TrailerMate

Project Plan

Version 1.0
10/19/2023

Authors

Abdessamad Amadar
Malaurie Bernard
Sarah Bobillot
Emilie Fraumar
Killian Gonet
Réda Kharoubi
Antonin Laborde-Tastet

Abstract

For our final-year project, we are going to enhance an existing miniaturized electric car model by implementing new functionalities to increase its autonomous driving abilities and enable it to perform reverse maneuvers with a trailer. It means that we will have to set up control loops to continuously monitor speed, direction and achieve complex rear maneuvers. We will also implement object detection features with machine learning algorithms to avoid crashes and collisions. To complete these tasks, we will use various kinds of sensors such as Lidar, camera, ultrasonic sensors and an inertial measurement unit and process the data with a Raspberry PI and Nucleo STM32 card, running on linux with the middleware ROS2 to link each process.
We will organize our work according to the agile management method, that will enhance the communication between team members, stakeholders and make it easier to adjust. As a result, it will allow us to ensure that the product meets with the client's expectations and the incremental development routine will lead to quick product releases.
In the end, through this challenge, the team members skills should cover the entire technical scope of the project.

Keywords

Reverse maneuvers, control loop, machine learning, object detection, IMU, ultrasonic sensors, Raspberry PI, ROS2, Agile

Table of Contents

Abstract

Keywords

Table of Contents

1. Executive summary
  - 1.1. Product description
  - 1.2. Stakeholders
  - 1.3. Features
  - 1.4. Release vision
  - 1.5. Operational requirements
  - 1.6. Reusable features
  - 1.7. Validation plan

2. Team

3. Milestones and deliverables

4. Project planning
  - 4.1. First sprint planning
  - 4.2. Product backlog
  - 4.3. Schedule control

5. Risk management
  - 5.1. Method
  - 5.2. Risk and actions

1. Executive summary

1.1. Product description

Nowadays, marketing and industry are omnipresent. In fact, thirty percent of transports are commercials, therefore we decided to work on simplifying truck driver's tasks, by creating an autonomous car equipped with a trailer. Our new product, TrailerMate, is a life changer.
The objective of our product is to help drivers to park and maneuver with a trailer thanks to autonomous functionalities.
More precisely, our car will be able to:

1. Drive forward autonomously
2. Reverse autonomously: thanks to a reverse gear library
3. Avoid obstacles and operate safely in a real environment

1.2. Stakeholders

As this project is a final school year project, the number of stakeholders is reduced, we can mention:

1. The clients: Ms Chanthery, Mr Le Botlan and Mr Ariba, will receive the final product.
2. The project team: Mr Amadar, Ms Bobillot, Ms Bernard, Ms Fraumar, Mr Gonet, Mr Kharoubi and Mr Laborde-Tastet, will develop the product.
3. The suppliers: they will sell us the components needed for our product.
4. The technical experts: teachers from the Electronic and Computer Science Department, will help us in case of any problems.

Nevertheless, we can consider we are a real company developing a product that will be sold in the future. Our stakeholders would be:

1. **The suppliers:** sell us the components needed for our product.
2. **Our risk engineers:** help us with the legislation and standards.
3. **The sponsors:** help to finance the project.
4. **The project team:** is composed of us, technical experts, engineers.
5. **The testers:** answer a call for tender to help us testing the beta version of the product.

6. **The clients:** see the following Persona.

<table>
<tr><td rowspan="2"></td><td colspan="2">Identity</td></tr>
<tr><td colspan="2">Name: Frank
Line of business: Road transport of goods</td></tr>
<tr><td colspan="2">Features</td><td>Objectives</td></tr>
<tr><td colspan="2">● Qualified and respected company for twenty years
● Lack of new technology
● Spend a lot of drivers’ time when parking the truck</td><td>● Improve the convenience of his employees
● Reduce the human error factor
● Reduce the transit time of his transports
● Improve its competitivity by introducing autonomous trucks</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td colspan="3">● Acquire autonomous trucks that can effortlessly execute reverse parking.
● Enhance the comfort and speed of its drivers.
● Regain the upper hand over emerging companies equipped with cutting-edge technologies.</td></tr>
</table>

7. **The users:** see the following Persona.

