

**CG1112 Engineering Principle and Practice**

Semester 2 2017/2018

**“Vincent to the Rescue”**

**Design Report**

**Team: 03-04-01**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student # | Sub-Team | Role |
| Saif Uddin Mahmud | A0170896N | Software | Head of PI-Human Interface coding |
| Shreyas Kuthanoor Prakash | A0177064B | Software | Head of PI-Arduino Interface coding |
| Pham Quang Minh | A0170723L | Hardware | Head of Physical Design |
| Bhatt Paras Krishna | A0168819L | Hardware | Head of Physical Optimization and Calibration |

**Section 1: System Functionalities**

Vincent is currently being designed as a semi-autonomous, tele-operated robot, which is capable of accurately navigating through its surroundings. It will emulate a simple remotely controlled search-and-rescue robot. To that end, the Vincent system will feature and support the following functionalities:

1. Wireless Control

In order to achieve this, VNC[1] and SSH[2] will be used to interface with Vincent’s Raspberry Pi-Arduino control system.

1. Accurate Motion in 2D

Wheel encoders, motors, and magnetometers will enable Vincent to reliably navigate the obstacles in the maze.

1. Mapping of Environment

LIDAR[3] data will be fed into a SLAM[4] algorithm to procure an approximate topography of the arena.

1. Autonomous Backtracking to Marked Positions

Memory of the previous commands will enable Vincent to determine, without user intervention, the path to previously marked positions. An LED display will alert users of Vincent’s arrival at the aforementioned positions.

1. Optional- Ability to Perform the Above with Complete Autonomy

If possible, Vincent will be able to move around and scan the room autonomously to draw the full map of a room without human intervention.

**Section 2: Review of State of the Art**

Currently, there exist two broad categories of search and rescue robots: **tele-operating** robots and **autonomous** robots. These are differentiated by the level of user involvement, with autonomous robots having very limited controller interaction, and tele-operating robots requiring extensive controller interaction. An example of a tele-operatingsearch and rescue robotic platform is RAPOSA [5]. Conversely, the Robot BigDog[6] is an example of an autonomous rescue robot.

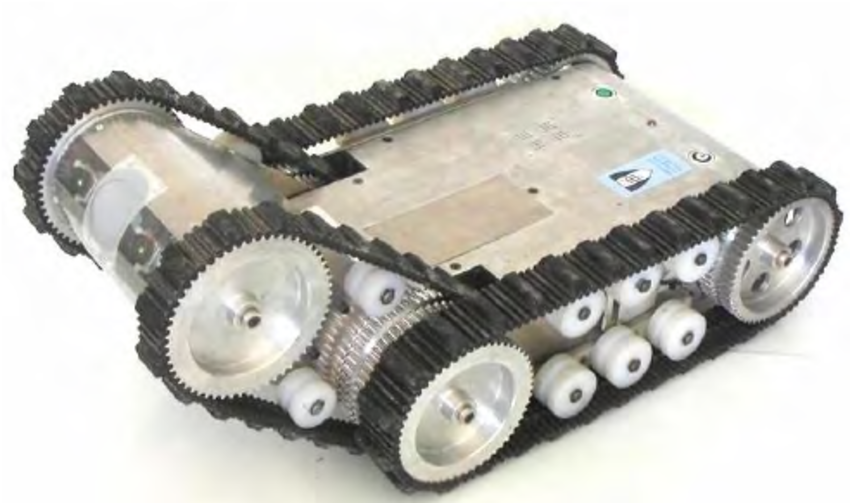


Figure 1: RAPOSA [5] Figure 2: BigDog [6]

RAPOSA can search and rescue survivors in hostile environments, such as rubble, narrow passages, and can even be suspended inside holes. It can be used to rescue victims of natural / manmade disasters like earthquakes, landslides, hurricanes, terrorist attacks, etc. RAPOSA has a Graphical User Interface (GUI) and supports both tethered and wireless communication. It has a main body and a frontal body, two-side tracked wheels and flipping capability. The sensors and other devices connected to it include: two webcams and two thermal cameras for a 360 degree view, a gamepad interface to control the bot, temperature, humidity, gas sensors and analog accelerometers[7].

It can efficiently tackle environments such as dark tunnels, deep holes and can even climb and descend stairs. The webcams also have Infrared and Thermal imaging capabilities. However, it is not fully autonomous, so it is heavily dependent on the operator for its working. Thus, it cannot be easily extended to multi-robot rescue operations.

In contrast, BigDog is a fully autonomous four legged robot. It uses the following components and functionalities: LIDAR, Stereo vision, Ring Laser, Gyro and Linear accelerometers, hydraulic pumps and a gasoline engine. Using perception and navigation algorithms, it is easily able to perform autonomous navigation to goal positions. [8] The on-board computer which controls the navigation and the internal hydraulic systems is “a ruggedized [PC/104](https://en.wikipedia.org/wiki/PC/104) board stack with a [Pentium 4](https://en.wikipedia.org/wiki/Pentium_4) class computer running [QNX](https://en.wikipedia.org/wiki/QNX).” [9] Some of its strengths include exceptional performance in harsh terrain such as water, snow and forests, and self-righting capability. However, a major weakness for BigDog is that it has noisy operation, so it is not capable in various operations, especially in combat-rescue situations.

