

CS-358 Making Intelligent Things
Project : Simultaneous Localization And Mapping

Timofey Kreslo, Sylvain Pichot, Finn Mac Namara, Alonso Coaguila, Florian Dejean

March 2023

Contents

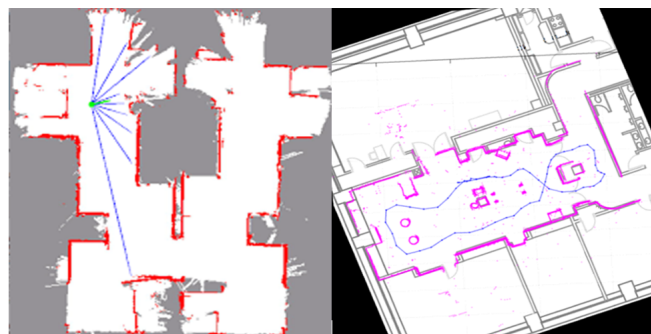
0.1	Description and motivation of the project	1
0.2	How are we going to build it ?	2
0.3	Resources	3
0.4	Risks	3
0.5	Buy list	5
0.6	Milestones	6
0.7	Weekly Tasks	7
0.8	Progress documentation	8
0.9	References	10

0.1 Description and motivation of the project

Our project involves building a robot that will perform Simultaneous Localization and Mapping (SLAM). SLAM is a crucial technology for robotic systems that need to move in dynamic and unknown environments. Robots equipped with SLAM can map and navigate their surroundings without human intervention, making them valuable for various applications such as warehouse automation, search and rescue operations, etc... In general, the sensors used on SLAM robot are quite expensive and big so our objective would also be to reduce the size and price.

We will utilize an ultrasonic sensors system (see following section for the sensor discussion) to implement the SLAM algorithm, and we will develop a software program that will render the robot autonomous via Wi-Fi. This will require integrating various components of the robot, such as the motor control system, the ultrasonic sensor system, and the Wi-Fi communication module.

Overall, the project consists of deploying an autonomous robot that will map its surrounding area using SLAM algorithm with an end goal of displaying a cohesive 2D-map of the area.



Source : Figure 1

0.2 How are we going to build it ?

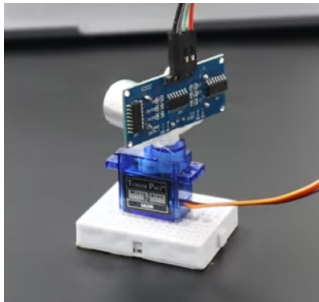
Obstacle detection

Lidar and ultrasound are both commonly used sensors for Simultaneous Localization and Mapping applications. However, Lidar has the following advantages:

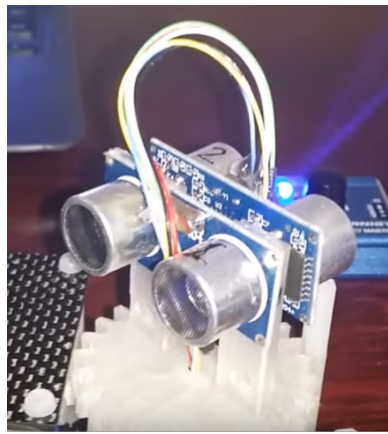
1. **Accuracy:** Lidar sensors provide much higher accuracy measurements than ultrasound. Lidar can measure distances with a precision of a few millimeters, whereas ultrasound can only measure distances to within a few centimeters. Also the resolution is much higher.
2. **Noise immunity:** Lidar sensors are less susceptible to noise and interference than ultrasound sensors. Lidar uses light to measure distances, whereas ultrasound uses sound waves. This makes Lidar less sensitive to acoustic noise and interference.
3. **Mechanical aspect:** Lidar has a bigger range or aperture angle (in degrees) so it would need to be spun around, unlike the ultrasound solution.