<table>
<tr><td rowspan="2"></td><td colspan="2">Identity</td></tr>
<tr><td colspan="2">Name: Paul Torres
Age: 39 years old
Activity: Work for Frank since 2009</td></tr>
<tr><td colspan="2">Features</td><td>Objectives</td></tr>
<tr><td colspan="2">● Qualified and assiduous employee
● Works a lot
● Has to concentrate to park his truck after a long time driving</td><td>● Keep on being the best employee
● Avoid fatigue and back pain
● Reduce accident risk while parking
● Would like to make maneuvering in the dark easier
● Save time while parking</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td colspan="3">● His company purchases an autonomous truck for him.
● Paul experiences autonomous parking.He feels more efficient and less fatigued.
● The company enhances its competitiveness.</td></tr>
</table>

1.3. Features

🔴 : Sprint 1 🔵 : Sprint 2 🟡 : Sprint 3 🟢 : Sprint 4 🟤 : Sprint 5

Highest priority

Car moving	Simulation	Navigation	Reverse maneuvers with trailer	HMI	Security and sensor control
Maintain a constant speed while moving forward 🔴	Make a model of the car in Matlab/Simulink Create a car model using Matlab/Simulink.. 🔵	Detect obstacles. 🟡	Drive straight in reverse for a minimum of 10 meters 🟡	Incorporate fundamental controls (forward movement, left/right turns) into the controller. 🔵	Integrate an emergency button into the controller. 🔴
Adjust speed based on manual or automatic instructions. 🔵	Develop and implement all control laws in Matlab before deploying them in the real robot. 🔵	Detect obstacles with the trailer. 🟡	Execute a reverse maneuver with a 90-degree turn, without road width constraints. 🟢	Create an interface to provide user information when errors occur. 🟢	Enable manual controller reversion when the car operates in automatic mode. 🟡
Take curves according to the sensor values 🔵		Achieve autonomous point-to-point navigation using GPS. 🟤	Identify parking spots and execute automatic parking. 🟢	Design an application to display the planned trajectory in a top-down view and present obstacles detected by the car. 🟤	
Enable obstacle detection to trigger a stop mechanism. 🟢		Detect traffic signs and adjust the car's behavior accordingly. 🟤			
Implement obstacle avoidance 🟢					

Lowest priority

1.4. Release vision

Our work will be separated into six sprints. Each sprint will last two weeks. The first sprint is Sprint 0 where we discover the project, the car, the expectations of our clients-professors, and develop the project plan. The other five sprints will be the ones we will develop the product, according to the priorities of the user stories. The composition of each sprint can be seen in the following table:

Sprint 1 🔴	**Sprint 2** 🔵	**Sprint 3** 🟡	**Sprint 4** 🟢	**Sprint 5** 🟤
Understand the pre-existing code and ROS structure	Adapt the car's speed to a manual or an automatic instruction	Detect obstacles	Be able to stop when there is an obstacle	Detect traffic signs and adapt the behavior of the car to it
Understand the code on ROS 2	Make a model of the car in Matlab/Simulink	Perform a straight reverse gear for at least 10 meters	Perform a reverse maneuver with a 90-degree turn, without road width constraints.	Perform a reverse maneuver with a 90-degree turn and a constraint with the width of the road
Get values from sensors	Implement all the control laws in Matlab before putting in the real robot	Detect obstacles with the trailer	Be able to avoid obstacles	Go from point A to point B autonomously using GPS
Moving forward with a fixed speed	Implement basic controls (go forward, turn left/right) to the controller	Switch from automatic to manual control mode	Be able to detect the parking slot and park automatically	Make an application to get the planned trajectory in top view and the detected obstacles by the car
Look for existing trailers control loops (state of the art)	Be able to take curves according to the sensor values		Make an interface to give information to user when error occurs	
Get an emergency button in the controller				

1.5. Operational requirements

Main requirements

Moves around independently	Detection and handling of obstacles	Automatically parking with the trailer
• Go forward • Reverse • Change of direction	• Stop in front of an obstacle • Avoid obstacle	• Park in a space facilitated by markers

1.6. Reusable features

We will reuse the existing code on github, the pre-installed operating systems on each micro-controllers. At the end of our project, we would like to implement a GPS feature, so we will reuse previous project code or the AutoCab project code with their permission. It will allow transit to a specific localisation using GPS coordinates.

1.7. Validation plan

Sprint 1 🔴

Can we get sensor values?

- ultrasonic sensor:
  - **Initial state:** The car is stopped, close to a large obstacle such as a wall.
  - **Result:** The sensor values we obtain are identical to the distance to the obstacle (+/- the sensor's margin of error).
- LIDAR:
  - **Initial state:** The car is stopped, a few meters from a large obstacle such as a building or a wall.
  - **Result:** The sensor values we obtain are identical to the distance to the obstacle (+/- the sensor's margin of error).
- IMU:
  - **Initial state:** The car is stopped, the sensor is initialized with the trailer straight.
  - **Action:** Place the trailer at a specific angle.
  - **Result:** The sensor values we obtain give the correct angle between the trailer and the car (+/- the sensor's margin of error).

