CG1112 Engineering Principle and Practice

Semester 2 2017/2018

"Vincent to the Rescue"

Design Report

Team: 02-04-02

Name	Student #	Sub-Team	Role
Liu Tianhang A0167038B	1	Hardware	Sensors, LIDAR mapping and I2C
Wang Jiannan A0172069B	2	Software	Navigation
Hu Yidi A0173378U	3	Hardware	Motor control
Lim Jit Wei A0170832J	4	Software	Communication
Li Jiawei A0177400M	5	Software	Surrounding feedback

Section 1 System Functionalities

The 2008 China Wenchuan earthquake was a total disaster. Near 70000 lost their lives, 3750000 were injured and the total affected population exceeded 45 million. After the earthquake, the Chinese government has immediately mobilize all resources available for rescue efforts, including over 100 thousands soldiers, etc. However, given the complex and dangerous environments in disaster-affected areas, many rescue missions have been constrained by environments inaccessible to human rescuers (e.g. building debris) and many lives that could have been saved were regretfully lost. With the advancement of technologies and the increasing awareness of the application of technology in disaster rescue efforts, various technological solutions have been proposed catering to different disaster scenarios. In this report, the design of a rescue robot "Vincent" has been elaborated. When designing "Vincent", the team has put into considerations different constraints and needs that a practical rescue robot has to tackle in disasters like an earthquake. Therefore, "Vincent" is designed to fulfill the following system functionalities:

- Accurately self-navigate through disaster-stricken areas
- Turn by desired angles
- Monitor and report specific environmental variables, such as temperature, humidity, atmospheric components, etc
- Avoid water puddles that might damage the robot's electronic components
- Mark current locations
- Stop at marked locations
- Backtrack to starting point through the same route
- Establish video live-stream of surroundings
- Identify victims and establish tele-communications between victims and rescuers

Section 2 Review of State of the Art

Tele-operating search and rescue robotic platform

a. The Emergency Integrated Lifesaving Lanyard (EMILY) from US company Hydronalix is a self-propelled remote-controlled robotic rescue boat. It is equipped with ropes at four of its sides and is buoyant enough to keep up to five people alive [2]. EMILY is powered by a jet pump with inlet grate instead of a propeller or rudder so it will not harm the victim or catch on rocks or sand [5]. EMILY is powered by a battery that can last 10km on a single battery charge [5]. EMILY can "provide flotation until a rescuer arrives, deliver life jackets, or pull a recovery rescue line up to 800 yards through strong currents and large surf [5]." It has a built in Doppler sonar to help avoiding collisions with unsuspecting swimmers [2]. It also has sensors that detect underwater movements to look out for swimmers in distress [2]. The microphone and loudspeaker on EMILY enables the human operator to distinguish children splashing around from swimmers struggling for their lives [2].

b. Strength: Can travel 12 times as fast as a lifeguard (up to 48 km/hr), able to track straight during wave breaching, highly durable, easy to deploy, no propeller or rudder that will harm victims accidentally [2].

Weakness: EMILY is unable to save unconscious swimmers because they cannot grab hold of the device's ropes [2]. It requires the guidance and control of human to carry out rescue operation and does not support multi-language instructions (instructions are made from the lifeguards, thus limited by the languages they speak) [5].

"EMILY Is A Robotic Lifeguard That Jumps From The Helicopter To Save Refugees In Peril On The Sea [1]."

