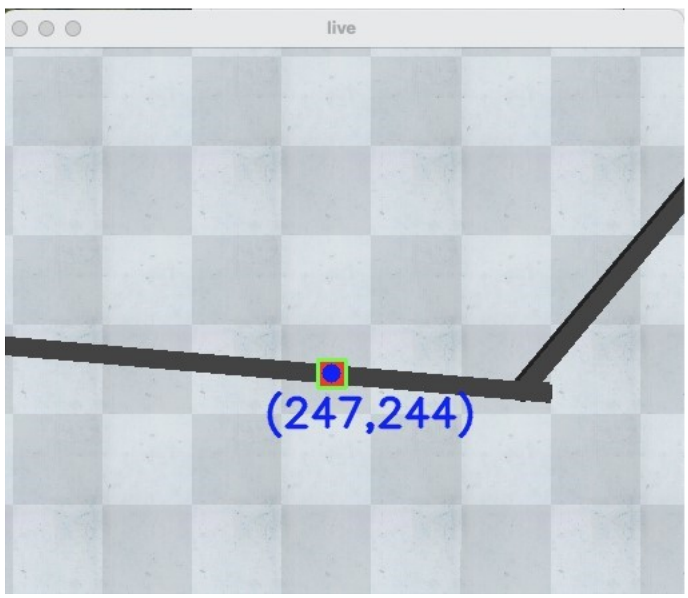
Methodology

Modeling Frameworks

The system needed to complete three major tasks successfully to achieve the final goal: identify and localize object(s) in the scene, move the pioneer robot to the object, and manipulate the object with a robot arm attached to the pioneer robot.

To identify the area of fault on the powerline (marked as blue in the simulation), a Python library, OpenCV, was used to process the data sent back by the sensors. OpenCV (*Open-Source Computer Vision Library*) is a free and open-source package of programming functions developed by Intel that solves real-time computer vision problems. While the drone was in flight over the power lines, its vision sensor would be collecting real-time image pixel and color data in RGB (Red, Green and Blue). In the meantime, the proximity sensor was detecting any objects in the sensor view and gathering the relative position data to the sensor itself. Both streams of data were communicated back to our Python program via API. The system was designed to use Gaussian Blur to first reduce the image size by removing noise and keeping most of the image intact. Then, it would convert the image from RGB to HSV (Hue, Saturation, Value) to better adapt to the simulated environment lighting for more accurate detections. Once the image was pre-processed, OpenCV would extract the blue color in the camera vision, draw a rectangular borderline on the blue-colored area and mark it as the detected object. While the drone was patrolling over the powerline, a list of object center coordinates was returned by the program. Only the pair that was closest to the center pixels would be selected as the final center coordinates.

Figure 2. *Live vision sensor image with identified object and its center coordinates*