**Section 3: System Architecture**

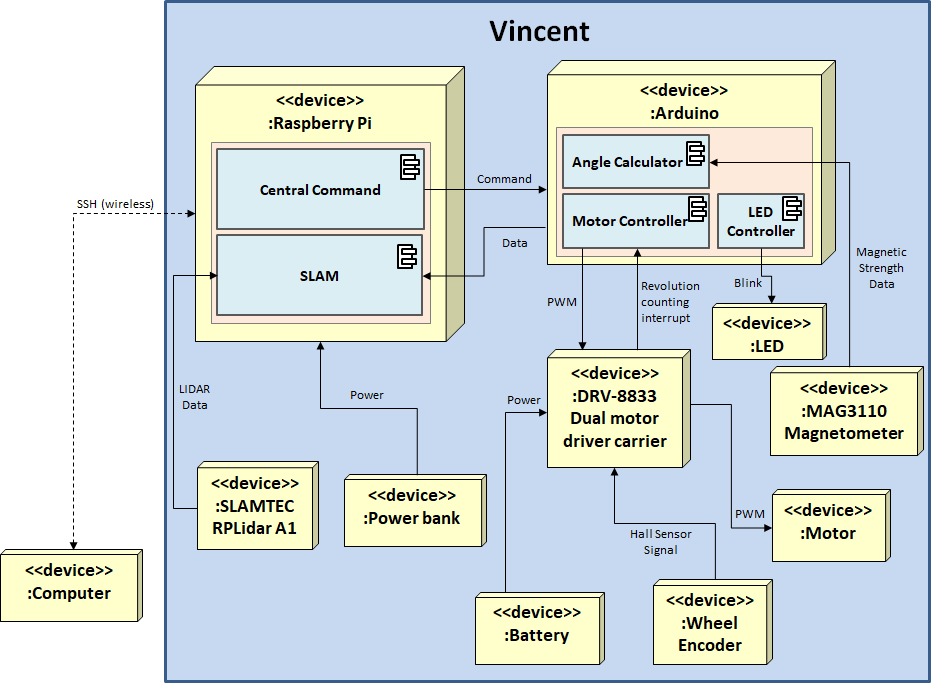


Figure 3: Vincent’s System Architecture

**Section 4: Component Design**

|  |
| --- |
| **Vincent’s Operational Overview**  1. Initialization   2. During Navigation Phase:  a. Scan the environment and send back data to user as required  b. If interrupted by user-issued command, receive and parse command  c. Store a log of commands given   d. Carry out command and report back to user  3. During Backtracking Phase:   a. Retrieve relevant commands and turn them into their “undo” counterparts.   b. Execute transformed commands in LIFO order  c. Scan and send topographical data back to user from SLAM (for debugging)  d. Upon reaching a previously marked position, give a cue to the assessment team |

**Detailed Breakdown**

1. Initialization:  
 a. Establish SSH and VNC connection between Admin and Pi   
 b. Execute Handshaking Protocol between Pi and Arduino   
 c. Spin Vincent once to calibrate the magnetometer  
 d. Build initial map with LIDAR readings   
 2. During Navigation Phase:

a. Scan environment and send back data as required:

- LIDAR will scan and send data to Pi ONLY when stationary.   
 - Pi processes the data using SLAM (RANSAC[10] and Scan-Matching[11])

- Generated map is sent to user   
 b. Human-Pi Interaction  
 - User issues commands to Pi via SSH based on the map  
 - Pi parses the commands  
 - The list of commands are: move(speed, distance), turn(angle), markHere()   
 c. Pi-Arduino Interaction  
 - Pi issues commands to Arduino (eg: Activate Motors, turn on LED)  
 - Pi maintains a log of commands sent to Arduino in the form of a stack   
 d. Execution  
 - Arduino executes command and signals Pi when done  
 - Pi reports back to user

- Go to 2a

[Navigation Phase is terminated by user command]

