



## RAPPORT FINAL

### Course de voitures autonomes (Groupe 2)



#### *Équipe :*

Luiz Felipe DE ANDRADE BARBALHO

Guillaume DEMEESTERE

Ninon DUPEUBLE

Caio ELMOR LANG

Yassine HALLOUL

Mai-Xuân LÊ-DANGUY DES DÉSERTS

Matheus LINS ADLER

14 Mai 2021

## Remerciements

L'ensemble de l'équipe tenait à remercier :

Eric Fenaux pour son encadrement, ses conseils avisés et ses encouragements tout au long de ce projet. Ses remarques nous ont permis de nous remettre en questions à certains moments clés du projet et de continuer à avancer dans la bonne direction.

Franck Tarrufi pour son accompagnement lors des différentes phases PIE et pour l'organisation du projet.

Séverine Bournaud pour ses conseils dans notre manière de mettre en place les outils de gestion de projet et notre gestion de configuration.

L'ENSTA pour nous avoir permis de participer à un premier travail d'ingénierie en groupe sur une durée relativement importante pour des étudiants.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Contextualisation</b>	<b>4</b>
2.1	Enjeu du projet . . . . .	4
2.2	Cahier des charges . . . . .	4
2.3	Présentation de l'équipe . . . . .	5
2.3.1	Répartition des rôles . . . . .	5
2.3.2	Description des rôles . . . . .	5
2.4	Organisation de la gestion de configuration . . . . .	7
2.5	Livrables . . . . .	7
2.5.1	Livrables matériels . . . . .	7
2.5.2	Livrables documentaires . . . . .	7
2.6	Matrice SWOT . . . . .	9
2.7	Analyse des risques . . . . .	10
<b>3</b>	<b>Conception</b>	<b>11</b>
3.1	Description fonctionnelle . . . . .	11
3.2	Planification . . . . .	12
3.3	Premiers et nouveaux objectifs . . . . .	13
3.4	Simulation . . . . .	14
3.4.1	Fonctionnement de la simulation . . . . .	14
3.4.2	Les évolutions par rapport à l'année précédente . . . . .	14
3.4.3	Perspectives d'améliorations . . . . .	16
<b>4</b>	<b>Réalisation</b>	<b>17</b>
4.1	Conception mécanique . . . . .	17
4.1.1	Montage du véhicule . . . . .	17
4.1.2	Agencement de la voiture . . . . .	17
4.1.3	Détermination du rayon de courbure de la voiture . . . . .	18
4.2	Conception électronique . . . . .	19
4.3	LIDAR . . . . .	20
4.4	Servomoteur . . . . .	21
4.5	Centrale Inertielle . . . . .	22
4.6	Loi de Commande . . . . .	23
4.6.1	Première Loi de Commande . . . . .	23
4.6.2	Seconde Loi de Commande . . . . .	23
<b>5</b>	<b>Bilan</b>	<b>25</b>
<b>6</b>	<b>Prise en Main</b>	<b>26</b>
6.1	Où trouver la documentation . . . . .	26
6.2	Quelques directions possibles pour la reprise du projet . . . . .	26
<b>7</b>	<b>Annexe</b>	<b>27</b>

# 1 Introduction

La deuxième année de formation d'ingénieur à l'ENSTA offre l'opportunité de prendre part à un Projet d'Ingénieur par Equipe (PIE). Plusieurs challenges étudiants sont proposés aux élèves qui sont ensuite amenés à travailler en équipe sur un projet de longue durée. Les équipes sont composées d'étudiants aux profils culturels et disciplinaires variés, ce qui rend le projet d'autant plus enrichissant. Outre ce travail d'équipe, l'objectif est de mettre en oeuvre les diverses méthodes d'ingénierie système et de gestion de projet enseignées au cours du cursus à l'ENSTA.

Dans notre cas, nous avons fait le choix de travailler sur le projet *Course de Voitures Autonomes*. En effet ce thème majeur de l'informatique embarquée représente un enjeu important en ce qui concerne la sécurité routière et la fluidité du trafic routier. Nous avons travaillé à notre échelle sur la construction d'un modèle réduit de voiture autonome. Ce rapport rassemble et décrit l'ensemble de la démarche entreprise par l'équipe pour mener à bien ce projet et les outils employés. Nous étayerons la gestion de configuration mise en place, nous permettant de garder une trace de l'ensemble du travail fourni à l'aide notamment d'une tenue rigoureuse de livrables documentaires. Nous reviendrons également sur les étapes de conception du produit final, le résultat obtenu ainsi que les difficultés rencontrées.

## 2 Contextualisation

### 2.1 Enjeu du projet

Dans le cadre du projet *Course de Voitures Autonomes*, l'objectif final est de présenter à la fin de l'année un modèle réduit de voiture autonome sur la ligne de départ d'une course avec d'autres concurrents. La voiture dont nous disposons est partagée avec un autre groupe de PIE participant à la compétition. Le principal défi de ce projet est de réussir à assembler et faire fonctionner la voiture avec sa loi de commande. L'équipe qui présentera la meilleure loi de commande le jour de la compétition implémentera sa version sur le véhicule.

### 2.2 Cahier des charges

Voici le cahier des charges du projet : il nous faut construire un modèle réduit de voiture à échelle 1/10 capable de parcourir un circuit de course et de franchir la ligne d'arrivée (avant les autres participants dans l'idéal) sans intervention extérieure de notre part, en respectant toutes les contraintes imposées par le règlement de la compétition. Nous devons ainsi veiller au fait que la voiture suive une trajectoire performante et ne sorte pas du circuit.

La communication de l'équipe avec le véhicule doit se limiter à l'envoi d'un signal de démarrage et d'un signal d'arrêt. Si jamais pour une raison ou une autre la connexion entre l'équipe et le véhicule est interrompue plus de 2 sec, la voiture doit s'arrêter de manière autonome.

Le circuit de course sera une simple piste balisée par une bordure de 20 cm de hauteur composées d'éléments droits et d'arcs de cercles. La forme de ce circuit ne sera pas connue à l'avance.

Liste du matériel imposé :

- Un kit châssis moteur Tamiya TT02
- Une batterie NimH 7,2V 3000mAh
- Un LIDAR
- Une carte microcontrôleur
- une carrosserie recouvrant le véhicule à plus de 80%

Liste des critères de performance :

- Le véhicule doit avoir une autonomie suffisante pour fonctionner durant toute la course (imposé par la batterie).
- Le véhicule doit se déplacer de manière autonome entre les bords du circuit et les obstacles sans les percuter et sans percuter ses adversaires.
- Le véhicule doit adopter une trajectoire lui permettant de conserver une vitesse aussi grande que possible étant donné les caractéristiques du châssis.

## 2.3 Présentation de l'équipe

### 2.3.1 Répartition des rôles

Pour mener à bien ce projet, nous étions une équipe composée de sept élèves de l'ENSTA. La particularité de cette équipe provient des diversités de profils de chaque étudiant aussi bien au niveau des parcours scolaires qu'au niveau des nationalités. Nous avons défini et réparti des rôles, qui seront décrits dans la section suivante, en prenant en compte les forces et appétences de chacun :

- **Guillaume DEMEESTRE**, filière mécanique : Chef de projet et Porte-Parole
- **Ninon DUPEUBLE**, filière mécanique : Secrétaire et Responsable Pôle Electronique
- **Luiz Felipe BARBALHO**, filière STIC, AST Brésilien : Responsable Simulation
- **Matheus LINS ADLER**, filière mécanique, AST Brésilien : Responsable achat matériel et Responsable montage
- **Caio ELMOR LANG**, filière STIC, AST brésilien : Responsable DATA et Responsable loi de commande
- **Mai-Xuân LE-DANGUY DES DESERTS**, filière mécanique : Responsable agencement châssis, Responsable Sûreté et Sécurité
- **Yassine HALLOUL**, filière mécanique, ENIT : Responsable récupération des données

### 2.3.2 Description des rôles

Pour répondre à la problématique de gestion du groupe et pour assurer une bonne communication avec le corps enseignant, nous avons introduit les trois rôles suivants :

#### **Chef de projet :**

- Assure le bon déroulement général du projet en se tenant informé des avancées de chacun.
- Adapte le cahier des charges en accord avec les encadrants
- Prend des décisions
- Résout les conflits relationnels

#### **Secrétaire :**

- Rédige les comptes-rendus des réunions et l'ordre du jour
- Assure le bon déroulement des réunions
- Met à jour le carnet de bord du projet avec les tâches attribuées à chacun et le planning des réunions

#### **Porte-parole :**

- Gère la communication de l'équipe avec le corps enseignant.