Overall, Lidar sensors provide much more accurate and detailed information about the environment, which is essential for accurate SLAM. The problem with Lidar is that it is expensive and has the potential to blow the budget.

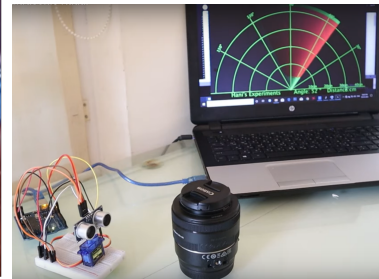
Consequently, we will opt for an ultrasonic sensor which is a cheaper alternative of a Lidar sensor. The sensor gives us the distance D with the detected obstacle. Two ultrasonic sensors will be rotating on top of the robot (back-to-back on a 180 degrees servo, so we can plot any obstacle around the robot) (ex: <https://howtomechatronics.com/projects/arduino-radar-project/>). Software wise : Knowing the direction of the ultrasonic sensor we can create a block (representing an obstacle) of a distance D from the sensor).



Source : Figure 2



Source : Figure 3



Source : Figure 4

Robot localization

To determine where the robot is on the map we will use two stepper motors, the inputs given to the motors allow us to calculate the position of the robot based on its starting point. (<https://www.youtube.com/watch?v=5CmjB4WF5XA>) . Regarding the software, we need to create an algorithm that creates a link between the current position and the new targeted position without going through obstacle. To limit inaccuracy, we will use an IMU to determine the

orientation of the robot (ex: <https://www.youtube.com/watch?v=KMhbV1p3MWk>).

Computational Work

Unfortunately, the SLAM algorithm is too memory bound and computationally slow for a micro-controller. To deal with this problem, we plan having the micro-controller connect through Wi-Fi to a capable external computer, the Command Computer (CC), hence why the micro-controller used has Wi-Fi capability. The Wi-Fi connection also allows the robot to move freely. Therefore, the micro-controller only receives commands for movement and returns the sensors data to the CC.

Algorithm & Data

To create the map and control the robot with the CC, the micro-controller will collect and send to the CC the following elements: the position of the robot, the orientation of the robot (IMU), the distance detected from the ultrasonic sensor and the orientation of the sensor (180 Servo). Then using the data it receives, the CC generates the map and sends new commands to the robot from a distance. We want to use Python and the Robot Operation System (ROS) as a framework since its industry standard and can provide a myriad of useful tools/resources. Regarding ROS and Arduino compatibility, we have found [the following solution](#). If we have time, we might use some basic machine learning algorithms to detect noise.

0.3 Resources

Firstly, we have discovered open-source projects and tutorials available at :

- <https://wired.chillibasket.com/3d-printed-wall-e/>
- <https://3delworld.com/how-to-make-smars-robot-3d-printed-2023/>
- <https://youtu.be/htoBvSq8jLA> (position tracking)

That gives us access to numerous STL files for SMARS parts. These files include components that we can use to construct our SMARS robots for SLAM. This will save us a significant amount of time and effort, as we do not need to design and model these parts from scratch.

Additionally, we have also developed a program, last year, for efficient route finding using Dijkstra algorithms. This program will be instrumental in enabling our robot to navigate through the environment autonomously while avoiding obstacles. The Dijkstra algorithm is a graph-based algorithm that finds the shortest path between two points in a graph. We will modify this algorithm to suit the needs of our project, such as incorporating the sensor data from the camera vision system to avoid obstacles in real-time.

0.4 Risks

1. Mechanical issues:

We are using precise steppers motors, to navigate through the map but we still need to take into account error accumulation. This can be an issue since we are planning continuously updating the robots position via the stepping information of the motors.

Also using tracks could turn out to be a bad idea, because tracks slip when touring on the spot, resulting in screwed up data for odometry (= "use of data from motion sensors to estimate change in position over time").

Solution: We added an IMU sensor to check the rotational information. In case this does not work, we need to consider a simplified design of the chassis, a bit like an automatic Hoover. (That is, only two wheels and one stabilizing thing in front/back).