Can we move forward with a fixed speed? (0.5 m/s ; 1 m/s ; 1.5 m/s)

- **Initial state:** The car is stopped.
- **Action:** Press a controller button.
- **Result:** The car is moving at the requested speed.

Can we stop the car with an emergency button?

- **Initial state :** The car is moving.
- **Action:** Press the emergency button.
- **Result:** The car stops in less than 0.6 sec.

Sprint 2 🔵

Can the car adapt its speed to a manual instruction?

- **Initial state:** The car is moving straight forward.
- **Action:** Use LT and RT buttons to respectively increase or decrease the speed.
- **Result:** The car's speed changed

Are our simulation results fitting with our theoretical calculations?

- **Initial state:** The car’s model is idle.
- **Action:** Add a speed command to the model.
- **Result:** The car’s model adapts its speed.

Can the car turn to the left? Turn to the right?

- **Initial state:** The car is stopped or moving straight forward.
- **Action:** Use the joystick.
- **Result:** The car is turning to the side with the angle requested.

Can the car detect obstacles?

- **Initial state:** The car is moving.
- **Action:** Put an obstacle in its path.
- **Result:** The car stops before hitting the obstacle, in less than 30 cm.

2. Team

Here are our project team members and a few main skills:

Abdessamad Amadar	• software (C++/C/Python) • hardware
Malaurie Bernard	• software (C/C++/Python) • hardware • team management
Sarah Bobillot	• control • command • team management
Emilie Fraumar	• software (C/C++/Python) • hardware
Killian Gonet	• software (C/C++/Python) • ROS • AI • system-oriented thinking
Réda Kharoubi	• Software • AI • organization
Antonin Laborde-T	• Software • Hardware • AI (CNN)

Control team:

Abdessamad Amadar, Sarah Bobillot, & Antonin Laborde-T.

- Look for simulation possibilities on Simulink/Matlab
- Take note of existing automotive control projects on
    - control loop to move forward
    - control loop for reverse maneuver

AI team:
Malaurie Bernard, Killian Gonet & Réda Kharoubi.

- Look for what can be done with the set of sensors we have
- Look for a model that fit our problem
- Find a way to produce a dataset or get one from open source projects

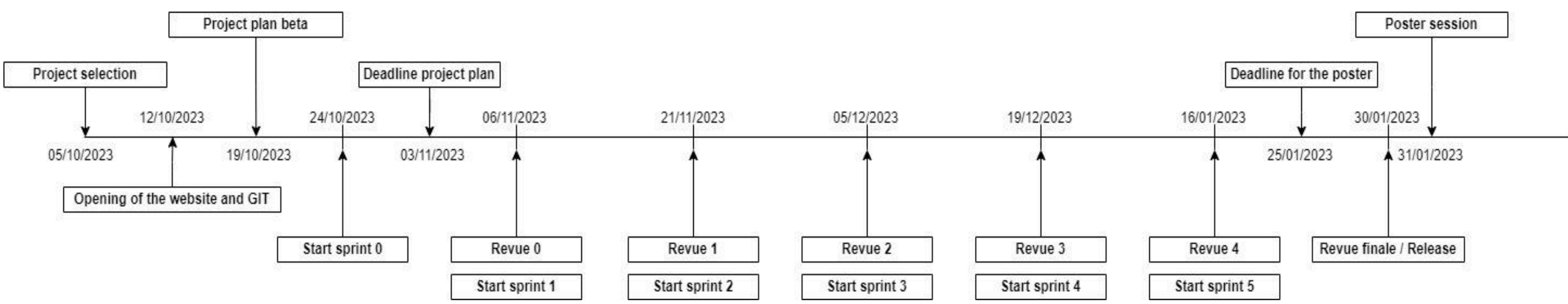
Development team:
Abdessamad Amadar, Malaurie Bernard, Sarah Bobillot, Emilie Fraumar, Killian Gonet, Réda Kharoub & Antonin Laborde-T.

- Implement basic movements control
- Data processing

The scrum master for the first sprint will be Killian Gonet.

3. Milestones and deliverables

Important dates for our project and deliverables:



The deliverables for each sprint are defined as we go along, and will be specified at a later date.

4. Project planning

4.1. First sprint planning

The goal of the first sprint will be to discover the system: centralizing the ROS code, retrieve data from sensors, understand all we have for now and try to go forward with the car.