Afin de maximiser l'efficacité et l'avancement de chacun, nous avons ensuite divisé le projet en lots de travail. Compte tenu du sujet traité, nous avons jugé pertinent de se diviser en 2 pôles principaux : le **pôle mécanique** et le **pôle informatique**. Ces deux pôles sont ensuite divisés en plusieurs lots avec pour chacun un ensemble de tâches, de livrables et de responsabilités définis.

#### **Pôle Mécanique**

Responsable achat et matériel :

- Rôle : Etablit la liste de commande de matériel nécessaire, récupère le nouveau matériel, assure son intégrité et sa mise à disposition de l'équipe
- Livrable matériel : Tous les composants nécessaires au montage de la voiture
- Échéance : Première échéance fin décembre (1ère commande matériel), seconde échéance fin février (2ème commande).

Responsable montage mécanique :

- Rôle : Supervise le montage mécanique du modèle réduit de la voiture
- Livrable documentaire : "Montage mécanique". (échéance : 12 février)
- Livrable matériel : Voiture montée et fonctionnelle. (échéance : 12 février)

Responsable agencement châssis :

- Rôle : Optimise l'agencement des composants du châssis
- Livrable : Structure (planche en bois et carrosserie) permettant l'agencement des composants. (échéance : 26 mars)

Responsable électronique :

- Rôle : Teste les capteurs et actionneurs de la voiture, veille à la bonne distribution des puissances.
- Livrable papier : " Montage électrique"
- Livrable matériel : Une voiture avec des branchements corrects. (échéance : 26 mars)

Responsable Sûreté et Sécurité :

- Rôle : Identifie les potentiels risques pouvant nuire à la sécurité, assure le respect de consignes de sécurité
- Livrable : Tableau des risques. (échéance : Décembre)

### **Pôle Informatique**

Responsable DATA :

- Rôle : Assure l'accès et l'organisation de l'espace collaboratif GitHub (code) et Teams
- Livrable informatique : Un dépôt GitHub fonctionnel. (échéance : Fin Décembre)

Responsable Simulation

- Rôle : Améliorer les performances de la simulation de l'année précédente
- Livrable informatique : Simulation. (échéance : 1 mars)
- Livrable documentaire : Organigramme du code, documentation du code (Readme, commentaires...). (échéance : 1 mars)

Responsable Récupération des Données :

- Rôle : Assure une bonne acquisition des données de la voiture (données LIDAR notamment), création d'algorithme de tests pour les composants de la voiture.
- Livrable informatique : Codes d'acquisition des données des capteurs (tests et réels) item Livrable documentaire : Documentation du code (Readme, commentaires...)

Responsable Loi de Commande

- Rôle : Implémenter la loi de commande du véhicule à partir du code de calcul de trajectoire de la simulation et des données des capteurs.
- Livrable informatique : Loi de commande. (échéance : mi-avril)
- Livrable documentaire : Documentation du code (Readme, commentaires...). (échéance : mi-avril)

Cette répartition en lots nous a semblé et s'est avérée être la plus pertinente. Nous avons comme prévu prêté une attention particulière aux interfaces entre les différents lots afin que le passage d'informations se fasse au mieux. Pour cela, nous organisons des points d'avancement réguliers afin de se tenir informés du travail de chacun.

## 2.4 Organisation de la gestion de configuration

Comme nous sommes une petite équipe, nous avons choisi une structure d'organisation horizontale non hiérarchisée pour maximiser le potentiel de chaque membre. Ainsi chacun hérite d'une ou plusieurs responsabilités au sein de son lot de travail.

Étant donné le cahier des charges qui nous est imposé et l'ampleur du projet, nous avons souhaité travailler par itérations et sous forme de circuits courts. De cette manière, nous avons à tout instant une idée précise de l'avancement du projet par rapport à l'objectif final et nous évitons que les lots de travail se dispersent chacun dans leur direction.

Le corollaire immédiat de cette stratégie est de mettre en commun régulièrement les livrables de chaque pôle afin de mettre en place les interfaces entre les lots de travail et de faire des tests d'intégration à l'échelle des composants et/ou du système. C'est pourquoi nous avons décidé de nous réunir en présentiel régulièrement, à hauteur d'une fois par semaine environ, afin de se tenir informés de l'avancement des tâches en cours et d'avoir une bonne dynamique de groupe.

Une difficulté et opportunité supplémentaire est que nous avons dû partager les composants de la voiture avec une autre équipe sans pour autant fusionner nos approches. L'enjeu d'une bonne communication est particulièrement crucial puisque le risque est la perte, la ruine ou, de manière plus générale, l'impossibilité d'utiliser une partie du matériel. Néanmoins, cela nous apporte aussi plus de diversité et de flexibilité en termes de ressources humaines.

La distinction d'approche concernait donc la loi de commande que chacune des équipes a implémenté dans la voiture. Nous avons fait le choix de la baser sur le code de calcul de trajectoire présent dans la simulation de l'équipe de 2019. Il s'agit d'en trouver les points d'amélioration, les corriger et finalement en faire une loi de commande embarquée sur la voiture réelle. Voyant que cet objectif devenait difficile à atteindre, nous avons finalement codé en parallèle une loi de commande plus légère mais fonctionnelle.

Pour nous aider à être efficaces dans cette organisation, nous avons essayé au maximum de nous appuyer sur les outils de gestion de projet, en particulier au quotidien le WBS et le GANTT (pour préciser les tâches des lots et programmer les mises en commun de leur livrables). Vous aurez un aperçu plus exhaustif de ces outils dans la suite du rapport.

## 2.5 Livrables

### 2.5.1 Livrables matériels

L'essence même de notre projet repose sur notre livrable matériel : une voiture électrique équipée d'un lidar capable de détecter son environnement. Nous implémenterons une loi de commande dans cette voiture afin de la rendre autonome.

### 2.5.2 Livrables documentaires

Tout au long du projet, nous avons produit divers documents dans le but de mettre en place une gestion de configuration du projet. Cela permet d'avoir une vue d'ensemble de l'avancée du projet et de faciliter sa reprise par les futures équipes d'étudiants qui seront amenées à travailler sur le challenge.

Nous distinguerons dans ce qui suit les livrables projet des livrables produit. En effet les premiers, généralement associés à un jalon, relèvent de la gestion de projet et du suivi au fil de l'eau. Ils permettent à l'équipe de présenter un état d'avancement du projet afin de vérifier la bonne compréhension des besoins et des attentes. Les livrables produit, quant à eux, sont l'ensemble des productions finales du projet destinées au client. Ils servent aussi de mesure du degré de succès du projet.



## Livrables projet

- **Cahier des charges** : Il permet d'encadrer le projet de *Course de Voiture Autonome* en spécifiant les règles de la courses ainsi que les spécification techniques et fonctionnelles que la voiture devra respecter.
- **Carnet de bord** : Contient l'intégralité des comptes-rendus des réunions ainsi que les tableaux d'action fixant les objectifs à accomplir par les membres de l'équipe.
- **Synthèse d'ingénierie système** (19 décembre 2020) : Il constitue un résumé du projet avec un état des lieux précis des composantes (étude théorique, développement de soft, volet industriel, documentation fournie dans le cadre du projet, ...). Il résume aussi rapidement la démarche d'ingénierie système proposée (application totale, limitée à la descente du cycle en V, avec gestion de configuration, ...).
- **Documents d'ingénierie système** : Cycle en V, diagramme FAST...
- **Documents de gestion de projet** : Le découpage en tâches techniques (WBS) et leur planification détaillée (diagramme de GANTT), la matrice SWOT, l'analyse des risques...
- **Rapport initial de description du projet** (12 février 2021) : Il contient les enjeux et objectifs du projet, son périmètre, la composition de l'équipe, la répartition des rôles, les parties prenantes du projet, la matrice SWOT, l'organisation du projet, les livrables, le budget prévu, l'analyse de risque ainsi que la planification initiale du projet (GANTT initial).

## Livrables produit

- **Rapport final de description du projet** (14 mai 2021) : Il décrit la manière dont le projet s'est effectivement déroulé et revient sur les difficultés rencontrées. Il comporte la mise à jour du planning de réalisation du projet, l'analyse des raisons des écarts éventuels, la liste des différents livrables effectivement réalisés, le bilan et l'expérience acquise. Pris en compte par les encadrants pour l'évaluation.
- **Véhicule autonome** (14 mai 2021) : le véhicule complet et sa loi de commande, en état de marche.
- **Vidéo de présentation du projet** (14 mai 2021) : une vidéo à la fois promotionnelle et informative destinée à présenter le projet. Prise en compte par les encadrants pour l'évaluation.

Chaque lot de travail a également créé les documents nécessaire à la compréhension et à la prise en main du projet. Nous les avons rédigé de sorte que ces qu'ils soient le plus compréhensibles possible par des membres extérieurs au projet. Ils sont à la fois destinés à notre équipe, aux "clients" pour l'exploitation du véhicule et à nos successeurs.

