



PFE - VOITURE AUTONOME

Rapport Ingénierie Système

Auteurs Nicolas Hammje

Ibrahima Toure

Nathalie Lebel

Marc Dro

01. 12. 2025

Table des matières

1	Présentation du projet d'architecture système	3
1.1	Contexte et objectifs	3
1.2	Éléments de planning et d'organisation du projet	3
1.3	Organisation du travail	4
2	Architecture opérationnelle	6
2.1	Analyse de l'environnement	6
2.2	Interfaces opérationnelles	7
2.3	Analyse des besoins	7
2.4	Cycle de vie et scenarios opérationnels	8
3	Architecture fonctionnelle	10
3.1	Exigences fonctionnelles	10
3.2	Développement des fonctions	10
3.2.1	Faire un tour de piste rapide	10
3.2.2	Éviter les obstacles	11
3.2.3	Ne jamais être à l'arrêt	11
3.2.4	Toujours être en communication	12
3.3	Architecture fonctionnelle dynamique	13
3.4	Modes de fonctionnement	13
3.4.1	Cartographie de la piste	13
3.4.2	Génération de la trajectoire	14
3.4.3	Course	14
4	Architecture physique	16
4.1	Protocoles de communication	17
5	Architecture logicielle	18
5.1	SLAM	18
5.2	MPC	18
6	Architecture organique	21
6.1	Organisation des composants	22
6.2	Flux MEI de la voiture	22
7	Dimensionnement du système	24
7.1	Dimensions physiques	24
7.2	Performance	24
7.3	Modes de dysfonctionnements	25
8	Conclusion	27

1 Présentation du projet d'architecture système

1.1 Contexte et objectifs

Tous les ans, l'ENS Paris Saclay organise une Course de Voitures Autonomes échelle 1/10e entre écoles et universités baptisée CoVAPSY . Le hardware (partie motrice, châssis) est identique entre les équipes, la différenciation se faisant uniquement à travers l'heuristique et les algorithmes de navigation choisis. Les voitures sont équipées de LiDAR et/ou de caméra(s) en fonction des besoins, et la course se déroule en deux phases.

La première phase, dite de qualification, se déroule sur circuit libre. L'objectif est d'accomplir un tour de circuit le plus rapidement possible, avec deux essais autorisés. Dans un deuxième temps, le circuit change et un obstacle statique est ajouté sur la piste, visant à mettre en avant le besoin d'éviter les obstacles. La somme des deux meilleurs temps est retenu et sert à établir un classement qui détermine la position de départ des voitures pour la course principale.

La deuxième phase est une course en deux manches, sur un nouveau circuit, où 6 voitures s'affrontent en même temps. Le but est d'accomplir 5 tours de circuits, et des points sont attribués en fonction de l'ordre d'arrivée. Le gagnant de la compétition est celui qui gagne le plus grand nombre de points.

Sorbonne Université a déjà participé à trois éditions avant 2025, et le projet vise à améliorer les véhicules d'année en année. En 2024, l'équipe Sorbonne Université a couru avec deux véhicules utilisant deux heuristiques différentes, arrivant 2e et 4e et remportant le Prix de l'innovation technologique pour l'utilisation de SLAM pour la navigation de l'une.

L'objectif pour 2025 est d'introduire une composante utilisant de la vision, pour détecter les obstacles et éventuellement naviguer dans le circuit.

1.2 Éléments de planning et d'organisation du projet

Trouvez ci-dessous le planning des séances de PFE, le diagramme de GANTT, et la matrice des tâches.

Tout au long du projet, les erreurs, bugs, et autres problèmes rencontrés sont systématiquement rapportées soit dans un PDF déposé sur le GitHub, soit dans une issue ouverte sur GitHub si le problème n'est pas résolu ou qu'il implique plusieurs collaborateurs. Cette rigueur permet à la fois de ne pas perdre du temps à retrouver la solution si l'on rencontre le même problème une deuxième fois, mais sert également de ressources aux autres ainsi qu'aux futurs participants, qui peuvent y trouver un répertoire de tous les problèmes et leurs solutions.