After all, there is a reason why those vacuum cleaners are built the way they are.

Now, if it still has problems on the localizing part, we need modifying the area. to map and setting up some kind of orientation system (reflective tape around cylinders is common way in industrial environments, an easier way would be to simply add a grid like pattern for the robot to follow on the ground).

2. Sensor accuracy:

Our SLAM robot rely heavily on ultrasonic sensors to navigate and build maps of their environment. These sensors must be accurate and reliable, otherwise the robot will struggle to locate itself or map its surroundings. The sensors should be as good as possible.

Solution: If the ultrasonic sensors are inaccurate, means that we would have to change the placement of the sensors, worst case scenario the sensor has to be right in front of the wall, we would put the sensor in front of the car and make it work similar to a Roomba.

3. Environmental Interference:

The surface on which the robot is must be flat and stable, if those condition aren't respected we will easily lost the position of the robot, making it challenging for the robot to navigate and detect its surroundings accurately, resulting in a poor map.

Solution: We will restrain the robot environment to avoid that, and in case of such environment we will be able to detect them due the gyro (IMU) and stop the algorithm if the environment is unavoidable. An example of such will be a small jump that if the robot is not aware of it, it will get flipped.

4. Computational power:

Building a SLAM robot requires significant computational power to process sensor data, run algorithms, and control the robot's movements. The ESP8266 lacks this power.

Solution: Computation will be done on a laptop with wireless communication.

5. Communication problems:

Our SLAM robot will rely on wireless communication to receive commands or send data, and any communication problems can lead to delays or malfunctions in its real time mapping ability.

Solution: We could try Bluetooth instead but the problem might still arise. We can use a cable but that strongly reduces the mobility of the robot. Another approach will be to await till good communication is made, every time there seem to be a perturbation.

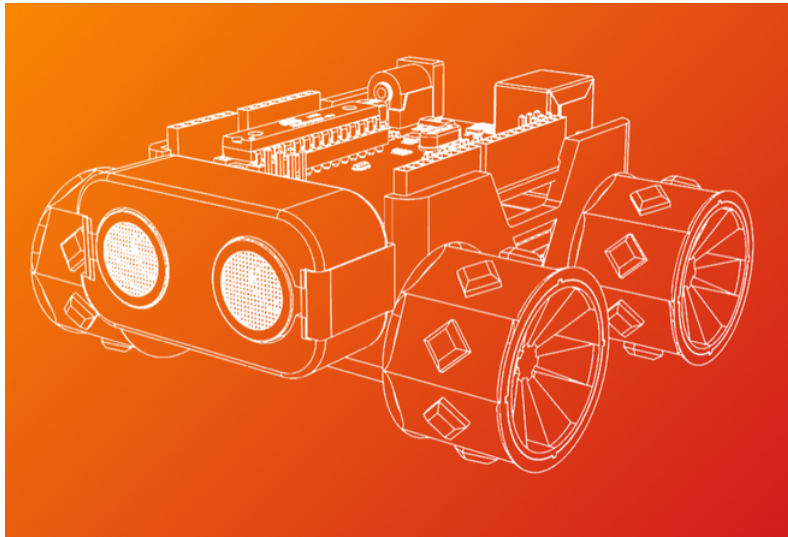
0.5 Buy list

SMARS robot (total 120.- frs):

Electronics 1x SMARS

#	Name	Quantity (Price/item)
1	ESP-32	1 (14.-)
2	Stepper Motor / (replaced by smaller motor 5V)	2 (22.-)
3	A4988 Motor Driver	2 (3,9)
4	Ultrasonic Module Distance Sensor HC-SR04	2 (4,5)
5	Battery Block Coupler 6 Battery 9V [Same as the lego car]	1 (5.-)
6	9 axis ICM-20948 9DoF IMU	1 (24.-)
7	180 Servo Motor	1 (2 for 15.30.-)

We wish to build a second SMARS for optional goal and to have some redundancy in case one part fails. SMARS robot Illustration:



Source : Figure 5

0.6 Milestones

- **Milestone 1 : Build a 3D PRINTED prototype of the chassis with working stepper motors, check track driving behaviour and get communication between the computer and the vehicle+sensors to work (Deadline : 1st May).**

(From Week 6 - 7)

Task 1: Design and print the chassis prototype for our vehicle.

