POLYTECH
NICE SOPHIA

UNIVERSITÉ
CÔTE D'AZUR

RAPPORT DE PROJET ROBOTIQUE

WALL E

PREPARED AND PRESENTED BY

SATRAGNO ANTHONY
REGIS THOMAS

2024 - 2025

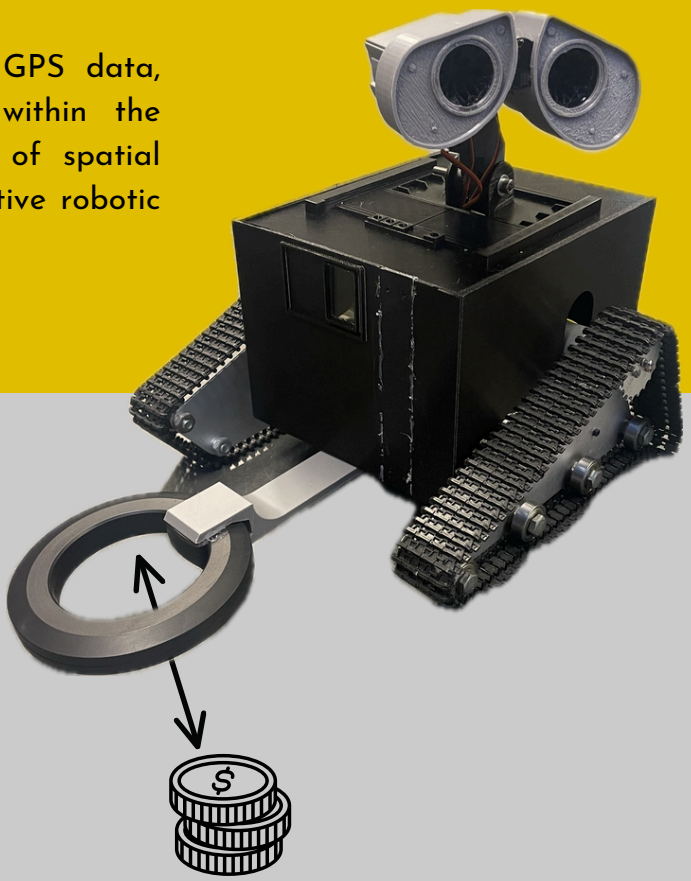
SUPERVISING TEACHER

JUAN FREDERIC

INTRODUCTION

Our robot is a playful prototype inspired by the iconic appearance of Wall-E, designed to detect metals using a standard metal detector attached to its frame. The idea for this project emerged from our desire to combine entertainment and technical utility in a robotic system. By integrating whimsical animations and actions triggered upon detecting metals, WALL E offers an engaging way to explore its environment while fulfilling its detection purpose.
Technically, our project incorporates advanced modeling and simulation techniques. The robot's movements and detection capabilities are simulated in 3D using Gazebo, a powerful robotic simulation tool.

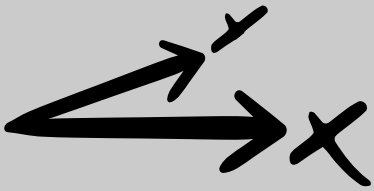
Furthermore, each detection is geotagged using GPS data, enabling precise mapping of detected metals within the environment. This feature introduces an element of spatial analysis, making WALL E a functional and interactive robotic solution with practical and educational applications.



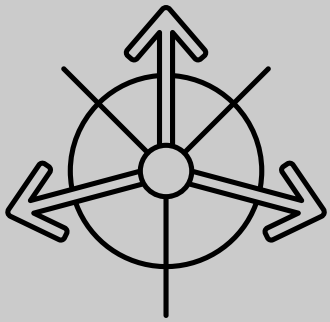
BILL OF SPECIFICATIONS

- The metal detector must be precise enough to locate a 20-cent coin without being disturbed by the metallic chassis.

