Zain Abbas

06/07/25

Assignment # 1: Robotic Vacuum Cleaner

Abstract

The modern technology has revolutionized our lives in many ways from automation to robots. New vacuum cleaners are designed to act as reflex agents, meaning they respond to the environment in real time. In this project, we created a Behavior Tree (BT) that enables the vacuum cleaner to check battery and other conditions before performing cleaning tasks. We created two BT's: a base BT and an experimental BT. The base BT is based on a given set of conditions provided in the assignment, while the experimental BT incorporates creative enhancements.

Introduction:

In this project we are designing a new type of robot vacuum cleaner that operates on simple reflex rules. The robot is fully autonomous meaning it can check the battery before any cleaning tasks and if needed it will dock itself to charging stations. This autonomous mode simplifies the cleaning process and reduces human involvement. With minimal human interaction, this new vacuum cleaner performs various cleaning tasks (i.e. spot cleaning, general cleaning, dusty spot sensor). Along with cleaning, the new vacuum cleaner is equipped with AI technology that performs animal face recognition, identifies the pet location and cleans it as necessary.

Methodology:

To accurately perform cleaning tasks, the new vacuum cleaner robot automatically detects dirty spots and cleans them. The vacuum cleaner algorithm is based on reflex agents or (reactive planning) that perform tasks based on current environment. The vacuum cleaner has two main features: base and experimental. Figure 1 below illustrates the base feature behavior tree.

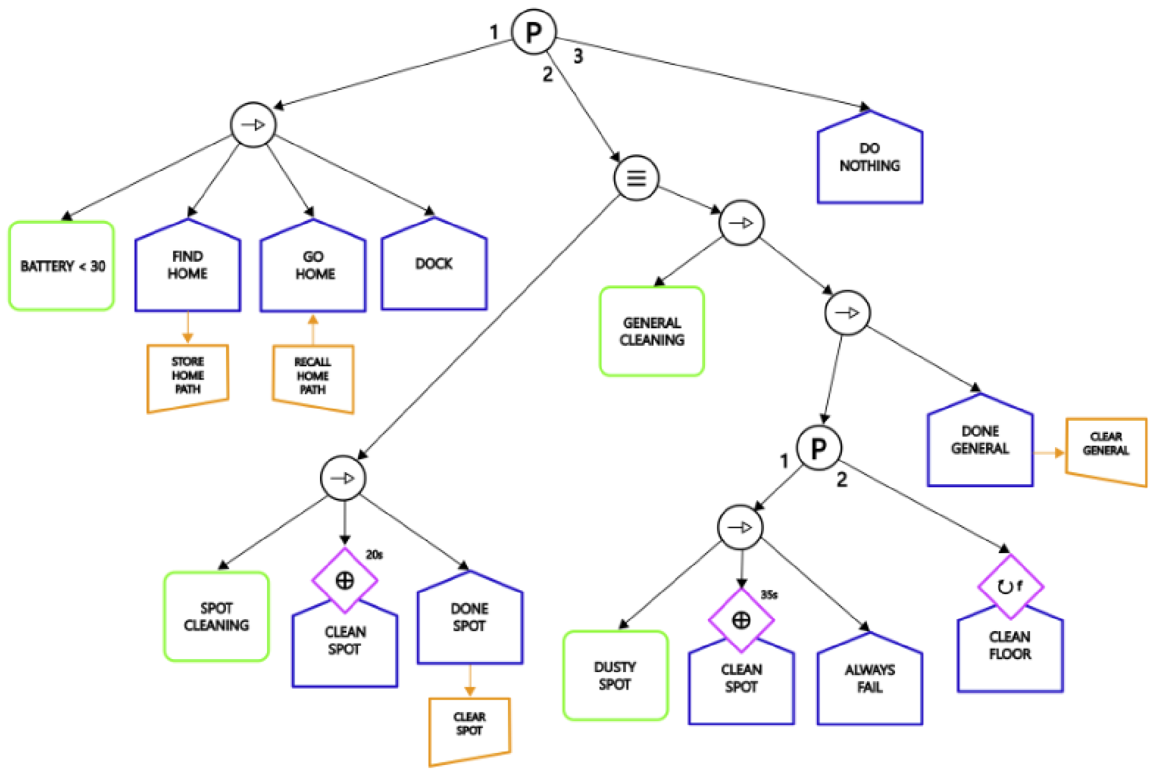

Figure 1: The above BT shows vacuum cleaner base implementation. The initialization of the BT is based on priority node