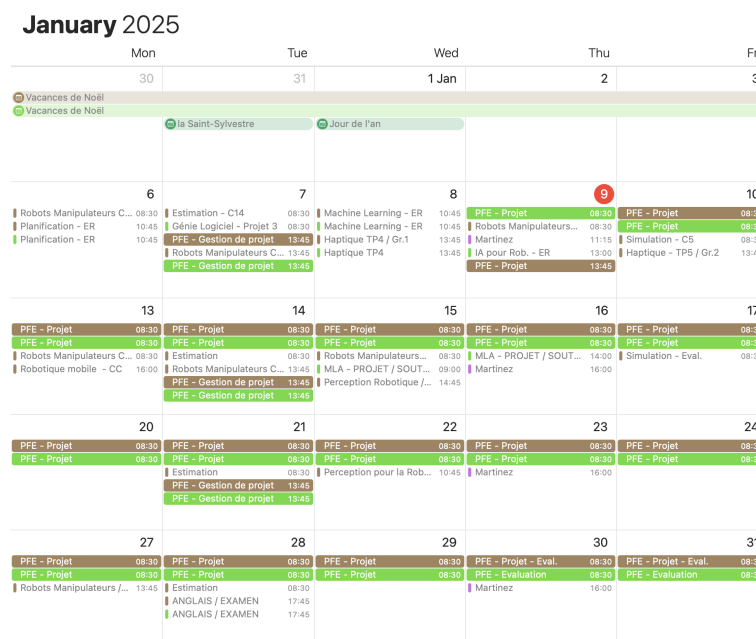


Figure 1.1 Planning des séances PFE du mois de Janvier 2025

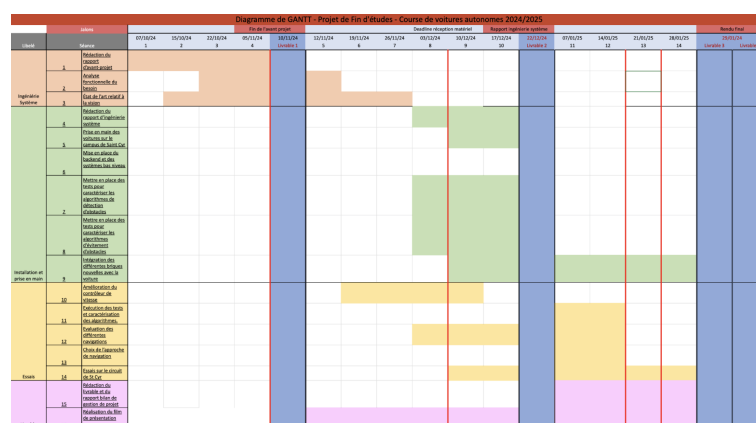


Figure 1.2 Diagramme de GANTT

1.3 Organisation du travail

L'utilisation d'un GitHub facilite le travail, à la fois pour la collaboration intra-équipe, mais surtout pour la transmission facile et rapide du projet aux équipes suivantes. Le repo de 2025 reprends le repo de 2024, qui reprend le repo de 2023, etc. L'utilisation de plusieurs branches (dev, main) permet également de rapidement déployer de nouvelles fonctionnalités aux voitures, tout en gardant une version des programmes ayant fait ses preuves. Une fois que les nouvelles fonctionnalités ont été rigoureusement testées et éprouvées, les changements sont fusionnés avec la branche main.

Tâches \ Acteurs	Echéance	Nathalie	Marc	Ibrahima	Nicolas
Phase 1 - Avant Projet	22/12/24				
Établir les lois de commande du véhicule dans un contexte hautement dynamique				S	R
Mettre en place le pipeline de vision		P	R		S
Mettre en place d'un modèle de classificateur pour détection d'obstacles		R	P		A
Faire l'état de l'art concernant la détection et/ou l'évitement d'obstacles				R	A
Rédiger du rapport d'avant projet		P	R	P	A
Phase 2 - Mise en place	30/01/25				
Installer du matériel sur la voiture			P		R
Intégration avec l'architecture système existant			P	P	R
Évaluation de la qualité de la classification		R			A
Récette de données pour entraîner le modèle		R	P		A
Récette de données pour identifier les paramètres du MPC				P	R
Amélioration des systèmes bas niveau pour baisser latence de commande			P	P	R
Amélioration de la localisation du véhicule dans la piste			P	S	R
Quantification de la performance de la classification et ajustement de la heuristique si nécessaire		R	P		A
Tests de calculs déportés pour un meilleur overhead - Réduction de la latence de détection		R	S	S	P
Installation du RB5, intégration et quantification de la performance vs R Pi 5 & 4B			S		R
Automatisation du mapping basé sur l'heuristique du bolide 1				R	A
Réalisation des tests d'évitement d'obstacles		R	S	P	A
Amélioration matérielles			P		R
[OPTIONNEL] Formulation d'une approche alternative utilisant les point clouds pour détecter les obstacles		P	P	R	A
[OPTIONNEL] Harmonisation des deux véhicules		S	S	S	R
Phase 3 - Rendu et passage de flamme	30/01/25				
Rédaction du rapport final		P	P	P	R
Réalisation de la vidéo de rendu		P	P	P	R
Rédaction et mise à jour du repo					R
Présentation aux M1		?	?	?	?
				Légende : P	Producteur
				R	Responsable
Mis à jour le :	09/01/25			A	Approuve
				V	Visa

Figure 1.3 Matrice des tâches



Fault report | RViz - lidar_frame tf failure

Author: Nicolas Hammje

Title: RViz - lidar_frame tf failure

Problem Encountered:

When setting the fixed frame to anything other than `lidar_frame`, the `tf` would bug out.