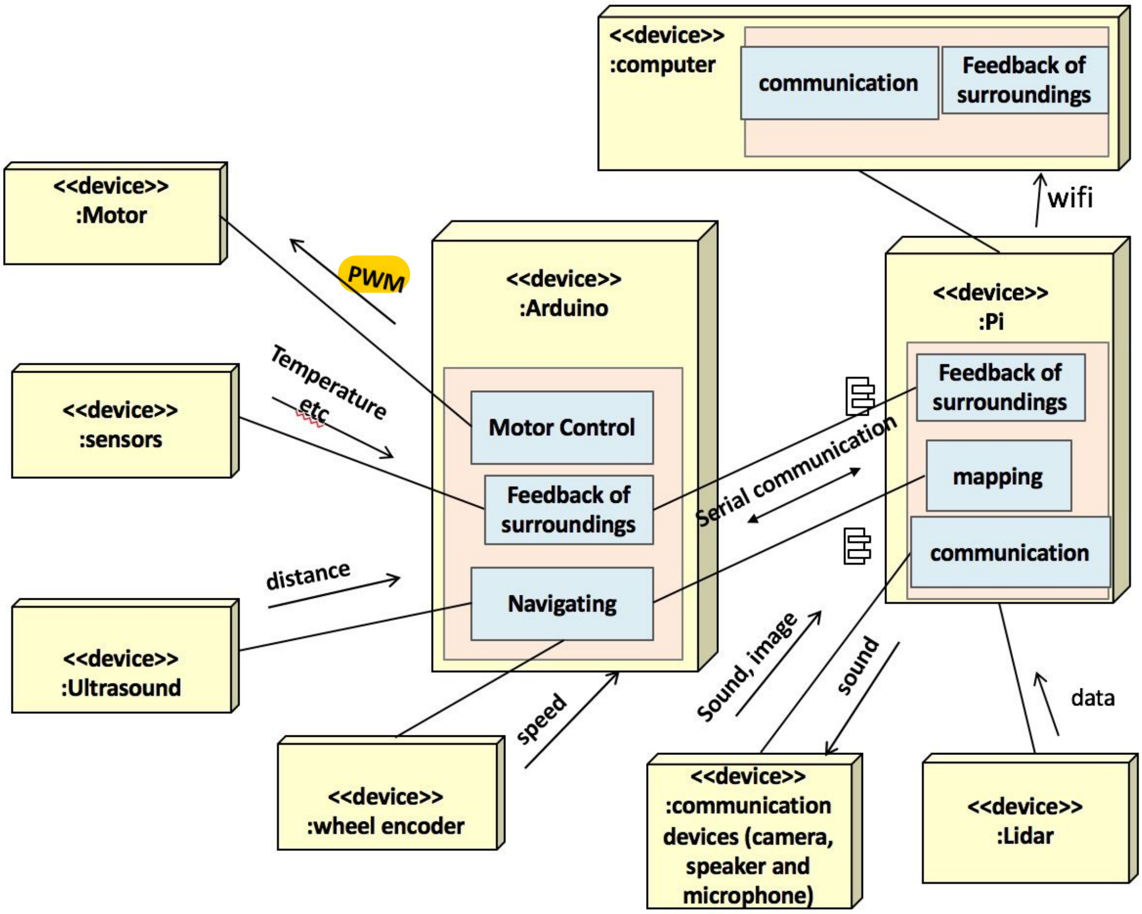
Autonomous search and rescue robotic platform

a. A group comprised of researchers from the Dalle Molle Institute for Artificial Intelligence, the University of Zurich, and NCCR Robotics, used deep learning neural networks to train an autonomous drone to navigate a forest [3]. The system consists of three GoPro cameras to capture images that are later used to train their deep-learning algorithm. For example, photos taken on the left camera will be used to train the drone to go right whenever it sees any images with high resemblances. (Take note this is only needed to obtain the training set) "The result was a deep-learning algorithm that allows a drone equipped with a single forward-facing color camera to navigate a previously-unseen trail completely on its own — no human interaction whatsoever [3]."

b. Strength: The drone can autonomously navigate a forest after being trained with deep learning neural networks using image set of the forest trail. The algorithm hits a higher accuracy of 85 percent for recognizing the correct direction to go as opposed to 82 percent for a human to determine the correct direction given an image of the trail [3].