- **Dossier Servomoteur** : Contient la documentation du servomoteur avec une présentation générale ainsi qu'un code de test en python.
- **Dossier Centrale inertielle** : Contient la documentation de la centrale inertielle ainsi qu'un code test en python. La démarche pour tester le bon fonctionnement du composant sans code python est également exposée.
- **Document LIDAR** : Contient la documentation nécessaire à la compréhension du LIDAR. Les explications pour tester le capteur avec et sans Raspberry sont décrites pas à pas avec des illustrations et des vidéos sur lesquelles s'appuyer.
- **Document Raspberry** : Contient la documentation et les instructions nécessaires pour faire fonctionner la Raspberry, connectée ou non au reste de la voiture.
- **Document Montage mécanique** : Contient nos remarques sur le montage du châssis et les points auxquels il faut prêter attention. Des images sont également incorporées pour aider le lecteur.
- **Dossier Mesures Courbure** : Contient deux dossiers correspondant à deux séries de mesures. Un document .txt explique la démarche pour déterminer le rayon de courbure à partir de la commande envoyée au servomoteur.
- **Dépôt GitHub** ([cliquer ici](#)) : Contient tout ce qui est relatif au code. Voir descriptif de la page d'accueil.

## 2.6 Matrice SWOT

Nous avons identifié les Forces (**S**trengths), Faiblesses (**W**eaknesses), Opportunités (**O**pportunities) et Menaces (**T**hreats) de l'équipe. Nous les avons rassemblées dans la matrice SWOT suivante :

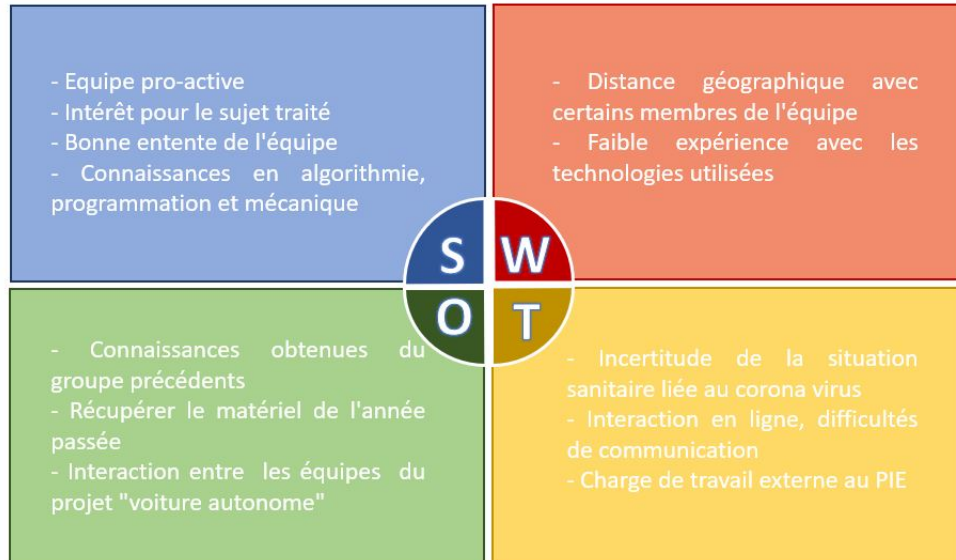


FIGURE 1 – Matrice SWOT

Pour minimiser les effets des *Menaces*, nous avons profité de la situation que nous offre le campus de l'ENSTA pour se réunir en présentiel lorsque les conditions le permettaient. Cela nous a d'une part facilité les échanges et la communication au sein du groupe, et d'autre part de travailler plus efficacement dans une ambiance conviviale.

En ce qui concerne les *Opportunités*, nous avons récupéré le code de simulation développé l'année dernière nous permettant ainsi de partir avec une base. De plus, une des particularités de cette année est la présence de quatre groupes de travail sur le projet *Course de voiture autonome*. Nous avons formé deux duos entre ces quatre équipes : le premier duo devra monter lui même sa voiture tandis que le second repartira de la voiture laissée par le groupe de l'année précédente. Nous avons formé avec l'équipe 1 le premier duo, autrement dit nous avons choisi de monter en repartant de zéro un modèle réduit de voiture autonome. Nous sommes donc entrés en contact avec l'équipe 1 le plus tôt possible afin de tirer le maximum d'avantages de cette interaction entre équipes.

## 2.7 Analyse des risques

Nous avons identifié plusieurs risques potentiels pouvant affecter le bon déroulement du projet et son objectif. Les caractéristiques de ces risques ont été évaluées sur trois critères : la fréquence, la gravité et la maîtrise. Pour prévenir au mieux leurs apparitions, nous avons mises en place des actions préventives qui sont détaillées dans le tableau suivant. :

Risques	Fréquence /10	Gravité /10	Maîtrise /10	Actions préventives
Retard dans les commandes des pièces	5	5	6	Commander les pièces au plus tôt
Retard lié à l'apprentissage	4	2,5	3	Prendre en compte le temps d'apprentissage dans la planification
Incompréhension du travail de l'année passée	3	3,5	7	Rester en contact avec l'équipe de l'année passée
Retard lié à un manque d'investissement	3	3,5	8	Planification régulière de points d'avancement
Friction avec l'équipe 1	6	1,5	3	Parler à l'équipe au plus tôt pour définir les objectifs, rôles et libertés de chacun, en s'accordant également avec les encadrants Mettre en place des documents de gestion pour s'assurer que le matériel est laissé correctement accessible
Mauvaise intégration de certains membres due à la distance	2	4	9	Points d'avancement réguliers
Blessures lors de la manipulation du matériel	4	7	8	Respect des consignes de sécurité
Risques électriques	1	4	8	Respect des consignes de sécurité
Risques de surchauffe de la batterie (d'autant plus que le matériel est partagé)	1	6	8	Attention particulière portée à cet élément
Risque d'endommagement du matériel lors de tests ou de l'utilisation du produit fini	3	5	7	Soin porté à ce détail durant l'élaboration d'algorithmes Vigilance lors des tests et si possible, adaptation du matériel de test pour minimiser les impacts
Risque de demi-tour pendant la course	2	8	2	Non pris en compte car trop complexe
Risque de panne d'un composant électronique ou mécanique	2	7	4	Tester les composants à leur arrivée et commander très rapidement en cas de panne
Risque d'incompatibilité des composants	2	6	2	Réaliser un inventaire dès l'arrivée des pièces Monter la voiture au plus vite pour détecter au plus tôt les problèmes

FIGURE 2 – Tableau d'analyse des risques du projet

Nous avons ainsi porté tout au long du projet une attention particulière aux risques identifiés et agi en conséquence dans notre organisation pour que ceux-ci ne soient pas un frein à la réalisation du projet. Lorsque certains risques se sont présentés, l'organisation en circuit court et ce tableau nous ont permis de réagir rapidement et de ne pas être surpris face aux événements.

Nous avons notamment identifié le risque des retard de commande : malgré nos efforts, certains composants sont venus à manquer et le montage de la voiture a été ralenti. Nous avons anticipé cette situation, et le pôle mécanique s'est alors immédiatement concentré sur les tests et la compréhension des composants à notre disposition. De plus, en utilisant le matériel personnel de certains membres de l'équipe n°1 et en le couplant avec du matériel prêté par l'U2IS, nous avons tout de même pu avancer sur le montage principal et ainsi combler certains retards. Une fois les composants réceptionnés, leur intégration a donc été très rapide, ce qui nous a permis de respecter les échéances du planning.

Concernant la partie informatique, nous avons sous-estimé les difficultés que pose l'implémentation d'une loi de commande à partir du code de l'année dernière. L'échéance du PIE se rapprochant, nous avons dû choisir entre poursuivre sur cette voie et ne pas finir le projet dans les délais ou bien changer d'approche. Nous avons opté pour l'implémentation d'une toute nouvelle loi de commande. Notre démarche sera expliquée dans de plus amples détails dans la suite du rapport.

## 3 Conception

### 3.1 Description fonctionnelle

L'objectif du projet sur lequel nous avons travaillé est avant tout de fournir une voiture capable de concourir et, si possible, de gagner une course. Cependant cette victoire ne peut pas se faire à n'importe quel prix. En effet, une seconde composante qui nous a semblé importante à prendre en compte dans notre analyse de fonctionnalité est d'assurer la sécurité des usagers et du matériel. Nous avons détaillé la décomposition fonctionnelle de la voiture dans un diagramme FAST (cf. **Annexe**) disponible dans nos livrables documentaires.

