C951 – Task 2 – NIP2: Disaster Relief Bot

Author: Jarek Cremeans

Scenario:

Real-time search-and-rescue robots are increasingly used to supplement the efforts of the first responders in areas affected by natural disasters. They are used to spot-check the situational awareness of people in distress, survey the extent of flood or tornado damage, evaluate the number of people that had not been evacuated from their neighborhoods, clean debris, and create passable routes.

For this task, you will use the Coppelia Robotics virtual robot and its environment to demonstrate how such robots may be used in disaster recovery. Your first step is to familiarize yourself with this technology by reviewing the information in the “Coppelia Robotics Resources Page” and “CoppeliaSim User Manual” provided in the Web Links section.

For the next step, you will thoroughly describe a disaster situation similar to the ones mentioned above. Next, you will create a virtual prototype of an autonomous robotic recovery system that demonstrates goal-seeking behaviors in navigating through a predefined area. The robotic recovery system will solve a disaster recovery problem of your choice by using the Coppelia Robotics BubbleRob and its environment as the starting point of your prototyping. You will also add sensors to the robot. These sensors will collect vital information to aid in the disaster recovery effort for the scenario you described.

Requirements:

Using the CoppeliaSim virtual robot, create a virtual prototype of an autonomous robotic recovery system that demonstrates goal-seeking behaviors in navigating through a predefined area by doing the following:

A.  Describe the disaster recovery environment you chose and the **two** obstacles you have added to the environment.

B.  Explain how the robot will improve disaster recovery in the environment from part A after you have added the **two** obstacles from part A.

C.  Justify the modifications you made to CoppeliaSim’s robot architecture, including **two** sensors you chose to add, and explain how these sensors will aid the disaster recovery effort.

D.  Describe how the robot maintains an internal representation of the environment.

E.  Explain how the robot implements the following **four** concepts to achieve its goal:

•   reasoning

•   knowledge representation

•   uncertainty

•   intelligence

F.  Explain how the prototype could be further improved, including how reinforced learning and advanced search algorithms can improve the prototype’s performance and learning.

G.  Submit the robot code that you created.

H.  Provide a Panopto video recording that describes the robot and demonstrates its functionalities to stakeholders who are nonpractitioners and include each of the following:

•   a statement of the disaster recovery problem

•   a summary of the environment and the obstacles

•   a summary of the robot’s goal and objectives

•   a description of the robot and its architecture

•   a demonstration of how the robot meets its disaster recovery goals

•   an assessment of the robot’s capabilities

•   an explanation of how to improve the prototype

*Note: For instructions on how to access and use Panopto, use the "Panopto How-To Videos" web link provided below. To access Panopto's website, navigate to the web link titled "Panopto Access," and then choose to log in using the “WGU” option. If prompted, log in using your WGU student portal credentials, and then it will forward you to Panopto’s website.*

*To submit your recording, upload it to the Panopto drop box titled "Intro to Artificial Intelligence NIP2 | C951 (student creators) [assignments]." Once the recording has been uploaded and processed in Panopto's system, retrieve the URL of the recording from Panopto and copy and paste it into the Links option. Upload the remaining task requirements using the Attachments option.*