Initially we have a priority node that will always check the battery level first. If the level is below 30%, it will plan a path to its charging base ("home"), go there, and start the docking procedure. If the battery is sufficient, it will start the function it was commanded to perform. There are two commands available:

1. Spot cleaning: it will perform a 20s intensive cleaning in a specific area.
2. General cleaning: vacuum dust around the room until the battery falls under 30% or completes the task. If the dust sensor detects a particularly dirty spot, the robot will perform a 35s spot cleaning.

The blackboard initializes and updates the nodes and tasks. Few the primary conditions and nodes are:

1. Battery level
2. Spot cleaning
3. General cleaning
4. Dusty spot sensor
5. Clean floor for a specific threshold
6. Pet Recognition
7. Identify pet location

After initialization of the tasks, a while loop continues to run until the resultant condition is set to True. The loop will terminate if the battery falls below 30% while performing any tasks or the robot successfully completes the task. Since the robot uses sensors to navigate and identify dusty spots, it aligns with the definition of reflex agents. By utilizing these sensors, the robot learns about its environment and employs actuators (suction cleaning, motor etc.) to clean the spot. This is a property of a reactive planning robot.

After checking the battery, the second priority is to perform spot cleaning for 20 seconds or general cleaning. The general cleaning mode is equipped with a dusty spot sensor that cleans the spot for 35 seconds if it detects dirt. If no dusty spots are found, the cleaning floor algorithm continues to clean the floor until it finishes the work, or battery is less than 30%. An additional feature of the vacuum cleaner is its ability to capture pets at home using a camera and an AI facial recognition package. This new feature is discussed in more detail in the experimental results section.

Results

This section summarizes the behavior of the vacuum cleaner algorithm.

Demo 1: Spot Cleaning and Floor Cleaning

In demo 1 we set spot cleaning and general cleaning to True. The robot checks the battery and performs spot cleaning for 20 seconds. After that it checks the battery again and cleans the floor until it finishes the task. Figure 2 summarizes the function of this behavior. Whereas figure 3 summarizes the results of cleaning floor behavior

```
 *** Choose the version to run ***
1. Base Version
2. Experimental Version
Enter 1 or 2: 1
Running Base Version ...
Using base version
BatteryLessThan30(0): Checking battery < 30
BatteryLessThan30(0): FAILED
Sequence(5): FAILED
SpotCleaning(6): Checking Spot Cleaning
SpotCleaning(6): SUCCEEDED
Timer(8): time-to-expiration = 19
CleanSpot(7): Clean Spot
CleanSpot(7): SUCCEEDED
Timer(8): RUNNING
Sequence(11): RUNNING
Selection(29): RUNNING
```

```
UntilFails(21): Running until clean floor fails...
CleanFloor(20): Cleaning Floor
CleanFloor(20): SUCCEEDED
Selection(23): RUNNING
Sequence(25): RUNNING
Sequence(26): RUNNING
Selection(29): RUNNING
UntilFails(21): Running until clean floor fails...
CleanFloor(20): Cleaning Floor
CleanFloor(20): FAILED
*** Finished Cleaning Floor ***
Selection(23): SUCCEEDED
DoneGeneral(24): GENERAL CLEANING DONE
DoneGeneral(24): SUCCEEDED
Sequence(25): SUCCEEDED
Sequence(26): SUCCEEDED
Selection(29): SUCCEEDED
All tasks completed. Shutting down.
```

Figure 2: The above diagram shows the results of spot cleaning and general cleaning. (a) on the left side shows the results of spot cleaning and (b) on the right side shows the results of floor cleaning.