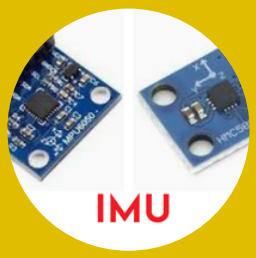
- The robot must be able to move on an XY axis thanks to a mechanical track system driven by two motors



- The various sensors of the robot must be able to indicate its inclination at any angle.



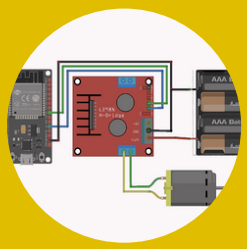
To facilitate the initial testing of the metal detector and the activation of servomotors upon object detection, we opted to remotely control the motors for our first prototype. A significant challenge we encountered was using an iPhone to operate the robot via Bluetooth, owing to Apple's extensive restrictions on such connectivity. Fortunately, we discovered an application named DABLLE that enables Bluetooth pairing with an ESP32 board linked to an H-bridge, thus allowing us to remotely manage the motors and pilot the robot.
To gather information about the robot's rotation, we will use IMUs (Inertial Measurement Units) and will filter and fuse the data using a Kalman filter..



IMU

KALMAN FILTER

BLUETOOTH CONTROL



MOTORS COMMAND

DETECTION

DETECTION

```
80  void loop(){
81
82    int signalValue1 = analogRead(metalDetectorPin);
83    int signalValue2 = analogRead(metalDetectorPin2);
84
85    // Calcul de la différence entre les deux valeurs
86    int signalValue = signalValue2 - signalValue1;
87
88    // Vérification si le signal dépasse le seuil
89    if (signalValue >= signalLevel) {
90      consecutiveCount++; // Incrémentation du compteur
91      if (consecutiveCount >= consecutiveThreshold) {
92        if (!isMetalDetected) { // Si métal non détecté précédemment
93          isMetalDetected = true;
94          Serial.println("Métal détecté !");
95          performDetectionAnimation(); // Déclencher l'animation
96        }
97      }
98    } else {
99      consecutiveCount = 0; // Réinitialisation du compteur si le signal est en
      dessous du seuil
100     isMetalDetected = false; // Réinitialisation de l'état de détection
101   }
102
103   // Affichage de la donnée correspondant au niveau de signal
104   Serial.print("Signal détecté : ");
105   Serial.println(signalValue);
106
107   // Pause avant la prochaine lecture
108   delay(200); // Ajuster le délai selon les besoins
109 }
```

READING THE POTENTIAL VALUES

VOLTAGE CALCULATION

ACTIVATION OF THE SERVOMOTORS

DETECTED EVERY 200 MS

GUIDANCE

```
#define CUSTOM_SETTINGS
#define INCLUDE_GAMEPAD_MODULE
#include <DabbleESP32.h>
```

DEFINE BLUETOOTH LIBRARIES

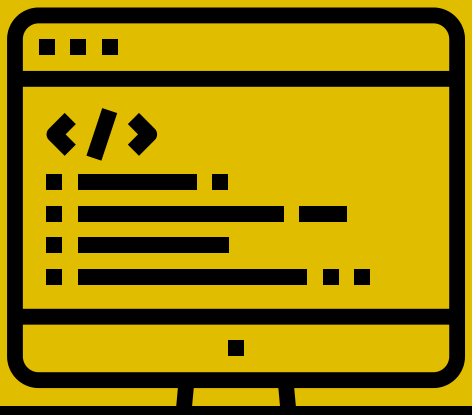
```
void loop()
{
  int rightMotorSpeed=0;
  int leftMotorSpeed=0;
  Dabble.processInput();
  if (GamePad.isUpPressed())
  {
    rightMotorSpeed = MAX_MOTOR_SPEED;
    leftMotorSpeed = MAX_MOTOR_SPEED;
  }

  if (GamePad.isRightPressed())
  {
    rightMotorSpeed = -MAX_MOTOR_SPEED_TURN;
    leftMotorSpeed = MAX_MOTOR_SPEED_TURN;
  }

  if (GamePad.isLeftPressed())
  {
    rightMotorSpeed = MAX_MOTOR_SPEED_TURN;
    leftMotorSpeed = -MAX_MOTOR_SPEED_TURN;
  }

  if (GamePad.isDownPressed())
  {
    rightMotorSpeed = -MAX_MOTOR_SPEED;
    leftMotorSpeed = -MAX_MOTOR_SPEED;
  }

  rotateMotor(rightMotorSpeed, leftMotorSpeed);
}
```