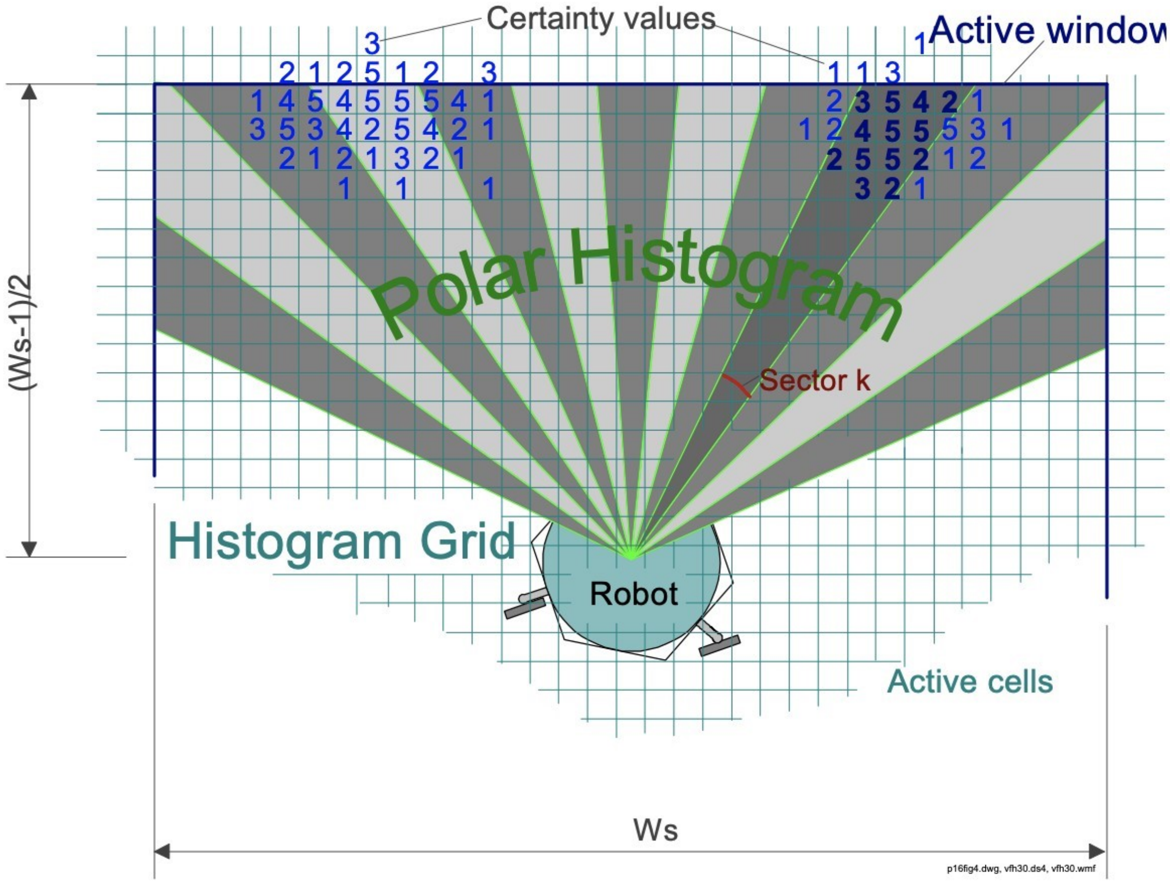


Once the object coordinates were obtained, the pioneer robot began to move towards the object location. The robot was expected to plan its path and avoid obstacles such as gas turbines on the way. The selected method to accomplish this task was called the improved Vector Field Histogram (VFH+), which enabled the mobile robot to detect obstacles along the way and avoid collisions while simultaneously steering itself to the target location. Developed by Ulrich and Borenstein (2000), the VFH+ model employed a four-stage data-reduction process in order to determine the next motion control for the vehicle. It first reduced the 2D histogram grid to a 1D polar histogram based on the pioneer robot's momentary location. Each polar histogram sector *k* contained a value that represented the polar obstacle density in that direction. Secondly, to avoid oscillations in the steering, a combination of primary polar histogram and hysteresis was introduced to the process to determine if the sectors were free or blocked. The third step aimed to build a masked polar histogram to indicate the certainty of a blocked sector and show which steering directions were possible at the current speed. If the masked polar histogram was blocked in all directions, the robot would have to stop moving immediately, so that it would not be trapped in a dead end. Finally, the algorithm built a cost function that considered three aspects of path planning: robot trajectory efficiency, variations in steering commands, and smoothness of steering commands. Only the relationship between the three aspects mattered, not their magnitudes. Hence, the following condition must be satisfied before running the robot:

Robot trajectory efficiency > (Variations in steering commands + Smoothness of steering commands)