Timebox: 11/06/2023 to 11/21/2023

Here is the planification of the first sprint:

Story name	Description	Definition of "done"	Member in charge
Understand the pre-existing code and ROS structure.	Understand the global architecture of the global project.	Everyone can understand the global structure of the project. Maybe a map can help.	Everyone
Centralizing the code on ROS 2.	Run the project in the car to get a first iteration of a working car.	The ROS nodes are working on the car and can interact between them.	Malaurie Réda Killian
Get values from sensors.	Set up communication with sensors and get their values.	We can read correct values related to real world environments on the micro-controllers	Emilie Antonin
Moving forward with a fixed speed.	Test with the controller and find the part of the code corresponding to forward travel.	The car can move forward with a fixed speed.	Sarah Abdessamad
Looking for existing trailers control loops (state of the art).	Anticipate sprint 2 and search for already existing truck control loops: what has already been done by companies and searchers.	Knowing approximately what has been done before and having an idea of what we can try to do.	Sarah Antonin
Get an emergency button in the controller.	Set the emergency stop in the red button of the controller.	The car can be stopped using the B button of the controller.	Réda Killian

4.2. Product backlog

The prioritized stories can be seen on the story mapping above.

4.3. Schedule control

To efficiently manage our schedule, we will lean on Agile principles to gauge our progress and slot our tasks into our sprints, starting with the initially planned sprint. Here is how we intend to go about it:

- Sprint Duration: Our sprints are set at a duration of 2 weeks, providing a consistent time frame for both planning and executing our work.

- Sprint Review: At the end of each sprint, we will do a sprint review to evaluate completed tasks and gather feedback. We will also adapt our next sprints based on what we have not achieved.

- Daily Standup Meetings: To ensure the entire team stays in sync and well-informed about the progress of ongoing tasks, we will convene daily standup meetings.

- Weekly Stakeholder Meetings: We have scheduled weekly meetings with our stakeholders to keep them in the loop regarding our progress, collect their valuable input, and ensure their requirements are met.

By integrating these Agile practices into our project management approach, our objective is to have a high level of transparency, maintain a continuous feedback loop, and the flexibility needed in our scheduling and progress tracking. This strategy empowers us to adapt to evolving priorities and effectively deliver value to our stakeholders.

5. Risk management

5.1. Method

During this project, we will use a declination of an agile method called SCRUM. This method allows more flexibility in the project, but can increase some risks. To solve these problems, there exist agile paradigms we will use like XProgramming (code every time in pair), Early Feedback (get information from client before the end of the sprint) and Test Driven Development (write the tests of the application before developing it). We will also add our own management errors detection, by inspiring us from AMDEC analysis. A risk is defined by five characteristics:

- Scope of the error
- Description of the error
- Probability of occurrence (🟩🟨🟧🟥)
- Criticity (🟩🟨🟧🟥)
- Action to solve

This analysis will help us to solve the problems faster when they occur.

5.2. Risk and actions

Scope	Description	Probability	Criticity	Action
INTERNAL	A teammate is unmotivated by the project and doesn't want to continue	🟧	🟧	Detection by making daily/weekly meeting to know the advancement of each person Discuss with him about his motivations and why he is not motivated anymore
	The teammates in work-study contracts can't work in the project during the holidays	🟥	🟩	Plan the sprints tasks according to this constraint (give less work during sprint with holidays to concerned mates)
	Disagreement about choices about the project	🟥	🟨	Make a meeting with all the team to discuss about the motivation of disagreement
	Impossibility to finish all the sprint tasks at time	🟧	🟨	Discuss with the client to the late and replan the tasks to the next sprint
	A teammate is sick and can't work during more than one week	🟨	🟧	Replan his tasks to the other teammates, and discuss to the situation with the client

	A teammate does not know what he needs to do	🟨	🟨	Discuss with all the team to give him new tasks from the current or the next sprint
	Two people works on the same tasks	🟧	🟨	Discuss internally using Discord or Messenger about the tasks Use a Kanban board (Trello/Kantree) and assign tasks to people and get a project monitoring
EXTERNAL	Material failure	🟧	🟥	Ask for S. Di Mercurio if it is possible to replace it fastly If not, find a way to do without it
	The client's needs has changed during a sprint	🟧	🟩	Make weekly meetings with client to avoid this problem If it occurs, discuss with him about that to resynchronize the work
	The deadline has changed	🟨	🟧	Adapt the sprint to the change by adding of remove some tasks
	A material/component costs too much or cannot come at time	🟨	🟧	When design the material needs, think about multiple possibilities of material and component
	The demonstration does not work during sprint review	🟥	🟨	Make a video of the working project before the review to present to the client
	The car is exploding and burst into flames	🟩	🟥	Know where the fire extinguishers are to stop the fire
MATERIAL	The command law is not working	🟥	🟩	Stop the car with emergency button
	A sensor stop working	🟧	🟩	Get a system to give info to the user that a sensor fails Stop the car with emergency button
	The trailer detached from the car	🟩	🟥	Be sure at each test that the trailer is correctly attacher to the car Get someone that follows the car to pick up the trailer if it is detached.

+