I.  Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

J.  Demonstrate professional communication in the content and presentation of your submission.

## Requirement Answers:

1. The disaster recovery environment is designed to represent a cave that has caved-in, trapping those who were inside during the cave-in. The entrance to the cave has been blocked in and a portion of the cave has further collapsed – blocking the exit, the center of the cave itself, and a smaller portion within the cave. The disaster relief bot has been tasked to scour the cave, identify any unsafe areas of the cave, and find possible survivors of the cave-in. The simulation depicts this scenario as the bot searches for people, identifies the inner collapses, and navigates the environment – including the two obstacles: both inner collapsed areas- while reporting found people.
2. As referenced in Section A, the simulation depicts the disaster relief robot exploring the environment while searching for people within the cave environment. Dangerous navigable areas are represented by red cylinders – the caved entrance and two inner collapses – and the people within the cave represented by blue cubes. As the robot moves through the disaster environment, it tracks where it has been and reports in the system log “A person has been detected!” This allows the rescue team to mark where it reported the person was found, use additional software to create a virtual map of the rescue zone, and create a plan of action and path to take once inside. This improves safety for the rescue team and make it so that rescue efforts are executed more quickly, minimizing potential delays and additional casualties.
3. The disaster bot created is based on the bubbleRob tutorial bot. To increase its efficiency, two proximity sensors were added to the front of the bot for better visibility when adjusting to obstacles. These additional sensors are modeled after the original sensor but do not have the additional vision sensor. On all three sensors, the Lua code was modified to reduce the turning penalty on each of the sensors for more granular adjustments. Originally, the bot would get stuck doing wide turns due to small portions of obstacles it came across triggering said wide turns. Changing the code allowed for the desired outcome. Additionally, a Lua modification was explicitly created for a fourth sensor used to detect “people” – represented by a cube. This allows the fourth sensor to report when it sees a person. This sensor also has a larger range than the others, allowing for recognition of a person before the bot turns again.
4. The robot has 4 total sensors – 3 proximity detection sensors and a singular sensor for detecting people. When the robot is about to hit an obstacle, the proximity sensors tell the robot to back up and go a different direction. Similarly, when the person detecting sensor detects a person (cube), the person detecting sensor relays a message in the chat log that a person has been found. The messages indicates that the robot is within 0.2 meters of a person – including touching the person. The robot then continues onward and keeps searching until the proximity sensor tells it that a person is not detectable – halting the message in the chat.
5. **Reasoning:** If the robot detects an obstacle with its proximity sensor, then the robot begins backing up to avoid making contact and hitting the obstacle. As it backs up, the robot’s direction pivots away from the obstacle slowly. If the bot detects a person (cube), it sends a message into the chat window notifying that it sees a person. When the person sensor no longer detects a person within range, it stops sending that message.  
   **Knowledge Representation:** As described in Section C, the robot has 4 total sensors – 3 used for navigation and 1 for detecting people (cubes). All these sensors allow for the bot to collect information about the environment it is navigating. The proximity sensors detect all objects within a specific proximity range of the bot, and the person detection sensor specifically detects people (cubes) within a specific proximity of the bot.  
   **Uncertainty:** Due to the adaptability of the robot’s sensors, it can navigate through an unknown environment without any issue. The robot’s proximity sensors will keep the robot safe while exploring any environment it is placed into. The person sensor will identify people as it comes across them whilst navigating the unknown environment.  
   **Intelligence:** The robots adaptability ensures that it can achieve its goal – assisting a rescue team – even in an unknown and dangerous environment with minimal setbacks. The proximity sensors keep the bot safe while the person sensor signals rescue teams the locations of survivors.
6. **Reinforced Learning:** Reinforced learning could be used to improve this robot significantly. Providing rewards and penalties for specific outcomes and actions would aid the robot in optimizing its routing - covering more ground in less time – while finding people more efficiently. For example, rewarding the robot for deviating from a path it has already taken while penalizing taking the exact same path as previously would cover more ground in a more efficient manner than it currently does.  
   **Advanced Search Algorithms:** Advanced search algorithms could be used to improve this robot significantly. Like reinforced learning, advanced search algorithms could ensure that all ground is covered and that all people are found but with drastically higher efficiency. Potentially utilizing optimized algorithms based on previous searches could allow for more streamlined approaches. Adding more sensors with wider ranges of detection and routing would also decrease search times for the bot when searching for people.
7. **The code is attached in this zip file.**
8. **The Panopto recording can be found here:** https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=1558b486-f608-4740-a315-b2900024069a#
9. **Sources:**
   1. Coppelia, & Xyz999. (2022, February 3). *‘Entity to Detect” not in version 4.3.0 [Online forum post]*. Coppelia Robotics. Retrieved February 23, 2025, from <https://forum.coppeliarobotics.com/viewtopic.php?t=9594>
   2. <https://manual.coppeliarobotics.com/en/apiFunctions.htm>
      1. The Coppelia API was accessed for explanations and examples of LUA and Python scripting that were leveraged in the Person detection sensor.