Les exigences fonctionnelles du projet découlent directement du règlement de la course. En premier lieu, la voiture doit être capable de rouler. Cette fonction est assurée par un châssis, un moteur et tous les composants électriques nécessaires pour permettre la liaison entre les systèmes mécaniques et la batterie.

La deuxième consigne pour la voiture est la capacité de fonctionner sans commandes externes. Pour ce faire, la voiture a besoin d'un ordinateur et d'un code adapté à ses composants physiques. Pour que la voiture puisse prendre des décisions à l'aide de son code, il lui faut également des capteurs capables de percevoir ses variables d'état, ainsi que sa position par rapport à son environnement.

Les composants qui découlent de ces fonctionnalités que nous avons retenus sont les suivants :

- Variateur Brushed/Lipo MSC-30BR-WP - Maverick MV30001
- YLIDAR X2
- Servo S3001 futaba
- Batterie type CRAFTSMAN CS-NS300D37C006, 7.2V, 3000mAh, Ni-MH
- Tamiya 57986 - "1 :10 RC TT-in 02"

## 3.2 Planification

En considérant que la création d’une voiture autonome est un projet complexe mais dont nous pouvons facilement anticiper certaines tâches, nous avons décidé d’utiliser une structure traditionnelle pour la planification des tâches : **le diagramme Gantt**. Pour la réalisation de ce diagramme, il faut avant tout procéder à un découpage du projet en tâches. Nous avons alors commencé cette étape de planification par la création d’un WBS (*Work Breakdown Structure*). Cet outil de gestion de projet permet en effet de rendre le travail plus accessible et maîtrisable.

Nous avons identifié deux grandes sections dans lesquelles diviser les tâches, une plutôt informatique, l’autre plutôt mécanique, ce qui a donné lieu à la création d’un WBS par pôle qui sont ensuite divisés par lots de travail. Les tâches de codage et la création des interfaces entre les différents systèmes de la voiture ont été décrites dans le WBS informatique. Les tâches d’assemblage de la voiture, d’intégrations des composants physiques, des tests des composants et de comportement dynamique de la voiture ont été décrites dans le WBS mécanique.

Une fois les deux WBS en main, nous avons pu commencer à décrire les relations de dépendances entre les tâches dans chaque pôle, afin de définir au mieux les interfaces entre les lots de travail, que ce soit entre les pôles informatique et mécanique ou bien à l’intérieur d’un même pôle. Une fois ces relations identifiées, il est apparu nécessaire de réaliser des points réguliers en équipe complète pour se tenir informés de l’avancement de chaque lot de travail. Ces réunions étaient également l’occasion de revoir nos stratégies afin d’éviter au mieux les retards, de surmonter les difficultés rencontrées et de garder en tête les enjeux du travail de chacun.

Avec toutes ces considérations prises en compte, nous avons commencé la création effective du diagramme de Gantt en faisant une estimation du nombre de séances requises pour compléter chacune des tâches du projet. Pour cela, nous avons divisé la durée présumée en heures de travail de ladite tâche par la durée d’une séance et le nombre de membres qui y travaillent.

Avec tous ces informations nous avons pu organiser les tâches dans le diagramme Gantt (cf **Annexe**), en plaçant des points entre équipes à la fin de chaque tâche importante et en les organisant de sorte qu’aucune ne soit lancée sans que ses prérequis soient remplis. Nous avons vraiment souhaité définir des échéances les plus réalistes possibles en nous inspirant d’une part du planning de l’année passé et d’autre part en essayant d’anticiper les imprévus (cf **Analyse des risques**).

Finalement nous avons dans l’ensemble réussi à respecter les échéances que nous nous étions imposées. Nous avons accumulé du retard dans le montage de la voiture suite à un retard dans la livraison de certains composants essentiels au fonctionnement de la voiture, mais comme expliqué dans la section **Analyse des risques** celui-ci a finalement été comblé.

### 3.3 Premiers et nouveaux objectifs

Durant le premier semestre, nous nous étions fixés comme objectif de repartir d'une nouvelle voiture à monter nous-même et d'améliorer le code de simulation fait par le groupe de l'année passée afin de parvenir à implémenter une loi de commande pour la voiture. Nous sommes donc repartis du travail existant pour la simulation en pensant que cela nous permettrait d'avoir le temps de se concentrer sur une loi de commande efficace.

Au début du projet, nous avons rencontré beaucoup de difficultés avec le programme car une mauvaise version nous avait été transmise. Une fois la version correcte en notre possession, nous avons pu avancer dans de meilleures conditions. Toutefois, l'appropriation du code s'est révélée complexe parce qu'il n'existait presque aucune documentation et que la structure nous est apparue assez contre-intuitive. Elle a donc demandé un travail conséquent à l'équipe informatique. Nous avons tout de même conservé l'idée de repartir de ce travail et d'y apporter des améliorations (détail dans la section **Simulation**).

Durant le dernier mois du projet, devant l'ampleur du travail à réaliser nous avons dû admettre que nous n'avions plus le temps d'adapter le code pré-existant en une loi de commande fonctionnelle. Afin de tenir nos délais, nous avons pris la décision de modifier non pas notre objectif mais la manière de l'atteindre. En effet, nous avions prévu d'implémenter une loi de commande à partir du code de la simulation de l'année précédente mais cela aurait demandé un travail trop fastidieux en raison de sa structure trop complexe qui pourrait être largement simplifiée. Par conséquent nous avons pris la décision de repartir de zéro pour cette loi de commande et de faire quelque chose de simple mais qui permettrait de faire participer notre voiture à la course. Le travail d'amélioration de la simulation est cependant bien effectif et, nous l'espérons, sera repris par les années suivantes.

En ce qui concerne l'équipe mécanique les objectifs sont restés sensiblement les mêmes que ceux prévus au début du projet. Nous avons cependant rencontré quelques difficultés sur le matériels car certaines pièces nécessaires au fonctionnement de la voiture n'étaient pas fournies dans le kit que nous avons acheté. Ce problème a toutefois été atténué grâce à un membre de l'équipe 1 possédant chez lui un kit de voiture télécommandé qui a donc pu nous fournir les pièces manquantes en attendant leur arrivée. Nous avons finalement réussi à monter châssis complet avec tous les composants nécessaires au bon fonctionnement de la voiture ainsi qu'une carrosserie.

### 3.4 Simulation

Comme évoqué précédemment, la stratégie initiale était de reprendre le code de l'équipe 2019-2020 pour valoriser leur travail et avancer plus efficacement vers l'objectif final. Ledit code était une simulation de la voiture sur un circuit, qui intégrait notamment des éléments de calcul de trajectoire, de comportement cinématique et bien sûr d'affichage vidéo.

La stratégie de la simulation a été fondamentale dans le déroulement du projet : elle permet de développer des algorithmes et de les tester avant même d'avoir un prototype prêt à rouler. Cela a pour conséquence de réduire la dépendance entre les lots de travail, notamment dans les premières phases du projet, et donc de diminuer les risques de paralysie générale (à cause d'un retard de livraison de matériel par exemple). Enfin, simuler sur ordinateur est beaucoup plus prudent, rapide et reproductible que tester les lois de commandes directement sur le prototype, au risque de l'endommager.

Le but de cette simulation informatique est donc d'obtenir, en jouant sur certains principes algorithmiques et paramètres de calcul, une base déjà relativement fiable et performante pour notre loi de commande embarquée.

#### 3.4.1 Fonctionnement de la simulation

A chaque pas de temps, le principe de la simulation est de déterminer le point le plus lointain que la voiture peut atteindre à partir des données simulées du Lidar, et d'en déduire une orientation des roues et une vitesse à imposer au véhicule. A la fin du calcul, on obtient le déplacement de la voiture sur le circuit en 2D.

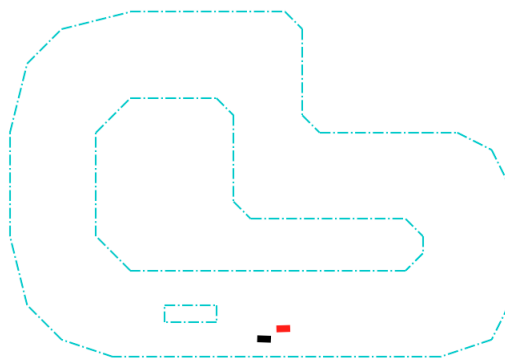


FIGURE 3 – Les premiers tests avec la simulation.

Les lois de comportement sont données par les équations cinématiques qui relient vitesse de lacet, rayon de courbure, accélération radiale et angle de braquage entre elles (avec l'intervention de paramètres propres à la voiture tels que le taux de surbraquage TSB).