Task 2: Assemble and collect all the necessary hardware and sensors. So we can rapidly combine it with the 3D parts

Task 3: (Start coding the software) Find a way to collect the data from the sensors/Arduino and send them to the CC via Wi-fi.

FIRST SMARS PROTOTYPE should be built on the 7th of April

(From Week 8 - 9)

Task 4: Check that the sensors work well and that the output are correct. So we can change the sensors if we aren't satisfied with the current ones

Task 5: Transfer the output of the robot on a map, so we can start utilizing the data.

Task 6: Create an algorithm that moves the car from an initial position to another given position. Starting to send instruction to the SMARS from the CC to see if are able to correctly control the robot.

- **Milestone 2: Implement an algorithm that allows the robot to create a map of its environment and return to its initial position using the map.(Deadline : 26th May)**

(From Week 10-11)

Task 1: Code the mapping and localization algorithm, needs to be autonomous. So we can start plotting the obstacles.

Task 2: Test the limit of our robot and debug the mapping and localization algorithm, if any problem are detected.

OVERALL WORKING SLAM ROBOT should be able to map on the 20th of May

(From Week 11-13)

Task 3: Have a working SLAM robot.

Task 4: Have a working UI.

- **Optional Goal: Scale up the number of SMARS**

The integration of another robot. The goal would be to reduce the time needed to map out an area. One of the possible solution would be by scaling up the number of robots working together. This is might be quite complicated, and we haven't really thought of it yet but we are leaving the door open.

- **Optional Goal: Sound integration**

It would be amazing if the robot could produce noises/music on command, or produce

noises in real time through Wi-Fi. This would allow the robot to communicate in real time to a person blocking its way or allowing it to call for help if it is stuck.

0.7 Weekly Tasks

This section is to split up the jobs, that have to be done, evenly with the team members. We plan weekly team meetings between us (without the Phd student) to keep us up to date on what is going on and where we are exactly in the project. We find it important to have good communication especially when it comes to asking for help. We will be focusing a lot on talking and understanding the other team members. With that said the tasks here will not always be done by the person mentioned because on the moment problems might arise or even the initial plan wasn't a good solution to the problem. On the other hand, we can guarantee that all team members will have a saying in the more general tasks in the Week n°# column