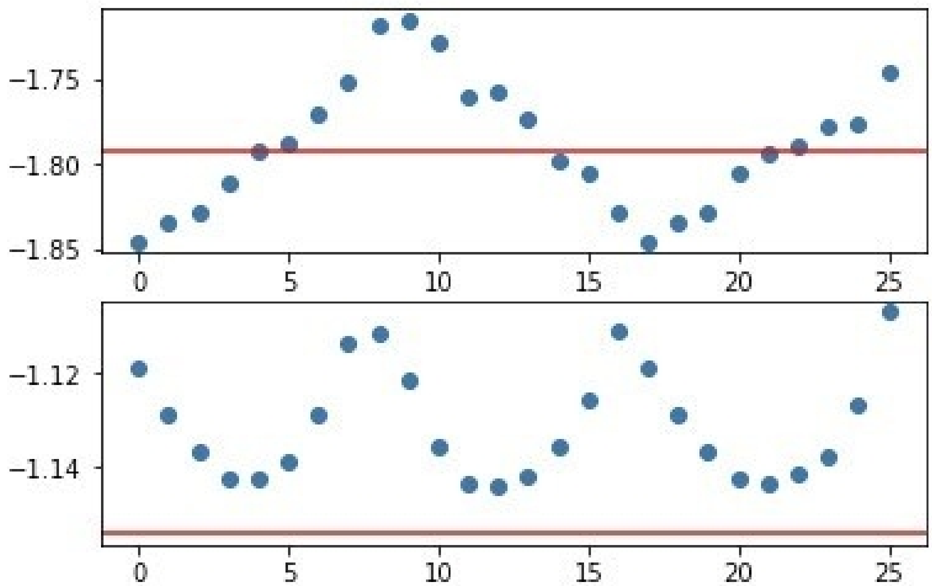
Depending on the task preference, the cost function could be modified to select the optimal direction of motion (Ulrich & Borenstein, 2000).

Figure 3. *Mapping of the robot onto the polar histogram.*



Finally, when the mobile robot arrived at the target location, its arm would pick up the object and drop it into the basket. To identify the object, the robot would have to scan the object with its proximity sensor, calculate the mean x value and the minimum y value to generate the (X, Y) coordinate. As the z value was already defined as the height of the power line in advance, (X, Y, Z) of the object could then be shared with the robot to perform the arm activities.

Figure 4. *Plotted estimated (x,y) coordinates, take the mean and minimum respectively*



Findings

Two path planning algorithms were considered during the project, namely Potential Field and an improved Vector Field Histogram (VFH+). It was observed that with the Potential Field algorithm, the robot took longer to navigate its way out of complex obstacles scenarios and got stuck in some instances, as indicated in figure 5 below. This wasn't the case for the improved Vector Field Histogram algorithm (VFH+), which is an improvement on the Virtual Force Field (VFF) and Vector Field Histogram (VFH) algorithms. The improved Vector Field Histogram (VFH+) algorithm consistently demonstrated its ability to get the robot to its goal position, as seen in figures 6 and 7, within the same simulated environment applied to the Potential Field algorithm.