Solution Implemented:

The `time_increment` on the `raw_lidar_data` was too big. There must be an issue in `webots`, but setting `lidar_process.py` to not carry over the `time_increment` but to instead set it to 0 arbitrarily solved the issue.

Report generated on: 2025-01-08 22:31:56

Figure 1.4 Exemple de rapport rédigé

2 Architecture opérationnelle

2.1 Analyse de l’environnement

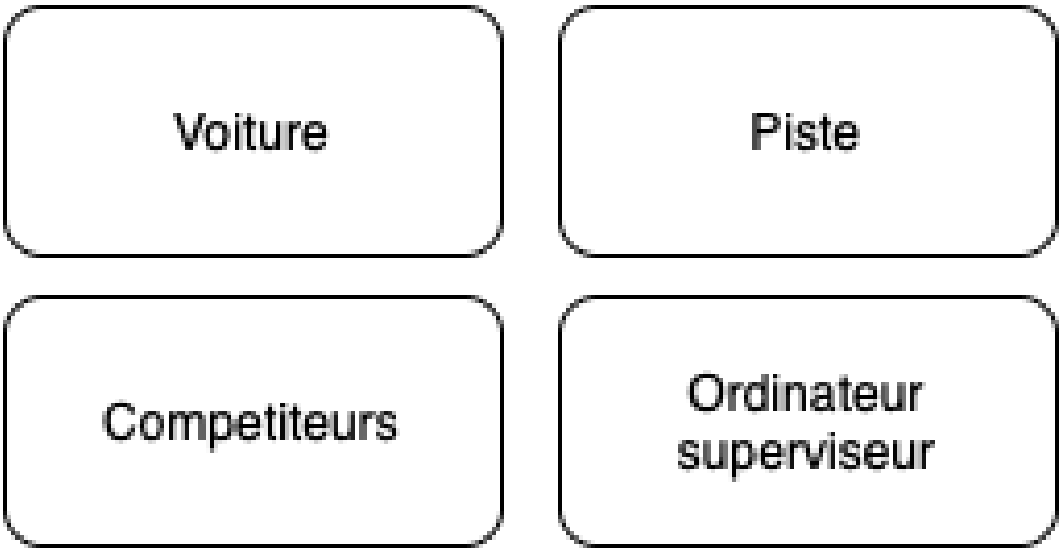


Figure 2.1 Environnement

Système	Description
Voiture	La voiture autonome, centre du projet
Piste	La piste sur laquelle la voiture roule, dans laquelle elle se repère
Competiteurs	Les autres voitures, qu’il faut doubler et éviter
Ordinateur superviseur	Ordinateur de la personne qui supervise la voiture. S’occupe de trouver la trajectoire optimale avant la course, et permet l’arrêt d’urgence de la voiture si besoin