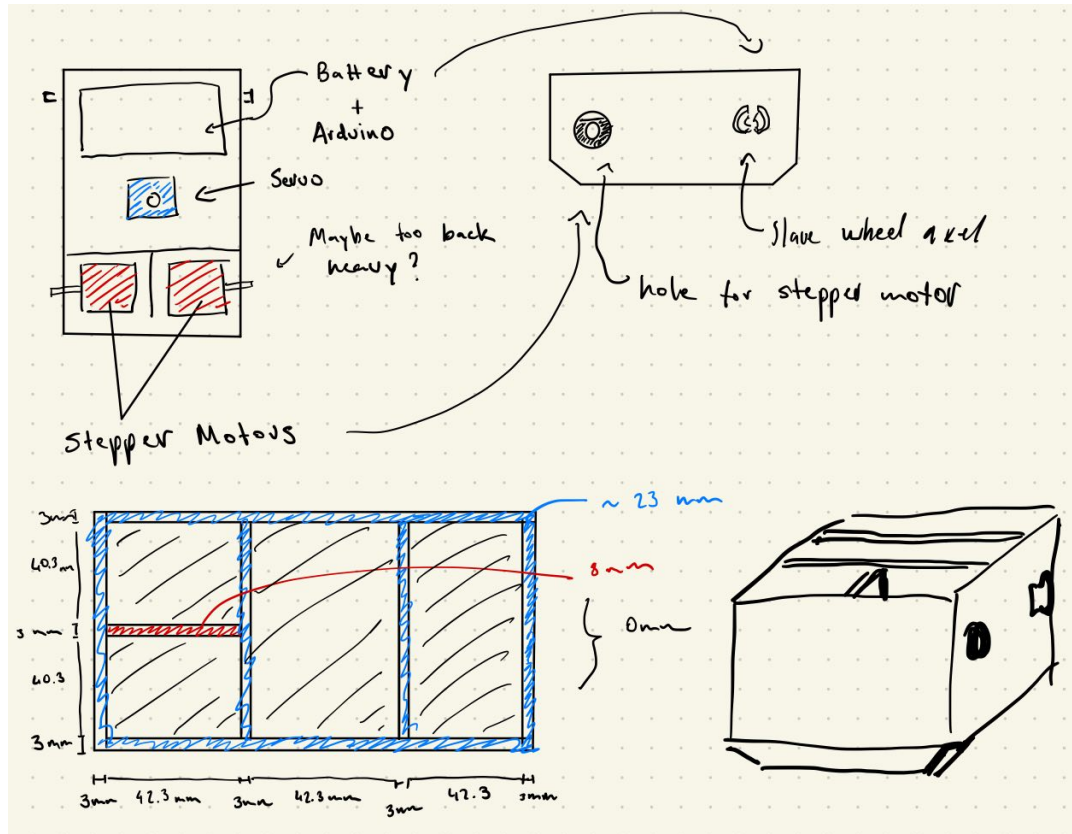
Week n°#	Tim	Florian	Finn	Alonso	Sylvain
6 (Discussion on how we are going to design the robot)	Building a prototype version of the SMARS (with smaller competents that Tim has)	Modelling on Fusion 360, first model with diagonal stepped motors	Modelling on Fusion 360 and brainstorming	Modelling on Fusion360, second model with side by side stepped motors	Modelling and brainstorming on the software a connectivity through Wi-Fi
7 (Analysing SMARS prototype and 3D printed prototype chassis as well as clarifying the electric wiring)	Analyse 3D printed models, and draw an electric circuit	Compare and analyse 3D printed models	Learn how to control a stepped motor	Compare and analyse 3D printed models	Draw an electric circuit to see if all connections are good
Holidays (Start the software research, to learn how to correctly use our equipment and starting to understand how we are going to design and code the software)	Do research on ultrasonic sensor and how to use it and see how we can use python to utilize it's data	Do research on stepper motor and how to use it and see how we can use python to utilize it's data	Advance on the circuit and review prototype with the constraints of the wiring, if done check how to create a 2D map with Python	Advance on the circuit and review prototype with the constraints of the wiring, if done check how to create a 2D map with Python	Do research on IMU and how to use it, and see how we can use python to utilize it's data
8 (Assemble the hardware, and the software)	Debugging the ultrasonic sensors code and real life testing	Playing around with the stepper motors and see how precious they are (code to real life ratio)	Debugging the ultrasonic sensors code and see how it would work with the 360 servo	Soldering and assembling the hardware	Soldering and assembling the hardware
9 (This should just be a debugging week and time to reflect a bit on how we are working as a team)	Debugging	Debugging (cable management if necessary)	Debugging (hardware if cables fall off)	Debugging (re-modelling if necessary)	Debugging (re-modelling if necessary)

For now, we only planned individual tasks till week 9 (for Milestone) to see the dynamic of our group. When the first milestone is achieved, we will plan new weekly tasks for the next milestone.

0.8 Progress documentation

In development from Week 7 onwards...