Figure 5. *Stuck mobile robot executing Potential Field algorithm*

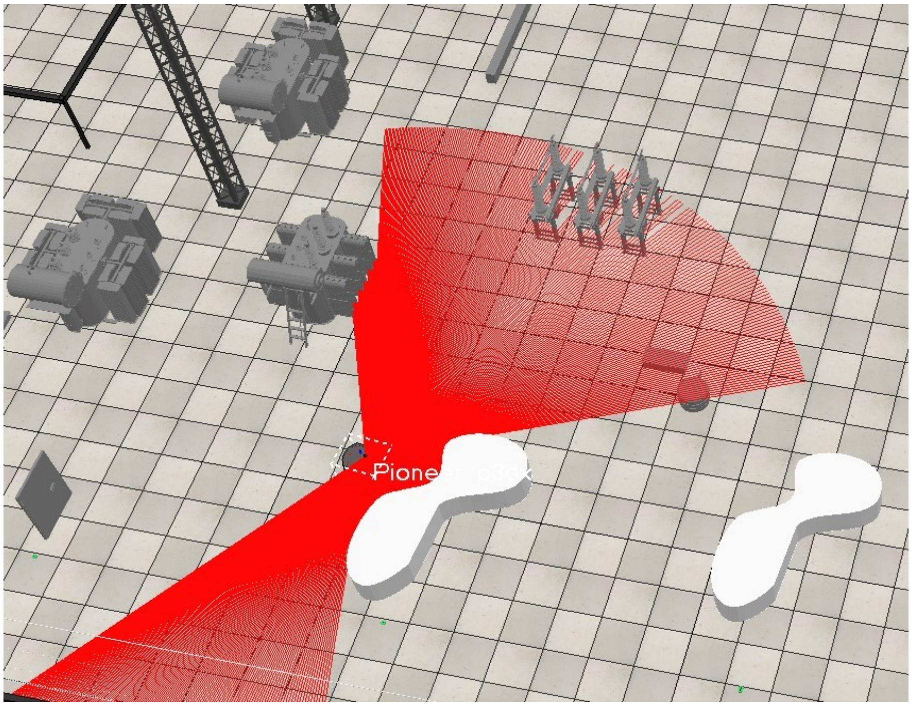


Figure 6. *Mobile robot executing improved Vector Field Histogram (VFH+) algorithm*

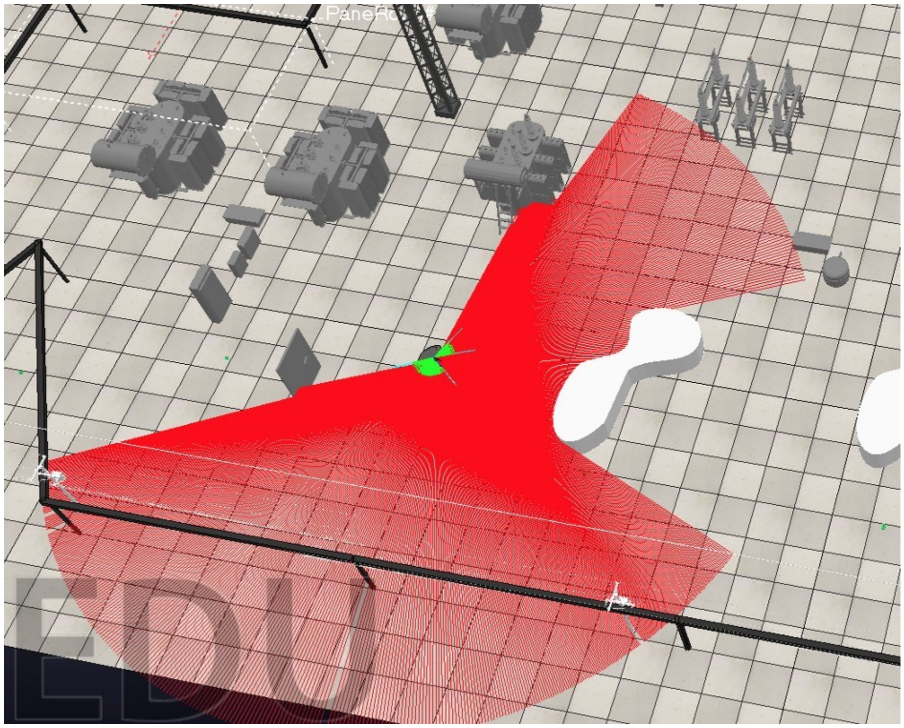
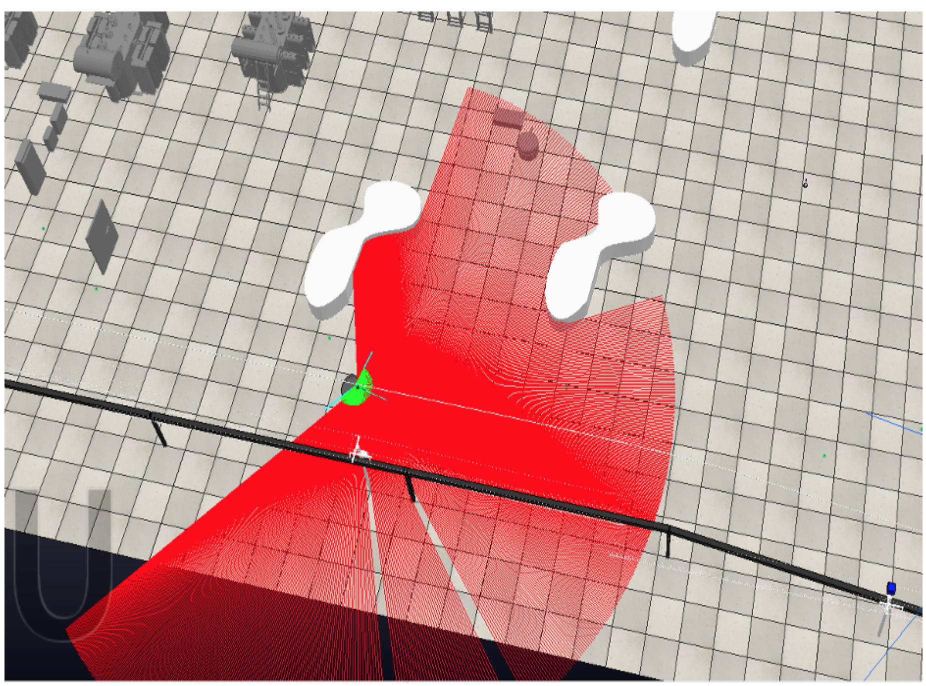