DEMO 2: Dusty Spot and Floor Cleaning

Demo 2 summarizes the behavior of dusty spot cleaning. The algorithm checks battery and spot cleaning. After both fails, it continues to dusty spot cleaning, where the sensor identifies the dusty spot and cleans it under it is fully done. After finishing the dusty spot cleaning, the tree continues to second priority which is cleaning the floor.

```
 *** Choose the version to run ***
1. Base Version
2. Experimental Version
Enter 1 or 2: 1
Running Base Version ...
Using base version
BatteryLessThan30(0): Checking battery < 30
BatteryLessThan30(0): FAILED
Sequence(5): FAILED
SpotCleaning(6): Checking Spot Cleaning
SpotCleaning(6): FAILED
Sequence(11): FAILED
GeneralCleaning(12): Checking General Cleaning
Current State: True
GeneralCleaning(12): SUCCEEDED
DustySpot(14): Checking Dusty Spot
DustySpot(14): SUCCEEDED
Timer(16): time-to-expiration = 34
CleanSpot(15): Clean Spot
CleanSpot(15): SUCCEEDED
Timer(16): RUNNING
```

```
CleanSpot(15): Clean Spot
CleanSpot(15): SUCCEEDED
Timer(16): RUNNING
Sequence(19): RUNNING
Selection(23): RUNNING
Sequence(25): RUNNING
Sequence(26): RUNNING
Selection(29): RUNNING
Timer(16): SUCCEEDED
AlwaysFail(18): Always Fail
AlwaysFail(18): FAILED
Sequence(19): FAILED
UntilFails(21): Running until clean floor fails...
CleanFloor(20): Cleaning Floor
CleanFloor(20): SUCCEEDED
```

Figure 3: This shows that once the dusty spot is cleaned, that vacuum will continue to clean the floor until the conditions meet.

DEMO3: Battery Interruption

Demo 3 represents battery interruption case. While performing the tasks, if the battery falls less than 30 percent, the behavior tree interrupts the algorithm, find the docking station and dock itself to charge the battery. This feature enhances vacuum cleaner capability to dock and charge the battery if needed.

```
Timer(8): time-to-expiration = 10
CleanSpot(7): Clean Spot
CleanSpot(7): SUCCEEDED
Timer(8): RUNNING
Sequence(11): RUNNING
Selection(29): RUNNING
LowBatteryInterrupt(9):  *** INTERRUPTED: Battery too low ***
LowBatteryInterrupt(9): FAILED
Sequence(11): FAILED
LowBatteryInterrupt(13):  *** INTERRUPTED: Battery too low ***
LowBatteryInterrupt(13): FAILED
Sequence(26): FAILED
DoNothing(27): Do Nothing
DoNothing(27): SUCCEEDED
Sequence(28): SUCCEEDED
Selection(29): SUCCEEDED
 ***  Battery is low ***  28
BatteryLessThan30(0): Checking battery < 30
BatteryLessThan30(0): SUCCEEDED
FindHome(1): Looking for a home
FindHome(1): SUCCEEDED
GoHome(2): Go Home
GoHome(2): SUCCEEDED
Docking(3): Docking
Docking(3): SUCCEEDED
ChargingBattery(4): Charging the battery
ChargingBattery(4): SUCCEEDED
Sequence(5): SUCCEEDED
Selection(29): SUCCEEDED
All tasks completed. Shutting down.
```

Figure 4: It shows the vacuum cleaner battery interruption while cleaning the floor.

Extra Experiment

In the extra experiment, we included an AI (Artificial Intelligence) Pet Recognition module that captures the image of the pet at home and responds accordingly. The vacuum now has a built-in camera embedded with an AI algorithm that recognizes pet movement and locks the grid space of the pet. It works as follows:

1. Check if the pet (e.g. cat or dog) is at home
2. Identify the location and check if that location is empty
3. If the pet is not present at the current location the vacuum cleaner cleans the spot for 10 seconds
4. If a pet is present at its current location, the vacuum continues with other tasks.