2.2 Interfaces opérationnelles

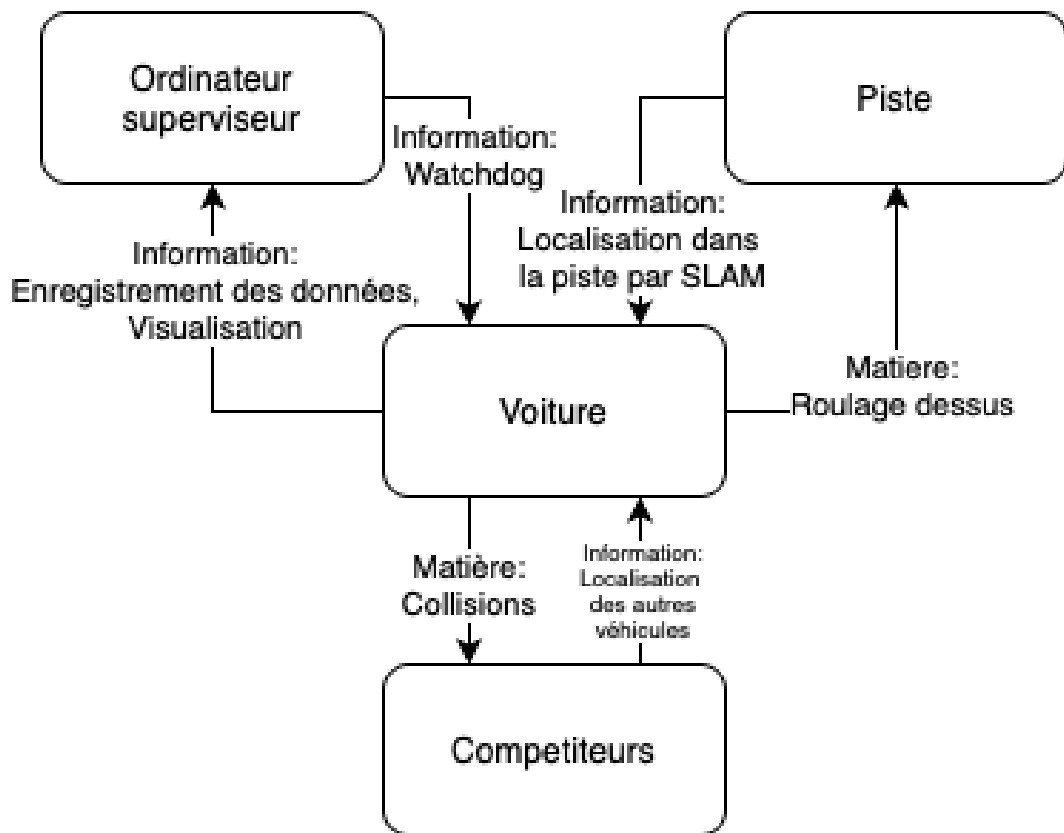


Figure 2.2 Interactions système-environnement

2.3 Analyse des besoins

L'environnement est crucial à l'opération de la voiture autonome, car la voiture utilise son LiDAR pour cartographier puis se repérer dans la piste, ce qui lui permet de suivre la trajectoire optimale calculée préalablement par l'ordinateur superviseur. Cette partie a déjà été effectuée l'année dernière, mais la détection et l'évitement des obstacles n'étaient pas assez abouti pour être utilisés.

L'enjeu de cette année est donc de détecter les autres voitures, qui sont des obstacles dynamiques, en un temps très court pour permettre de formuler, résoudre, puis effectuer une manœuvre d'évitement qui combine les informations de la piste, de la voiture, et des compétiteurs.

Environnement	Service	Capacité	Critère	Contexte
Voiture	-Localiser dans la piste -Rouler sur la piste	SLAM, roues	Temps de localisation, et précision.	La voiture doit se localiser sur la piste et y rouler
Ordinateur Superviseur	-Pouvoir arrêter la voiture si besoin	Noeud ROS	Temps d'exécution	La voiture doit pouvoir être arrêtée
Compétiteurs	-Pouvoir être évités	Détectés par le LiDAR et la caméra, la voiture les évite.	Non-collision	Les autres voitures sont sur la même piste lors de la course.

2.4 Cycle de vie et scenarios opérationnels

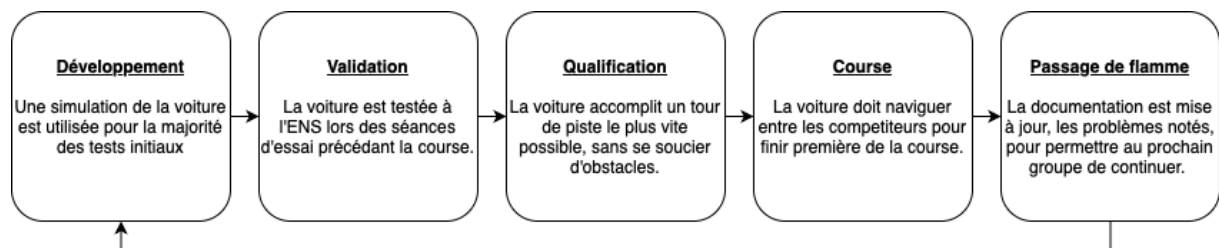


Figure 2.3 Cycle de vie de la voiture

Cartographie de la piste

L'utilisateur fait faire un tour de piste à la voiture pour construire une carte de celle-ci. Cette carte est ensuite transférée à l'ordinateur superviseur.

Optimisation de la trajectoire

L'ordinateur superviseur résout le problème d'optimisation du temps de tour pour trouver la trajectoire optimale à emprunter. Cette trajectoire est ensuite uploadée sur l'ordinateur de la voiture

Qualification

Pendant la phase de qualification, la voiture est seule sur la piste, et n'a donc pas besoin d'éviter les autres voitures. Il s'agit simplement de faire un tour, le plus vite possible.