Figure 7. *Mobile robot executing improved Vector Field Histogram (VFH+) navigating away from obstacle which earlier got it stuck while executing Potential Field algorithm*



Overall, the VFH+ algorithm demonstrated a consistently better performance than the Potential Field algorithm, as depicted in table 2 below. After avoiding all impending obstacles in its path towards the goal position, the robot arm situated on the mobile robot was activated to locate the object simulating the fault position on the power line, pick it up and reposition the object of the power line.

Table 3. *Algorithm performance summary*

Algorithm performance under different obstacle complexity scenarios				
Algorithm	Scene Start Coordinate (x, y)	Duration	Journey Completion Status	Remark
PF	-1.8429, +8.2570	Nil	Not Completed	Stuck in front of obstacle
VFH+	-1.8429, +8.2570	1 min 27 sec 94ms	Completed	
PF	+91321, -7.0180	2 min 03 sec 66ms	Completed	
VFH+	+91321, -7.0180	1 min 24sec 20ms	Completed	
PF	+4.9321, +3.6320	1 min 52sec 55ms	Completed	
VFH+	+4.9321, +3.6320	1 min 31 sec 18ms	Completed	
PF	+5.1571, +1.1820	1 min 46 sec 11ms	Completed	
VFH+	+5.1571, +1.1820	1 min 15 sec 59ms	Completed	
PF	-3.8429, +8.0820	Nil	Not Completed	Stuck in front of obstacle
VFH+	-3.8429, +8.0820	2 min 11 sec 21ms	Completed	

Another observation was around ensuring that the form factor of the robot arm was considered in tuning the "**maximum stop distance**" of the mobile robot from its goal position to allow for the effective operation of the robot arm. In addition, coding for real-time processing was also essential for successfully operating the robot arm manipulator.

Discussion

Given that the nature of the project is limited to a proof of concept to simulate the feasibility of our work within a simulation environment that obeyed the physics principles, we were able to achieve coordinated communication between a flying drone and a mobile robot that navigated to a goal position. This was realized with the mobile robot avoiding obstacles in its path to the goal.

The approach taken was such that a flight path was defined to ensure the drone flew over the power lines to ensure it was able to detect potential goal positions. On detecting a possible goal position using its attached vision and proximity sensors, which in the simulation context was depicted with a positioned box on the powerline, it determined the positional coordinate of the object and communicated it to a server accessible to the mobile robot. These coordinates were transformed to the scene's coordinate system within the simulator to ensure the robot could compute and plan its path given the robot's exact location.