Weakness: The system is only tested using image set from the same trail as the one the drone will be navigating. The accuracy would highly decrease when it is used to navigate an entirely new trail using image set from a different trail. Therefore this system is still not reliable enough to navigate an unexplored trail.

Section 3 System Architecture



Section 4 Component Design

1. Summarized steps of function

Phase 1:(tele-operated)
1. Initialization
2. Wait for start command
3. SLAM will starting working.
4. Operator will send the movement message through WIFI based on map by SLAM.
5. Mark the location as necessary
6. Sensors will continuously send feedbacks to arduino and PI for procession.
7. Data from the digital camera unit and telecommunication unit are continuously being sent back to PI
8. Repeat step 3 until Vincent has explored and mapped the entire arena

Phase 1:(self-navigating)
1. Initialization
2. Wait for start command
3. Start navigation and mapping (SLAM algorithm)
4. Mark the location as necessary(when detecting "object")
5. Sensors will continuously send feedbacks to arduino and PI
6. Data from the digital camera unit and telecommunication unit are continuously being sent back to PI
6. Repeat step 3 until Vincent has explored and mapped the entire arena

Phase 2: (continued)
6. Wait for "Start Autonomous Backtracking" command
7. Navigate backwards using the same path travelled in phase 1
8. Flash LEDs for 2 seconds whenever it reaches one of the marked locations

Further breakdown:

Phase 1 (self-navigation)

Step 1. Initialization

A. Pi pings the Arduino by sending an arbitrary chosen number and the Arduino will blink the LED the same number of times. During the process, serial transmission protocol is set up.

B. Pi will send back initial mapping of the current surrounding from the Lidar sensor and ask arduino for feedback about the distance from both sides of the wall with Ultrasonic sensor/ Infrared sensor. (Calibration)

C. Setup the magnetometer. This will give arduino and Human a sensor of direction relatively to the Earth magnetic field.

Step 2. Wait for start command

A serial message send out from pi side to arduino like “start”, the arduino will start to run all the sensors and hardware.

Step 3. Start navigation and mapping

Navigation will be done with the aid of LIDAR, ultrasound sensors and infrared sensors. Vincent will detect the optimal pathway based on information gathered from LIDAR and the sensors and commands will be given to the speed-control unit to control left and right wheel to rotate for respective rounds to achieve motion of the robot. At the same time, commands will be stored in a stack ADT in reverse way (turn left is turn right) (this is to help the robot backtrack). The mapping will be done by the LIDAR with the help of SLAM algorithm.

Step 4. Mark the location as necessary (when detecting “object”)

When we mark it, we will store a MARKED command into the stack.

Phase 2 (Backtracking)

Step 1. Navigate backwards using the same path travelled in phase 1

Since we stored the movement command in the stack, the only thing we need to do is to pop out the command. (stack is abstract data type follows the last in first out idea)

Step 2. Flash LEDs for 2 seconds whenever it reaches one of the marked locations

During the process of popping of history commands, when we see a MARKED command, an interrupt will be triggered where arduino will exit from popping history commands and run a flashLED algorithm to flash LED to indicate that it has reached a marked location.

2. Further breakdown into hardware:

Since Pi is a micro-pc, it will control the arduino to complete the task. On the other hand, arduino is like the hand of the human. It will be connected with all the sensors to complete the task.

In this project, we have add in a pressure sensor (which also serves as temperature sensor), a carbon monoxide (CO) sensor, an oxygen level sensor and a rain drop sensor. Also, ultrasound sensors and infrared sensors (tentative) are added in for better navigation.

For the pressure sensor, it will feedback the pressure and temperature. This will give the rescue team a rough estimation of the environmental conditions of the search area.

The magnetometer will a relative direction against the Earth's magnetic field

The carbon monoxide sensor will give feedback of the concentration of carbon monoxide while the oxygen sensor will report the concentration of oxygen in the environment back to operator. This would help the rescue team to be prepared in advance before going into the site.