FORWARD COMMAND

RIGHT COMMAND

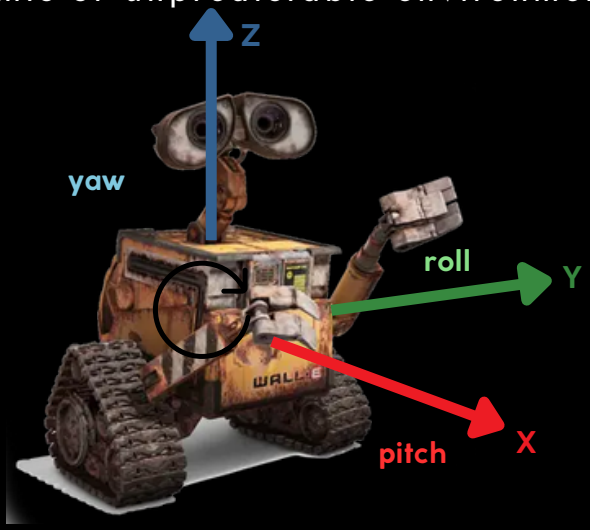
LEFT COMMAND

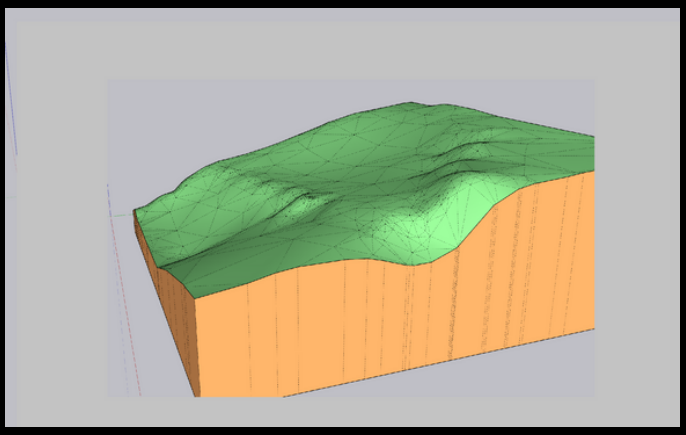
BACKWARD COMMAND



ROBOT'S TRAJECTORY

We have rotation information of the robot around its three axes, which allows us to modify the initially planned trajectory. This data provides a comprehensive understanding of the robot's orientation in three-dimensional space, enabling real-time adjustments to its path. By monitoring the robot's pitch, roll, and yaw, we can ensure it stays on course and makes necessary corrections to navigate effectively. This capability is crucial for maintaining the accuracy of the robot's movements, particularly in dynamic or unpredictable environments.