The mobile robot navigated the simulator scene with the improved vector field algorithm and consistently avoided obstacles in its path compared to the potential field algorithm. This could be attributed to the algorithm's consideration of the robot's kinematic limitations and its direction selection approach, which is based on a cost function that prevented it from getting trapped.

The mobile robot's mounted robot arm and equipped with a proximity sensor detects a simulated fault position depicted box on a powerline. An earlier approach to leverage a vision sensor wasn't effective in coordinating the processing required for detecting the visual image of the box and the corresponding action needed by the manipulating robot arm. This, however, led to the use of a proximity sensor which required no form of image processing and resulted in effectively coordinating interpretation of the proximity sensor signals and the corresponding robot arm manipulator instructions. Thus, it led to successful detection and picking the box depicting a fault location in the simulation environment.

Conclusion

Agent-based autonomous inspection and identification of powerline hot spots was the primary objective and outcome of this project. The automation of this process resulted in being able to execute maintenance routinely, repetitively, and even under conditions that might not be the most conducive for humans.

The project demonstrated the collaboration between two autonomous agents, a drone in flight and a mobile robot. Specifically, it showcased a drone monitoring a powerline environment with its flight path defined accordingly. Whilst operating within a simulated environment, we observed the feasibility of having an operational drone flying along its flight path over powerlines to determine potential failure points requiring attention. It further demonstrated the drone's coordinated communication with a ground-based robot that navigated to a goal position, having avoided all obstacles along its path.

In a bid to improve this solution, extending the functionalities of the robot arm for actual repair operations would be an area for significant future development. This could be realized with mounted micro visual sensors on the robot arm to aid its remote process by a technician.

Exploring additional motion planning algorithms, for example, enhanced Vector Field Histogram, also known as VFH*, could potentially improve the optimality of motion path planning and obstacle avoidance. This is because it leverages the capacity of the 'A' star algorithm to incorporate optimality in its path determination during motion planning.

Operating power considerations for the drone while in an operating environment could include designing capabilities within the mobile robot to provide a landing base that can double as a solar power supply charging base for the drone when not in flight. These improvements, when realized, will provide an opportunity for combined extended operational periods for the drone and robot while out in the field.

References

Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation, 7(3), 278–288. https://doi.org/10.1109/70.88137

Correll, N. (2016). Introduction to Autonomous Robots: Kinematics, Perception, Localization and Planning (2nd ed.) [E-book]. Magellan Scientific.

Chatterjee, A., Rakshit, A., & Singh, N. N. (2014). Vision Based Autonomous Robot Navigation: Algorithms and Implementations (Studies in Computational Intelligence, 455) (2013th ed.). Springer.

Jiang, B., & Mamishev, A. (2004). Robotic Monitoring of Power Systems. IEEE Transactions on Power Delivery, 19(3), 912–918. https://doi.org/10.1109/tpwrd.2004.829918

Kudriashov, A., Buratowski, T., & Giergiel, M. (2020). SLAM Techniques Application for Mobile Robot in Rough Terrain [E-book]. Springer.

Montambault, S., & Pouliot, N. (2007). Design and Validation of a Mobile Robot for Power Line Inspection and Maintenance. 6th International Conference on Field and Service Robotics - FSR 2007. Published.

Ulrich, I., & Borenstein, J. (1998). VFH+: reliable obstacle avoidance for fast mobile robots. Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146). Published. https://doi.org/10.1109/robot.1998.677362

Ulrich, I. and Borenstein, J., "VFH/sup */: local obstacle avoidance with look-ahead verification," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), 2000, pp. 2505-2511 vol.3, doi: 10.1109/ROBOT.2000.846405.

U.S. Bureau of Labor Statistics. (2020). Injuries and illnesses of line installers and repairers. The Economics Daily. Published.

UST-10LX. (2021). Hokuyo. https://hokuyo-usa.com/products/lidar-obstacle-detection/ust-10lx