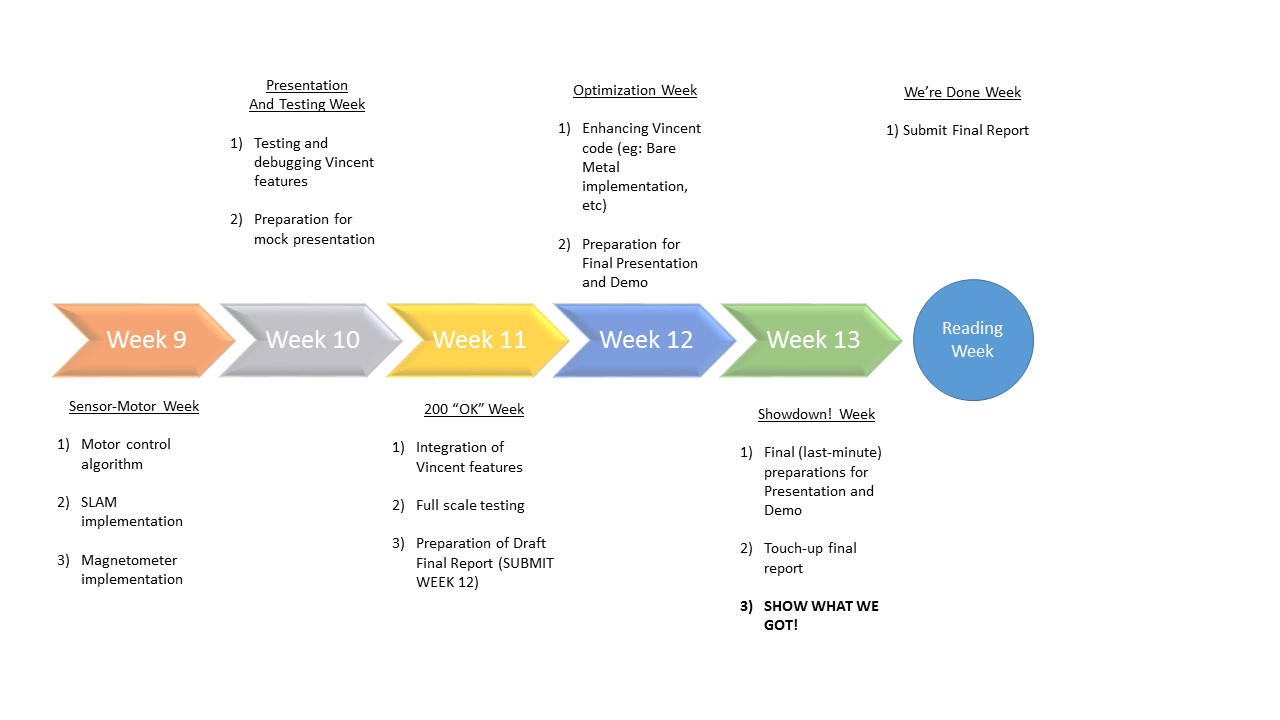
3. During Backtracking Phase  
 a. Retrieve relevant commands and turn them into their “undo” counterparts:  
 - Pop the top of command stack.   
 -- Multiply “distance” parameter to move() with -1  
 -- Multiply “angle” parameter to turn() with -1

-- Convert markHere() command to LEDshow()

- Pi issues new “undone” command to Arduino

- Arduino finishes command and reports back to Pi

- For debugging purposes, Pi reports back to user   
 b. Scan and send topographical data back to user from SLAM (for debugging)   
 [if stack.empty(), done with this phase]

**Section 5: Project Plan**

**References**

1. "Virtual Network Computing", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Virtual\_Network\_Computing. [Accessed: 16- Mar- 2018].
2. "Secure Shell", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Secure\_Shell. [Accessed: 16- Mar- 2018].
3. "What is LiDAR?", YouTube, 2018. [Online]. Available: https://www.youtube.com/watch?v=eBUCGxZq\_xg. [Accessed: 16- Mar- 2018].
4. "Simultaneous localization and mapping", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Simultaneous\_localization\_and\_mapping. [Accessed: 16- Mar- 2018].
5. RAPOSA: Semi-autonomous robot for rescue operations, 2006. DOI: 10.1109/IROS.2006.281836. [Accessed: 15- Mar- 2018].
6. BigDog: 2010 IEEE International Conference on Robotics and Automation. DOI: [10.1109/ROBOT.2010.5509226](http://doi.org/10.1109/ROBOT.2010.5509226). [Accessed: 15- Mar- 2018].
7. “RAPOSA: Semi-autonomous robot for rescue operations”, Researchgate.net, 2006. [Online]. Available: <https://www.researchgate.net/publication/221065506_RAPOSA_Semi-autonomous_robot_for_rescue_operations>. [Accessed: 15- Mar- 2018].
8. “Autonomous Search and Rescue Robots: ROS RoboCup Rescue Summer School.”, tedusar.eu, 2012. [Online]. Available: <http://www.tedusar.eu/files/summerschool2012/alexander_kleiner.pdf>.[Accessed:15-Mar- 2018].
9. ["BigDog Overview"](https://www.webcitation.org/664slSojD?url=http://www.bostondynamics.com/img/BigDog_Overview.pdf) (PDF), Boston Dynamics, November 22, 2008. [Online]. Avaiable: <http://www.bostondynamics.com/img/BigDog_Overview.pdf>. [Accessed: 15- Mar- 2018].
10. Cse.psu.edu, 2018. [Online]. Available: http://www.cse.psu.edu/~rtc12/CSE486/lecture15.pdf. [Accessed: 17- Mar- 2018].
11. People.eecs.berkeley.edu, 2018. [Online]. Available: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa11/slides/scan-matching.pdf. [Accessed: 17- Mar- 2018].