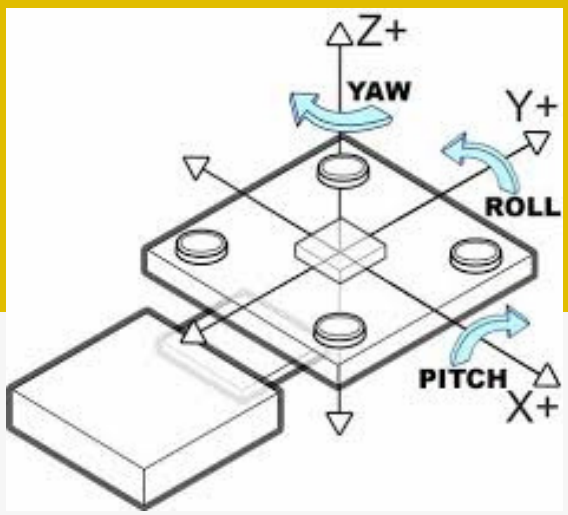




IMU and Extended kalman filter

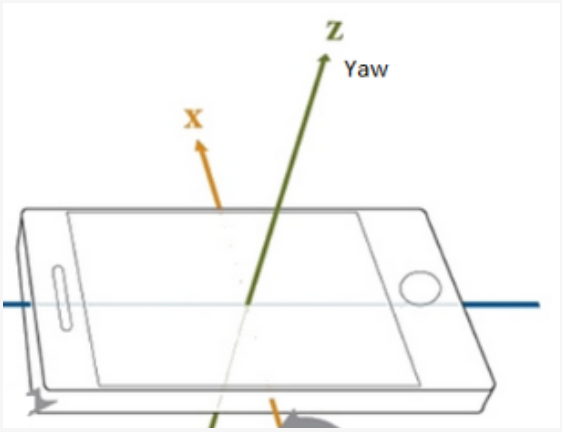
To calculate these rotation angles, we will use an accelerometer to determine the pitch and roll angles, and a magnetometer to measure the yaw angle. The accelerometer will provide data on the robot's tilt relative to the ground, allowing us to calculate the pitch (forward and backward tilt) and roll (side-to-side tilt). Meanwhile, the magnetometer will measure the robot's orientation relative to the Earth's magnetic field, giving us the yaw angle (directional heading). By integrating these sensors, we can obtain accurate and real-time information on the robot's rotational movements, which is essential for adjusting its trajectory as needed.

$$\text{Pitch} = \operatorname{atan}(-ax / (\sqrt{(ay^2 + az^2)}))$$



Accelerometer

$$\text{Roll} = \operatorname{atan}(ay / az)$$



Magnetometer

$$\text{Yaw} = \operatorname{atan}(my / mx)$$

IMU and Extended kalman filter

We will also use a gyroscope to complement the accelerometer and magnetometer. The gyroscope will provide additional data on the robot's angular velocity, enhancing the accuracy of our measurements. By combining the information from all three sensors—the accelerometer, magnetometer, and gyroscope—we can achieve a comprehensive understanding of the robot's orientation and movement. These sensor data will be fused using an Extended Kalman Filter, which allows us to effectively integrate the different types of measurements, reducing noise and improving the precision of the calculated rotation angles. This integrated approach ensures that the robot's trajectory can be accurately adjusted in real time.

$$\begin{pmatrix} \text{Roll} \\ \text{Pitch} \\ \text{Yaw} \end{pmatrix} = \begin{pmatrix} dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{pmatrix} \begin{pmatrix} \omega x \\ \omega y \\ \omega z \end{pmatrix}$$

ωx, ωy, ωz (rad.s-1)

The Kalman filter is a mathematical tool that helps us combine multiple sources of data to get the most accurate result possible. It works by taking in data, predicting what the next values should be, and then updating these predictions based on the actual measurements. This process is repeated continuously, making the filter very effective at reducing noise and improving accuracy.

In our case, we will use an Extended Kalman Filter (EKF) to handle the complex nature of our data. The EKF is a version of the Kalman filter that is designed to work with non-linear systems, like our robot's orientation data.

We will use the EKF to filter and fuse the orientation values from the magnetometer, accelerometer, and gyroscope. Here's how it will work:

1. Prediction Step: The gyroscope provides data on the robot's angular velocity. The EKF uses this data to predict the robot's orientation.

2. Update Step: The accelerometer gives us information about the pitch and roll angles, and the magnetometer provides the yaw angle. The EKF compares these measurements with its predictions and adjusts the orientation estimates accordingly.