#### 3.4.2 Les évolutions par rapport à l'année précédente

Avec cette simulation, le groupe a testé le code implémenté l'année dernière sur différents circuits et avec différents jeux de paramètres afin de chercher des points d'amélioration. Nous en avons trouvé trois principaux :

- La marche arrière engendre des comportements imprévisibles : étant incapable de détecter les obstacles derrière elle (contrainte structurelle), la voiture sortait du circuit. Elle a également de grandes difficultés à repartir en marche avant une fois la marche arrière engagée.
- La "safe zone" (marge de sécurité sur le bord du circuit) est trop importante, et finit par entraver le véhicule dans son mouvement.
- Le code ne fait pas la différence entre un obstacle fixe et un obstacle mobile. Typiquement, la voiture freine excessivement lorsqu'elle détecte un concurrent devant elle sur la piste, alors qu'il ne représente pas un danger.

Dans un premier temps, nous nous sommes donc concentrés sur ces aspects de la simulation. Le développement de la loi de commande quant à lui est expliqué en détail dans la section **Loi de Commande**.

### Marche arrière

Une solution radicale est de simplement supprimer la fonctionnalité « marche arrière » dans le code. Nous n'y perdions pas grand-chose puisque telle qu'elle était implémentée, elle ne permettait pas de se sortir de situations imprévues mais en provoquait de nouvelles. Une autre approche moins extrême que nous avons mise en place consistait à réduire la vitesse de la marche arrière et augmenter un paramètre d'« horizon temporel » de sorte à laisser le temps à la voiture de trouver une nouvelle trajectoire en marche avant. Cela a résolu les problèmes de marche arrière infinie, mais pas de sortie de piste.

### Obstacles mobiles

Une détection basée sur la distance séparant la voiture d'un obstacle ne représente pas fidèlement l'imminence de l'impact. Nous avons donc changé de point de vue pour nous baser sur le temps avant la collision, via la vitesse relative des objets détectés par rapport au véhicule. Pour cela, nous avons dû ajouter une variable mémorisant les anciennes données de sorte à calculer pour chaque incrément d'angle  $i$  du Lidar et à chaque étape  $k$  :

$$v_i(k.\Delta t) = \frac{x_i(k.\Delta t) - x_i((k-1)\Delta t)}{\Delta t} \quad (1)$$

Si cette vitesse est positive, l'obstacle s'éloigne du point de vue de la voiture : il ne représente pas un danger et est donc ignoré.

### Safezone

Une *safezone* trop importante entravait inutilement la progression de la voiture. Mais réduire la marge ne s'est pas avéré plus efficace puisqu'à l'inverse, une *safezone* trop mince donnait lieu à des sorties de route en virage.

Cela est dû au fait que, bien que la simulation soit en mesure de détecter que la voiture progresse à trop grande vitesse pour suivre la trajectoire calculée, elle n'était pas capable d'anticiper le temps qu'il lui faudrait pour décélérer jusqu'à ladite vitesse. En résulte un freinage trop tardif et une sortie de piste.

Après une visite à l'ENS pour voir une piste proche de la configuration officielle de la course, nous avons pu mettre à jour la simulation avec le vrai contour du circuit. Cela nous a permis de la confronter à un contexte plus proche du réel et de décélérer une nouvelle faiblesse majeure : même à faible vitesse, la voiture ne gère pas bien les virages serrés.



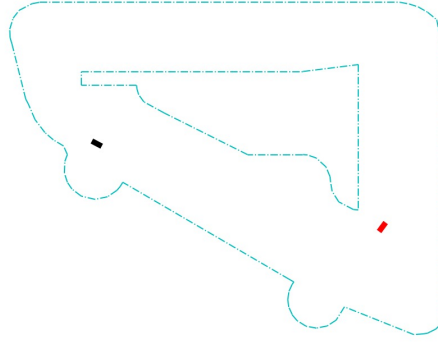


FIGURE 4 – La simulation avec le circuit vu à l’ENS. Le virage que s’apprête à prendre la voiture noire est particulièrement exigeant.

C’est pourquoi nous avons eu l’idée de faire évoluer le concept de *safezone* pour qu’elle prenne en compte ces phénomènes et agisse comme un « tampon » afin de prendre les virages en toute sécurité. Cela s’est manifesté par deux innovations algorithmiques :

- La marge de la zone devant le véhicule devient fonction de la vitesse d’approche de l’obstacle, comme indiqué précédemment. Cela permet d’anticiper le freinage.
- Le code détecte l’intérieur des virages grâce aux discontinuités **spatiales** des distances mesurées par le Lidar et y augmente la taille de la marge. Le but est de forcer la voiture à utiliser toute la largeur de la piste et ne pas prendre de virages serrés trop proche de la corde qui aboutissent systématiquement à une sortie de route.

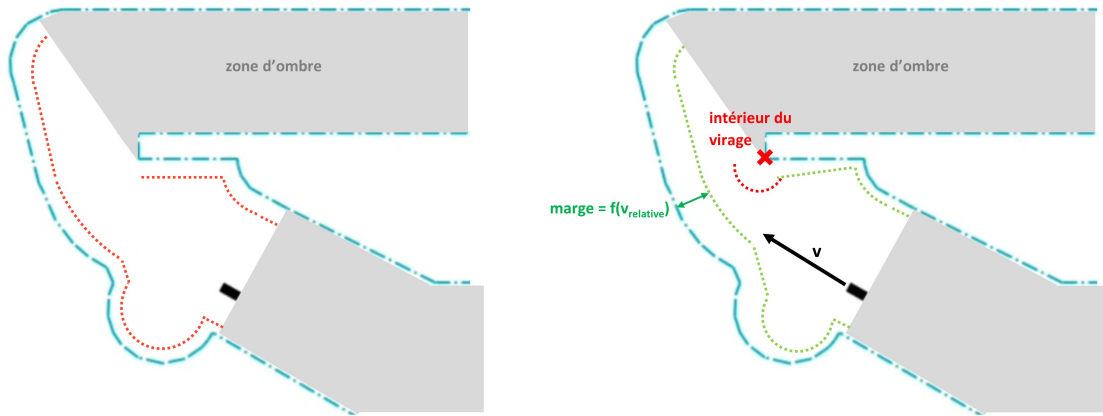


FIGURE 5 – A gauche, le principe initial de la *safezone* (en pointillés). A droite, la version que nous cherchons à implémenter.

### 3.4.3 Perspectives d’améliorations

Les innovations présentées ci-dessus ont montré des résultats encourageants mais méritent d’être poursuivies et davantage rationalisées. Pour l’instant, les critères de définition de la *safezone* sont en effet assez artisanaux et leur influence est difficilement identifiable. Un bon moyen de résoudre ce problème serait de visualiser la zone directement dans la vidéo de simulation, ce qui n’est actuellement pas possible.

## 4 Réalisation

### 4.1 Conception mécanique

#### 4.1.1 Montage du véhicule

Nous avons commencé par monter le châssis de la voiture en suivant les instructions du kit que nous avons commandé. Le corps de notre voiture est constitué d'un châssis Tamiya TT02 et sa propulsion est assurée par un moteur électrique alimenté par une batterie de tension nominale 7,2 Volts via un variateur de vitesse. Remarque importante : la batterie n'est pas fournie avec le châssis, et l'équipe de l'année dernière en avait fait les frais. Leur expérience nous a été utile pour bien l'inclure dans la liste de matériel à commander.

Afin de s'assurer de la fiabilité de notre montage et détecter d'éventuels composants défectueux, nous sommes allés tester la voiture à l'U2IS. Là-bas nous nous sommes aperçu qu'à trop faible puissance, une des roues avant et une des roues arrière de la voiture ne fonctionnaient pas en raison de frottements trop importants. Nous avons donc dû réajuster le montage des roues (cf figure 6).

Pour commander la direction de la voiture, un servomoteur est relié à l'essieu avant et permet d'imposer un angle de braquage des roues.



FIGURE 6 – Ajustement d'une des roues avant de la voiture

#### 4.1.2 Agencement de la voiture

L'agencement de la voiture est un point non négligeable de son montage. En effet, il faut que celle-ci puisse transporter tous les composants lui permettant de fonctionner de manière autonome. Nous avons pour cela simplement découpé une planche en bois fine mais assez robuste pour pouvoir supporter les éléments concernés. Le temps d'effectuer nos tests, nous avons commencer par découper simplement la planche en rectangle (cf. figure 7) sans finition particulière.

Par la suite, nous avons vissé de manière définitive la Raspberry Pi à la planche, ajouté le LIDAR à la bonne hauteur afin de pouvoir détecter les voitures concurrentes, et recouvert le tout d'une carrosserie en plastique réglementaire (cf figure 8). Nous avons également essayé d'optimiser l'arrangement des branchements pour plus de clarté.