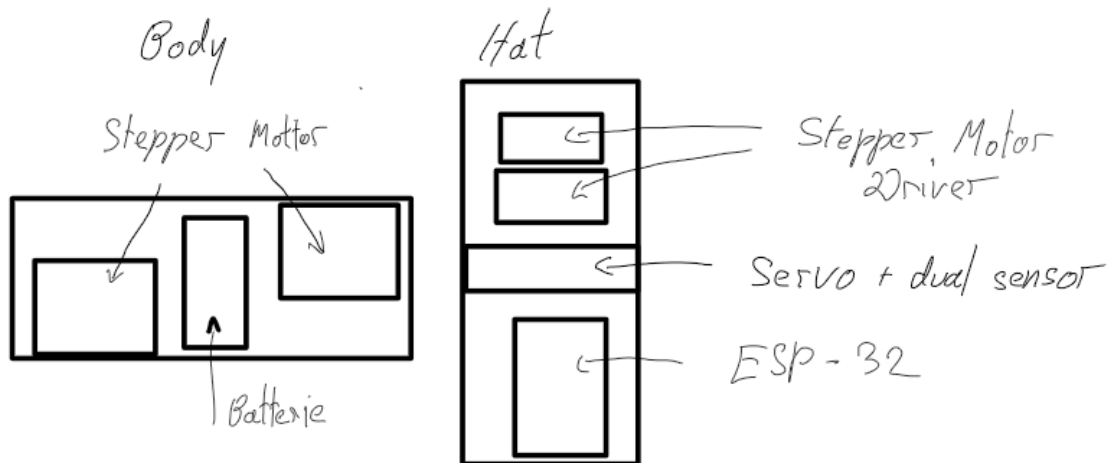
We started to develop the chassis for the robot. First we need consider what it needs to hold, followed by how to arrange it. The robot will hold the 2 stepper motors, a servo, a battery and a micro-controller. For a first prototype, the stepper motors both go in the back, to compensate for the back weight, we would put the batteries and micro-controller in the front, finally in the center the servo holding the sensors.



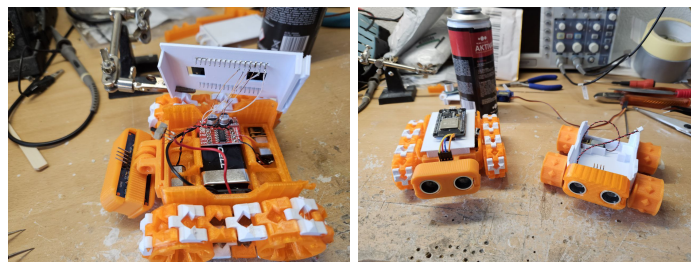
Our second prototype has the stepper motors diagonally split from each other. This version is more compact (smaller in terms of width). We want to test different configurations in turns of weight dispersion:

An alternative design

- such that the center of mass is even



Here is first SMARS prototype with our own equipment. This will allow us to quickly test if tracks are a good use in our project, so we can change our design if this isn't the case.



0.9 References

- [1] Figure 1 : Alain Godot. *Algorithmes SLAM (Simultaneous Localization and Mapping)*. May 2019. URL: <https://www.innowtech.com/2019/05/16/les-algorithmes-slam-simultaneous-localization-and-mapping/>
- [2] Figure 2 : @CodersCafeTech. *DIY Radar With Ultrasonic Sensor And Chat-GPT Generated Arduino Code — Coders Cafe*. Youtube. 2023. URL: <https://youtube.com/shorts/o7DMHJKhpws?feature=share>
- [3] Figure 3 : Channel Everything. *Two ultrasonic sensors on a servo for angle and distance data*. Youtube. 2016. URL: <https://www.youtube.com/watch?v=dHZBOWhLl8g&t=12s>
- [4] Figure 4 : Hani's Experiments. *How to make an ultrasonic Radar*. Youtube. 2021. URL: <https://www.youtube.com/watch?v=xngpwyQKnRw>
- [5] Figure 5 : Kevin McAleer. *SMARS Fan*. 2018. URL: <https://www.smarsfan.com>