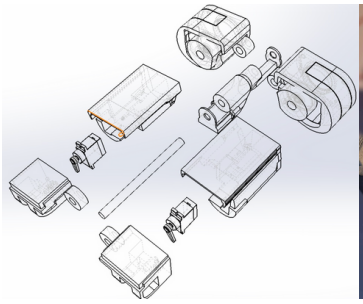
By continuously predicting and updating based on the sensor data, the EKF helps us maintain an accurate understanding of the robot's orientation. This allows us to adjust its trajectory in real-time, ensuring precise navigation and control.

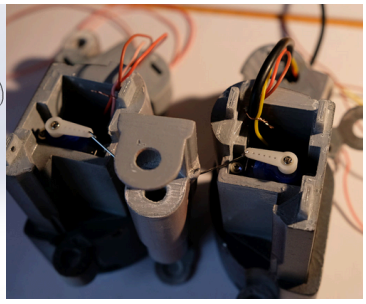
THE ROBOT'S CONCEPTION

The robot's movement operates on a simplified principle of tracked transmission. Each motor drives a sprocket, which in turn powers its track. Since the chassis was provided to us, we had to design the robot's body ourselves and adapt 3D models available online to ensure they fit perfectly with our chassis.

To simplify the process, we had developed the habit of fully modeling the robot in 3D. For this purpose, we used Fusion 360, which, although somewhat basic, was sufficient for our needs.

FROM 3D TO REALITY



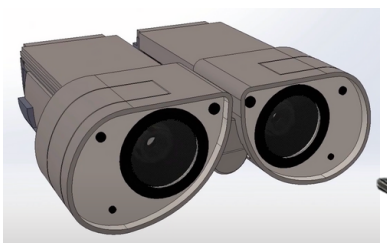


For the eyes of the robot, we utilized 3D models sourced from online repositories, which were then modified to fit our specific design requirements. These models were adapted to ensure they seamlessly integrated with the overall structure of the robot and maintained the aesthetic inspired by Wall-E. The eyes were printed using a standard 3D printer and assembled meticulously to house the necessary electronic components.

During the assembly, careful attention was paid to the servo motors that control the movement of the eyes. We conducted force and torque calculations to ensure the chosen motors could handle the weight of the components while maintaining smooth and precise movements.

These calculations were essential to prevent motor strain and to guarantee the long-term reliability of the system. However, the issue did not lie with the servo motors themselves but rather with the pins meant to connect them to the stationary parts. These pins, responsible for enabling the movement of the eyes, were not rigid enough, leading to a lack of proper functionality.

For the head's orientation, a different issue arose. The transmission shaft, which was supposed to be secured to the servo motor, tended to become loose over time and with repeated tests. Eventually, the shaft detached entirely, causing the servo to spin freely without transmitting any movement. This mechanical flaw highlighted the need for stronger connections and improved design for critical components.



PROJECT'S DEVELOPMENT

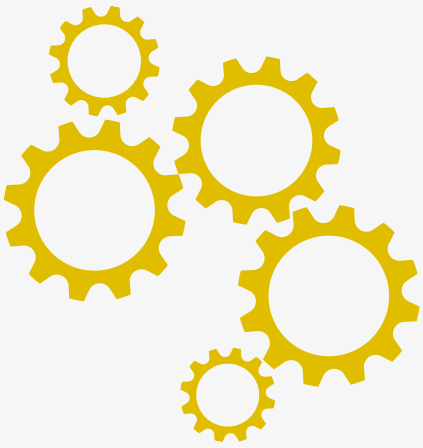
In hindsight, we have learned important lessons from the challenges we encountered. Choosing a pre-made metal chassis initially seemed like a time-saving solution, allowing us to focus on other design aspects. However, this decision brought unexpected complications. The metal chassis made it difficult to properly mount electronic components, forcing us to add extra layers to accommodate them, which increased the complexity of the assembly. Additionally, the poor quality of the servomotors we selected caused frequent malfunctions, requiring replacements and adjustments that delayed the assembly process. These setbacks emphasized the importance of carefully assessing component compatibility and quality early on to avoid unnecessary delays in future projects.