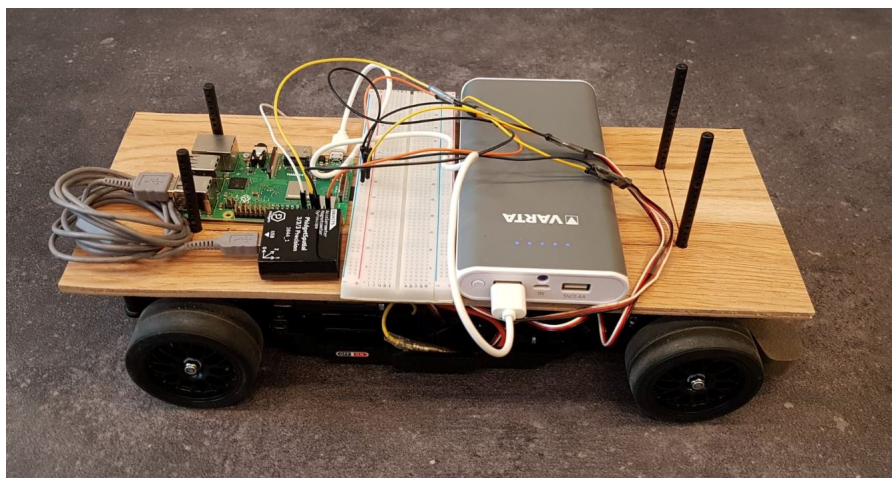


FIGURE 7 – Photo de l’agencement temporaire de la voiture (avant la réception de toutes les pièces) sur une planche en bois



FIGURE 8 – Voiture autonome finale

#### 4.1.3 Détermination du rayon de courbure de la voiture

Afin de mettre en place la loi de commande, nous avons voulu déterminer le rayon de courbure de la voiture à partir de la commande envoyée au servomoteur.

Nous avons souhaité déterminer directement le rayon de courbure en fixant une vitesse de commande et un angle de commande tous deux constants. Nous avons ainsi placé la voiture en régime stationnaire et mesuré l’accélération grâce à la centrale inertielle. Nous avons également mesuré la fréquence de rotation simplement l’aide d’un chronomètre. Ces tests ont été menés à l’ENS, sur le lino prévu pour la course, afin de prendre en compte d’éventuels problèmes d’adhérence.

Pour cela nous avons procédé à une première batterie de test pour différentes vitesses de commande et angles de commande. Cependant l’exploitation des résultats a montré que nous avons atteint l’angle de braquage maximum du servomoteur pour presque toutes les mesures. Celles-ci présentaient donc peu d’intérêt. L’exploitation des résultats est présentée dans le fichier “tableau recap” présent dans la documentation et a

été faites via les formules suivantes.

On part de l'expression de l'accélération mesurée à l'aide de l'accéléromètre :

$$a = \frac{-V^2}{R} \quad (2)$$

Avec  $V$  la vitesse et  $R$  le rayon de courbure de la voiture.

On peut alors en déduire la vitesse et le rayon de courbure de la voiture :

$$V = 2\pi f R \quad (3)$$

Avec  $f$  la fréquence.

Nous avons donc mené de nouveaux tests sur le parking de l'école, cette surface convenait tout à fait puisque qu'elle présentait une grande adhérence et que le lino utilisé pour la course assurait également une adhérence parfaite. De plus cette fois-ci, nous avons utilisé le gyroscope de la centrale inertielle. Celui-ci nous a permis de déterminer la vitesse du lacet notée  $\dot{\psi}$  ainsi que l'accélération latérale notée  $A$  en temps réelle et en moyenne. Les résultats ont été exploités par le fichier python `extraction.py` ainsi que le fichier libre office "resultat" et nous avons obtenu les figures présente dans le dossier "figures". Tous ces fichiers sont présents dans les livrables dans le dossier "Mesure\_courbure".

Nous pouvons tirer les conclusions suivantes :

- Pour chaque vitesse de commande le rayon de braquage est fonction affine de l'angle commande
- Pour chaque vitesse de commande la vitesse de lacet est fonction linéaire de l'angle de commande
- La vitesse réelle semble affectée par la courbure

Nous avons pu modéliser l'accélération latérale par la formule suivante :

$$A = \frac{\lambda.V.\alpha}{(1 + k.V^2) + \beta} \quad (4)$$

avec  $V$  la vitesse de commande,  $\alpha$  l'angle de commande et  $\lambda$ ,  $k$  et  $\beta$  des coefficients déterminés en minimisant le carré des erreurs.

Nous obtenons les coefficients suivants :

$\lambda$	0,025844002540133
$k$	-0,000655090832551
$\beta$	-1,71896951704068

Obtenir une modélisation de l'accélération latérale peut ainsi permettre de la maintenir en dessous d'un certain seuil afin de garantir la tenue de route. Le facteur limitant la vitesse n'a pas été la tenue de route mais la capacité à détecter un virage et tourner les roues très rapidement. Cependant, si dans les années à venir la loi de commande s'améliore, il est possible que l'adhérence rentre en ligne de compte.

## 4.2 Conception électronique

Pour alimenter notre LIDAR et la Raspberry Pi avec la tension adéquate nous avons utilisé un convertisseur DC-DC qui transforme la tension continue de la batterie (7,2V) en une autre tension continue de 5V.

Nous avons également utilisé un *Electronic Speed Controller* plus connu sous son acronyme ESC. Ce composant nous a permis de contrôler la vitesse du moteur et d'alimenter le servomoteur qui est commandé directement par la Raspberry (cf figure 9). Le moteur peut alors propulser la voiture tandis que le servomoteur permettra de la diriger. De plus amples informations sur le principe du servomoteur sont présentes

dans la section **Servomoteur**.

La Raspberry Pi contient notre loi de commande qui permet ainsi à notre voiture de prendre les décisions sur la tenue de roue à adopter en fonction des informations sur son environnement récupérées par le LIDAR. Notre voiture possède alors tous les composants pour être autonome.

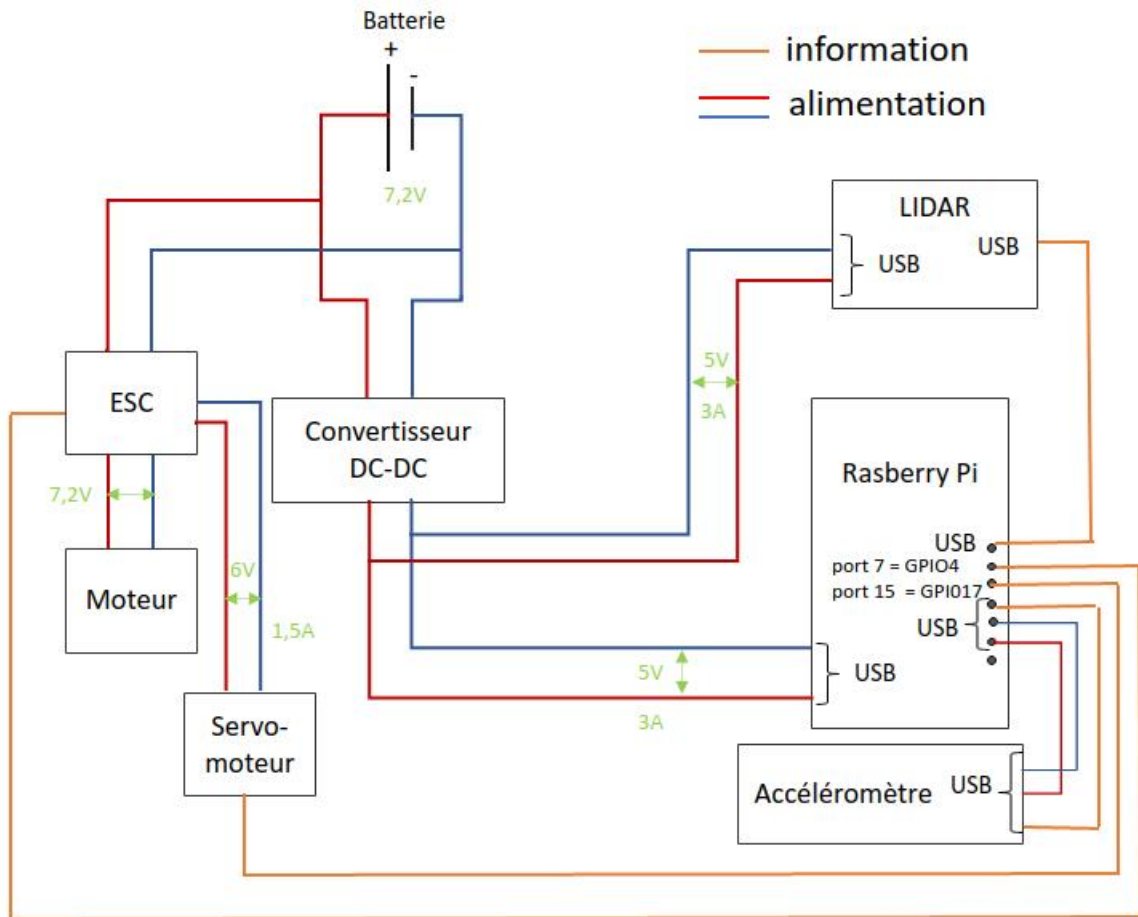


FIGURE 9 – Schéma du montage électronique

### 4.3 LIDAR

Le LIDAR est un dispositif de télédétection par laser venant de l'acronyme anglais *Laser Imaging Detection And Ranging*. Elle permet de mesurer la distance qui sépare le lidar d'un objet en fonction du délai entre l'émission d'une impulsion laser et la réception de l'impulsion réfléchie. Nous utilisons le modèle YDLidar X4. Deux tests ont été réalisés pour vérifier le bon fonctionnement du LIDAR.