This new built-in AI facial recognition feature enhances the capabilities of vacuum cleaner and bypass human involvement of cleaning cat or dog hairs. Figure 2: shows the behavior tree of this new feature.

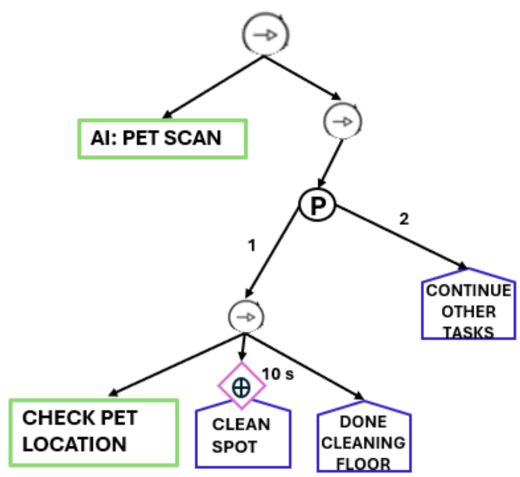

Figure 5: This new algorithm has AI facial recognition feature that focuses on animal movement. It carefully captures and identifies pet location and cleans it as necessary.

AI Pet Recognition:

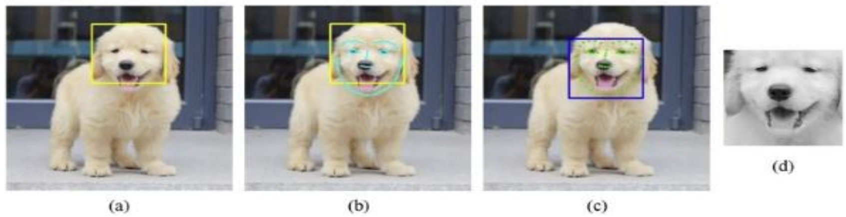

Figure 6: The camera captures the facial image. A. Recognizes general region of dog's face, Draw facial contour lines. Capture facial region and convert into grayscale for preprocessing [1].

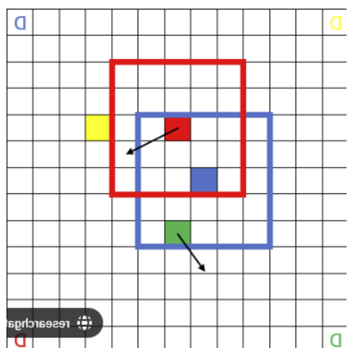

Figure 7: Check the location of the pet in a grid space and save the initial location. Check if that location is empty [2].

The new capability is included with the base implementation, enhancing the existing features of the vacuum cleaner. Figure 8 summarizes the results of this new technology.

```
 *** Choose the version to run ***
1. Base Version
2. Experimental Version
Enter 1 or 2: 2
Running Experimental Version ...
BatteryLessThan30(0): Checking battery < 30
BatteryLessThan30(0): FAILED
Sequence(5): FAILED
PetScanRecognition(6): Check if Cat or Dog is in house
PetScanRecognition(6): SUCCEEDED
CheckPetLocation(8):  *** Checking if the Pet location is empty and saving the grid point ***
CheckPetLocation(8): SUCCEEDED
Timer(10): time-to-expiration = 9
CleanSpot(9): Clean Spot
CleanSpot(9): SUCCEEDED
Timer(10): RUNNING
Sequence(13): RUNNING
Selection(16): RUNNING
Sequence(17): RUNNING
Sequence(18): RUNNING
Selection(42): RUNNING
Timer(10): time-to-expiration = 8
CleanSpot(9): Clean Spot
CleanSpot(9): SUCCEEDED
```

Figure 8: The above figure shows the implementation of AI face recognition technology to detect and track the pet at home. After that it performs cleaning tasks to that location.