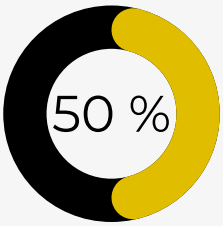
We encountered also several problems with the IMU sensors and the kalman filter.

Firstly, we observed significant drifts in the gyroscope measurements. These drifts accumulated over time, which distorted the calculations of the robot's orientation. Even with the use of the Kalman filter to correct these errors, the data remained unstable, compromising the accuracy needed for reliable navigation.

The magnetometer, on the other hand, was sensitive to electromagnetic interference in the environment. This interference caused unpredictable fluctuations in the yaw data, further complicating the task of data fusion by the Kalman filter.

Moreover, the Kalman filter itself required constant adjustments to remain effective. The calibration of the sensors was complex and needed to be regularly updated to account for changes in the robot's operating conditions. This need for frequent adjustment increased the system's complexity and maintenance.





Prototype

We estimate that we have finished our prototype at 50%, taking into account the initial objectives and those achieved.

COÛT FINANCIER

Personnel

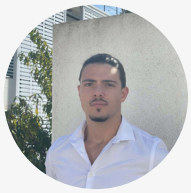
Matières	
Recherche & Dev S9	30 h
Total Personnel	30 h

Coût total	14 407 €

Matériel

Produits	Couts
Moteurs	52€ x2
Chenilles + Boitier	20 €
Matières premières (Filament, Bois, Plexiglass...)	220 €
Câbles + Cartes + Pont en H	10€ + 40€ + 10€
Détecteur de métaux	10 €
servo moteurs + IMU	10 €
Total Matériel	775 €

ÉLÈVES INGÉNIEURS



ANTHONY SATRAGNO

Responsible for modeling, design and mechanical assembly.

06 03 56 22 78
anthony.satragno@gmail.com

THOMAS REGIS

Electronic and IT Manager

06 48 29 66 86
thomasregis33@gmail.com

WHAT'S NEXT ?

The next phase of development will focus on enhancing Wall-E's functionality and autonomy. We plan to equip it with functional robotic arms for interaction and a speaker for audio communication.

A GPS module will be integrated to enable precise positioning and autonomous movement, while a 360° LiDAR sensor will allow real-time mapping of its surroundings. These upgrades will make Wall-E more capable and closer to becoming a fully autonomous and interactive robot.

CONCLUSION

There is still a lot of work ahead, but this series of projects has allowed us to develop our experience and establish a much more organized approach to project-related documentation. Although the start was slow and challenging, it will pave the way for faster development in the future. We are delivering work that is cleaner and more promising for what lies ahead.

BIBLIOGRAPHY

- https://www.youtube.com/watch?v=oj_0g-Z7N80&ab_channel=TV7/
- https://wired.chillibasket.com/3d-printed-wall-e/
- https://www.youtube.com/watch?v=rSCy1_GbC0w&t=312s&ab_channel=hashincludeelectronics
- http://users.polytech.unice.fr/~pmasson/Enseignement/Elements%20de%20robotique%20avec%20arduino%20-%20Moteurs%20-%20Projection%20-%20MASSON.pdf
- https://www.detecteurs.fr/blog/article/comment-fonctionne-un-detecteur-de-metaux.html
- https://www.ensta-bretagne.fr/jaulin/poly_kalman.pdf /
- https://www.youtube.com/watch?v=MinV5V1ioWg&ab_channel=MicWroEngr
- https://www.youtube.com/watch?v=-1tmYPE7MAQ&t=600s&ab_channel=MicWroEngr
- https://www.einfochips.com/fr/sensor-fusion-part-1/