Le premier a consisté à tester le capteur directement sur l'ordinateur, en utilisant un logiciel d'exploitation du fournisseur appelé *LidarViewer*, afin de vérifier l'enregistrement et l'exploitation des données en temps réel.

Le deuxième test s'est fait avec le Raspberry, pour bien comprendre le fonctionnement sur le microcontrôleur. Pour cela, on a utilisé des codes open source python montrant quelques commandes utiles pour programmer le LIDAR ainsi que la manière d'exploiter le code dans le cadre de la voiture.

Les tests ont également permis d'observer le format de livraison des données par le dispositif (*arrays* de coordonnées polaires). Tous les détails, liens, codes et sources se trouvent dans la documentation que nous avons produite dans le cadre de la gestion de configuration.



FIGURE 10 – Kit du YDLidar X4 complet

## 4.4 Servomoteur

Le servomoteur est un moteur qui a pour particularité de pouvoir tourner avec une liberté d'environ 180° et garder de manière relativement précise l'angle de rotation que l'on souhaite obtenir.

Pour faire fonctionner un servomoteur, il faut lui fournir en entrée un signal PWM (Pulse Width Modulation). Pour que le servomoteur reste à une position donnée, il faut transmettre toutes les 20 millisecondes (soit à une fréquence de 50Hz) une impulsion d'une longueur comprise entre 1 et 2 millisecondes. La durée de l'impulsion transmises correspond à un angle du servomoteur.

Il existe de nombreux tutoriels Rasberry qui permettent de prendre en main le servomoteur aussi bien au niveau du branchement qu'au niveau du fonctionnement avec des codes relativement simples.

En Python et sur la Raspberry Pi, l'angle du moteur est calculé comme un pourcentage du cycle. Ce pourcentage se nomme « Duty Cycle ». Un signal de 0.5ms correspond à 0° et un signal de 2.5ms à 180. Sachant que notre cycle est de 50 Hz, soit 20ms, cela nous permet de calculer les Duty Cycle pour 0° et 180° comme ceci :

$$— x = 0.5 / 20$$

$$— y = 2.5 / 20$$

Le Duty Cycle correspondant à 0° est 0.025, soit 2.5% et celui correspondant à 180° est 0.125, soit 12.5%.

Le premier test a été fait à l'aide d'un servomoteur au vulso (différent des servos attachés à la voiture) et le code `test_servomoteur` se trouve dans la documentation.



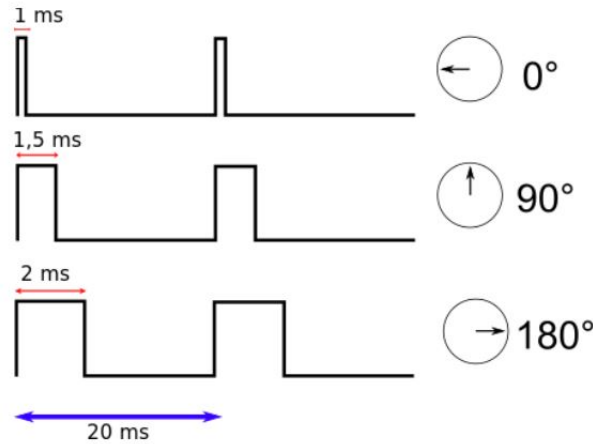


FIGURE 11 – Signal PWM envoyé au servomoteur

## 4.5 Centrale Inertielle

La centrale inertielle est un capteur utilisé pour prendre des données de l'accélération et l'accélération angulaire de la voiture. Dans notre cas, nous avons utilisé le modèle *1044\_1 PhidgetSpatial Precision 3/3/3 High Resolution* qui comprend également un magnétomètre et un gyroscope. Cette centrale nous a été prêtée par l'U2IS.

Pour tester le fonctionnement du capteur nous nous sommes servi en premier lieu de l'interface fournie par *Phidget* (cf. Figure 12) qui permet de voir les informations détectées sur les trois axes x,y,z de l'accéléromètre.

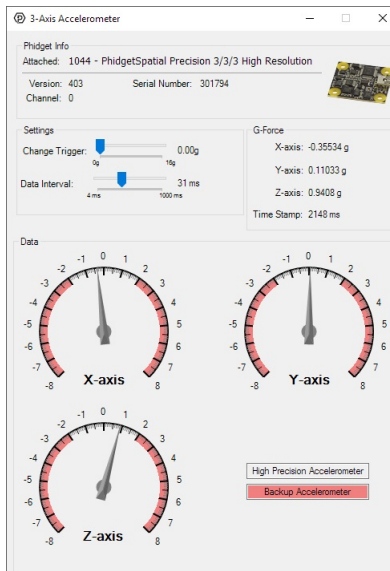


FIGURE 12 – Interface *Phidget22* de l'accéléromètre

Puis nous avons effectué des tests pour récupérer les données de l'accéléromètre en implémentant un code python à l'aide de code fournis également par le constructeur. Le code test se trouve dans la documentation sous le nom de `TEST_centrale-inertielle.py`.

## 4.6 Loi de Commande

### 4.6.1 Première Loi de Commande

A ce niveau d'avancement du projet, le montage mécanique et électronique de la voiture est finalisé, ainsi que l'interfaçage basique des systèmes mécaniques et des capteurs avec la Raspberry Pi. Nous nous sommes donc concentrés sur l'objectif de faire rouler la voiture avec une loi de commande embarquée.

La première loi de commande envisagée devait reprendre les principes algorithmiques de la simulation (cf section **Simulation**). Un des obstacles à cette transition résidait dans les choix de fondations du code faits par nos prédécesseurs, qui ne correspondaient pas à notre vision du problème.

Nous avons donc tenté de l'adapter à de nouvelles bases orientées "objet", en définissant des classes munies d'attributs plutôt que de traiter des dizaines de variables à travers plusieurs documents. En résultat un code plus épuré et compartimenté.

Malheureusement, d'autres raisons nous ont poussé à suspendre nos travaux sur cette loi de commande. Les tests de l'équipe 3-4 ont en effet montré que les lois basées sur le code de nos prédécesseurs ne fonctionnaient pas sur la Raspberry, faute de puissance de calcul d'après eux. Et de manière plus générale, toutes les équipes à ce stade là s'étaient tournées vers d'autres approches beaucoup plus simples mais fonctionnelles (cf partie suivante).

Nous avons donc décidé de développer notre propre loi de commande afin de remplir notre objectif de faire fonctionner la voiture en autonomie. Néanmoins, nous croyons toujours que la première loi de commande a le potentiel d'être la plus efficace si elle est optimisée pour la Raspberry.

### 4.6.2 Seconde Loi de Commande

Notre objectif premier étant que la voiture roule sous une loi de commande qui nous est propre, nous nous sommes tournées vers l'approche de développement d'une loi de commande plus simple, mais capable de diriger la voiture. Ce fut également l'approche des autres équipes de PIE travaillant sur la Voiture Autonome et elle a donné de très bons résultats lors de la course à l'ENS, le 16 avril 2021.

Le principe de cette loi de commande est simple : elle récupère les données du LIDAR, elle fait ensuite une moyenne glissante des données obtenues afin de lisser les erreurs de mesures, c'est à dire qu'au lieu de considérer la distance mesurée pour l'angle  $i$ , on considère la moyenne des distances mesurées pour les valeurs d'angle autour de  $i$ . Enfin, on détermine dans ces données "lissées" quel est le point le plus éloigné de la voiture dans la partie qui est devant elle et on détermine la courbure nécessaire pour s'y rendre en conservant l'approche d'un rayon de courbure maximal.