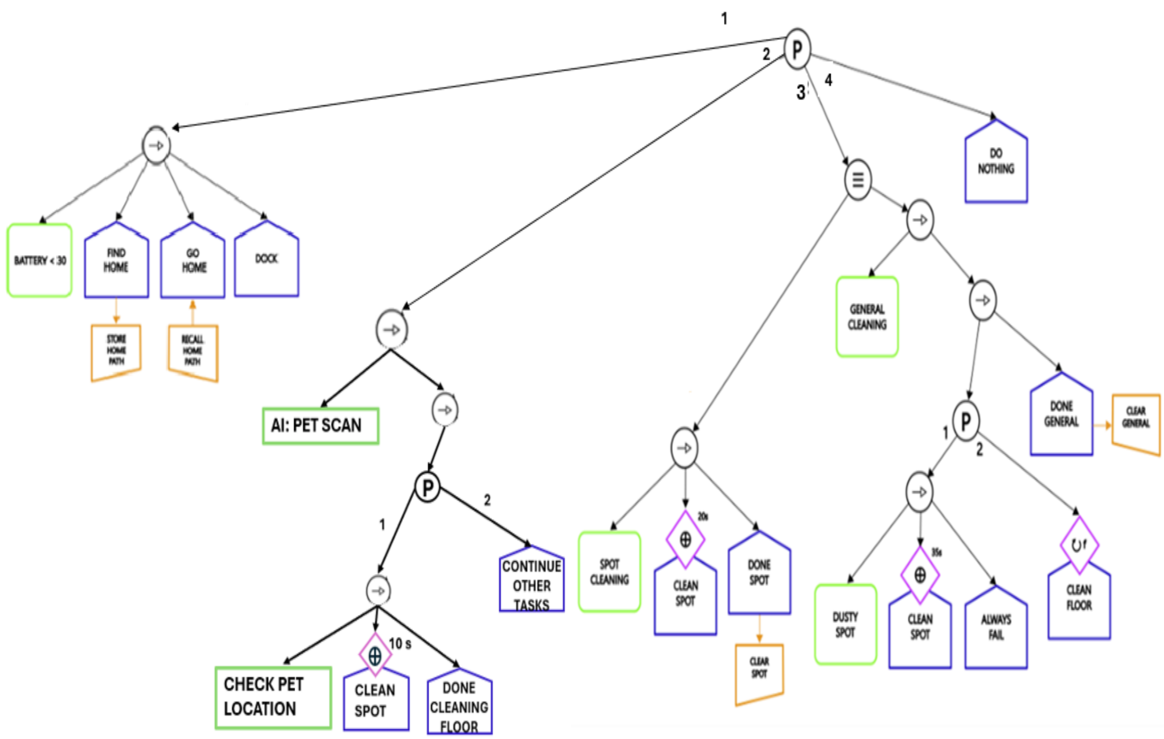

Figure 3: The tree shows existing tree along with new capability.

Conclusion

In this project we designed and developed an autonomous robot vacuum cleaner that incorporates reflex agent behavior for efficient task execution. The vacuum cleaner was programmed using a BT, which enables it to check its battery and perform cleaning tasks accordingly. Two distinct BTs were implemented: a base BT and experimental BT (including innovative AI facial recognition enhancements). The system autonomously performs cleaning tasks like spot cleaning and dusty spot cleaning with a primary focus on ensuring that the battery level remains above 30% during operation. Additionally, we integrated a camera and AI facial recognition technology to identify pets within the home, allowing the vacuum to adjust its cleaning tasks based on the pet's location.

From running the BT, we concluded that vacuum cleaner performed well in a controlled environment, with successful battery checks and responsive cleaning tasks. Overall, the project demonstrates significant potential for autonomous cleaning in homes with added benefit of pet recognition. However, there remains room for improvement in terms of sensing accuracy, battery management, and the overall adaptability of the system in more complex environments. Future work will focus on addressing these challenges and exploring additional functionalities to improve performance and user experience.

References

[1] *Connected-Vet. 20 March 2023*

https://connected-vet.com/pet-dog-facial-expression-recognition-based-on-convolutional-neural-network-and-improved-whale-optimization-algorithm

[2] *Wang, Weichang.* ***3M-RL: Multi-Resolution, Multi-Agent, Mean-Field Reinforcement Learning for Autonomous UAV Routing****. Published in June 2021.*