The raindrop sensors will be put on both wings of the rescue robot. So, when the water level is beyond a particular preset point that would pose harm to the electronics of our rescue robot, it will trigger the sensor to send a warning message to the arduino to trigger an interrupt and stop the vehicle.

As pressure sensors and magnetometer are digital sensors, we will use Inter-Integrated Circuit(I2C) interface to connect to arduino. While the carbon monoxide sensor and rain drop sensor,rain drop sensor and oxygen level sensor are analog, we just take in the analog voltage and convert it into data using the information of individual datasheet.

The microphone and camera will be extremely useful as it enables the rescue team to directly monitor the situation and even communicate with any survivors in the site to facilitate further rescue efforts.

Section 5 Project Plan

Week No.	Ideas & Performance	Report
7	Analyse the project specifications and understand the basic requirements as well as the bonus features of the robot Vincent. Brainstorm the ideas about extra features we could implement. Try out LIDAR.	Start writing the design report.
8	Finalise the plan of purchase for additional sensors, and the design of the robot Vincent. Start coding the necessary programs for Raspberry Pi and Arduino. Test the motors and wheel encoders with Arduino, whose program compiled by Raspberry Pi (the actual working flow for the project).	Finish the design report and submit. Beautify the Github repository and invite the instructor.
9	Complete the purchase of additional sensors and supplementary accessories. Continue to program for the critical features. Start to configure the robot with components that have completed the tests.	Revise the design report and change the related configurations. Internalise the requirements for the final report during the CELC session.
10	Accomplish the basic configuration of the robot (components connected by PCB / wires). Test the basic functions of the robot, such as going straight and turning by certain degree, under manual commands. Finish programming for the crucial components of the robot.	Prepare for the mock presentation, and give and receive peer review and feedback to or from other teams.
11	Implement all the codes on the actual design, and perform multiple trials about the task.	Start writing the final report.

	Modify any unsuitable part and revise the design. Start adding in extra features decided in the design report.	
12	Test the robot in the mock evaluation. Take the chance to figure out any inaccurate process the robot has gone through in the evaluation. Resolve the faults and improve the stability of the robot.	Finish the draft of the final report and submit. Start preparing the presentation for final evaluation.
13	Perform trial evaluation for multiple times. Modify the configurations and the codes to enhance the ability of the robot. Complete the final evaluation.	Rehearse for presentation. Revise scripts and slides for modifications. Revise and finalize the draft for final report.

Reference

Images from the Internet

[1] Wajeeha, "EMILY Is A Robotic Lifeguard That Jumps From The Helicopter To Save Refugees In Peril On The Sea," Wonderful Engineering, 10-May-2016. [Online]. Available: http://wonderfulengineering.com/emily-is-a-robotic-lifeguard-that-jumps-from-the-helicopter-to-save-refugees-in-peril-on-the-sea/.

Newspaper Article from the Internet

[2] "EMILY to the rescue," *The Economist*, 23-Aug-2010. [Online]. Available: https://www.economist.com/node/16877335. [Accessed: 17-Mar-2018].

[3] L. Silverman, "Meet Emily, The Lifeguard Robot That's Saving Refugees Crossing The Mediterranean Sea," *KERA News*. [Online]. Available: http://keranews.org/post/meet-emily-lifeguard-robot-thats-saving-refugees-crossing-mediterranean-sea. [Accessed: 17-Mar-2018].

[4] A. Brokaw, "Autonomous search-and-rescue drones outperform humans at navigating forest trails," *The Verge*, 11-Feb-2016. [Online]. Available: https://www.theverge.com/2016/2/11/10965414/autonomous-drones-deep-learning-navigation-mapping. [Accessed: 17-Mar-2018].

Official Internet Site

[5] "Key features," *EMILY - Robotic Rescue Boat*. [Online]. Available: http://www.emilyrobot.com.au/key-features/. [Accessed: 17-Mar-2018].