Cette loi de commande est désormais fonctionnelle et sera utilisée lors de la seconde course à l'ENS en juin. Les pistes d'améliorations possibles sont les suivantes :

- Mettre en place une marche arrière en cas de collision avec un mur : la voiture devrait être capable de repartir aisément avec un recul de quelques dizaines de centimètres.
- Mettre en place un système de vitesse relative comme dans la simulation : lors de la course du 16 avril, cela a manqué aux voitures les plus performantes.
- Mettre en place la détection des angles afin de déporter la voiture à l'extérieur du virage avant son entrée dans le virage. Cette mesure a été mise en place par les équipes 3 et 4 lors de la course à l'ENS du 16 avril et a bien fonctionné. Notre simulation en est également équipée.

Enfin nous avons pu tester cette loi de commande à l'U2IS à l'aide de panneaux en bois matérialisant le circuit, mais sans la présence d'autres participants. La course de juin nous le permettra, mais nous conseillons à nos successeurs de s'y confronter le plus vite possible directement sur la piste de l'U2IS. Un test dans la



simulation peut également s'avérer fructueux.

La mise en place d'une loi de commande et la prise en main à titre individuel de tous les composants , ainsi que l'ajustement des différents paramètres est chronophage, il faut donc veiller à démarrer cette étape suffisamment en amont de la course.

## 5 Bilan

Le PIE a été l'occasion de mettre en pratique pour la première fois les outils de gestion de projet et d'ingénierie système enseignés à l'ENSTA. Le processus de compréhension et de réalisation de ces outils s'est avéré crucial pour le bon déroulement du projet, assurant une organisation conjointe de l'équipe et de ses différents pôles, favorisant leur synchronisation et coopération dans les différents aspects que le projet impliquait. La mise en place de certains outils ainsi que la tenue à jour de la gestion de configuration a pu nous sembler quelque peu fastidieuse au début du projet, mais leur importance s'est révélée une fois le projet bien entamé. D'une manière générale, les délais et objectifs spécifiés dans ces outils ont été respectés. Dès que nécessaire, la modification de ces délais et objectifs s'est avérée raisonnable et non compromettante pour la réussite du projet, démontrant la bonne gestion de l'équipe et sa capacité à s'adapter aux changements et obstacles apparus tout au long de l'année.

En termes de matériel, le projet a été bien développé. Tous les composants ont un test de contrôle initial qui peut être facilement effectué en suivant les informations des livrables documentaires, et l'assemblage de la voiture complète a également été un succès, toutes les limitations mécaniques et électriques ayant été bien contournées. La simulation a été peu modifiée sur le fond par rapport à l'année précédente (beaucoup plus sur la forme), mais à des endroits cruciaux pour l'approche du problème. Une bonne voie de poursuite pour les futures équipes serait l'intégration du code de la simulation et de ses innovations dans une loi de commande : l'objectif serait d'améliorer les performances de la voiture sur piste, en modifiant les détails nécessaires et en perfectionnant son algorithme de contrôle de trajectoire. La deuxième loi de commande utilisée dans la voiture n'a en effet pas été conçue comme une solution à long terme : elle a surtout vocation à repérer et comprendre l'émergence de comportements inattendus propres au prototype réel. Rien n'empêche néanmoins de retravailler cette dernière puisqu'en l'état, c'est bel et bien elle qui nous permet de remplir notre objectif principal : livrer un modèle réduit capable de concourir sur piste en autonomie.

Ce PIE nous aura appris à être rigoureux aussi bien dans notre façon de d'utiliser les outils de gestion de projet que dans notre manière d'interagir au sein d'une équipe de travail. Nous sommes plutôt satisfaits du travail auquel nous avons abouti et avons hâte de participer à la course de l'ENS qui se tiendra au mois de juin prochain !

## 6 Prise en Main

Cette section du rapport concentre l'essentiel des informations nécessaires à la prise en main du projet par des autres équipes dans le futur.

### 6.1 Où trouver la documentation

Tous les compte rendus, outils de Gestion de Projet/Ingénierie Systèmes, rapports, références et documentation des composants réalisés par notre équipe sont disponibles sur un référentiel GitHub :

[github.com/caiolang/voiture-autonome](https://github.com/caiolang/voiture-autonome)

### 6.2 Quelques directions possibles pour la reprise du projet

#### Côté logiciel

1. Poursuite de l'implémentation du code tiré de la simulation dans une loi de commande.
2. Insertion des principes algorithmiques "time to impact" et "détection des virages" (voir Simulation (3.4)) dans la deuxième loi de commande.
3. Test de la deuxième loi de commande dans la simulation. De manière générale, tester si possible les nouvelles approches en simulation avant de les essayer sur le prototype.

#### Côté prototype

1. Le rayon de braquage en butée de notre prototype est assez médiocre en comparaison à ses concurrents de l'ENS. Rien de rédhibitoire cependant si la loi de commande est efficace.

## 7 Annexe

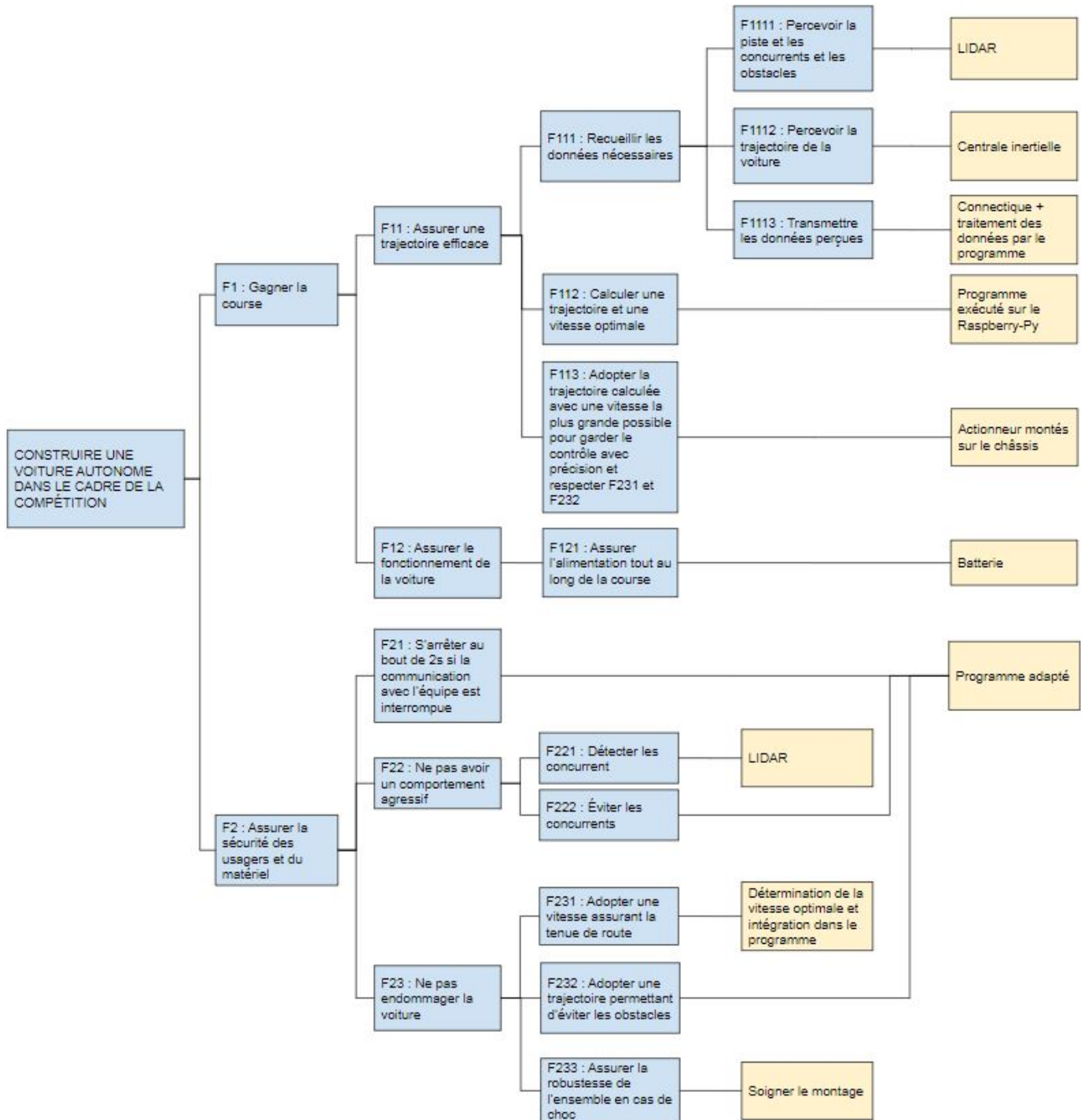


FIGURE 13 – Diagramme FAST



FIGURE 14 – Gantt sur la période janvier-mai