Course

Pendant la course, la présence d'autres voitures ajoute la nécessité de les détecter et de les éviter sous peine de perdre du temps ou de se retrouver immobilisé.

Après course

Il est important de tout mettre en place et de tout documenter pour permettre aux groupes des années suivantes de continuer le travail de manière efficace.

3 Architecture fonctionnelle

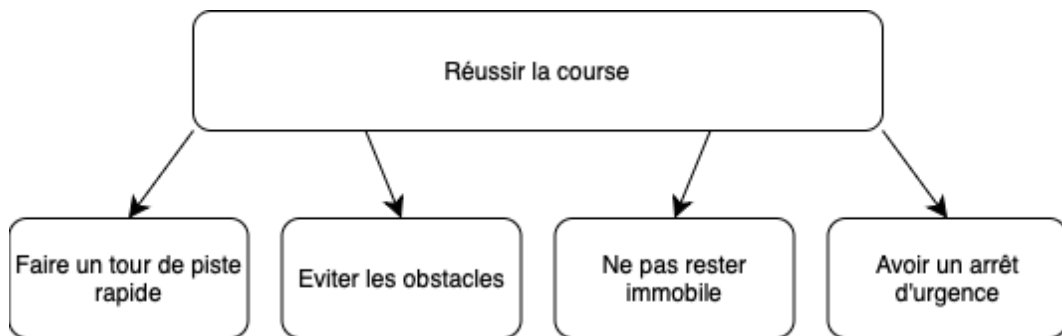


Figure 3.1 Architecture fonctionnelle

3.1 Exigences fonctionnelles

Nom	Critère	Contexte
Faire un tour de circuit rapide	Temps de tour	Pour bien se positionner, la voiture doit faire des qualifications rapides.
Éviter les obstacles	Temps perdu lors de la présence d'un obstacle	La voiture doit éviter les obstacles de manière optimale
Ne jamais être à l'arrêt	N/A	Une voiture arrêtée pendant 3 secondes est disqualifiée et enlevée de la piste.
Communication constante	Temps de réponse	Un arrêt d'urgence doit pouvoir être commandé.

3.2 Développement des fonctions

3.2.1 Faire un tour de piste rapide

Pour réussir, la rapidité des tours de pistes est une des plus grandes nécessité. La voiture suit donc une trajectoire optimisée pour le circuit donné, supposé sans obstacles. Pour se localiser dans la piste, une combinaison d'odométrie et de SLAM sont utilisés. La SLAM donne la position absolue dans la piste mais est limitée en fréquence par le LiDAR. L'odométrie donne l'évolution relative de la voiture, mais n'est qu'approximative. Ainsi, L'odométrie est fusionnée avec la SLAM pour obtenir une estimation de la position à fréquence élevée.

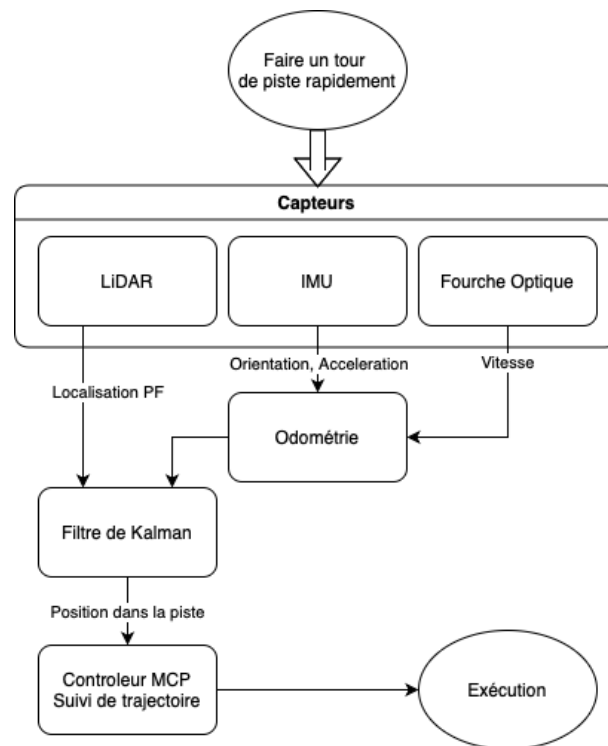


Figure 3.2 Tour de piste rapide

3.2.2 Éviter les obstacles

La deuxième exigence, et parfois la plus importante, est celle d'éviter de manière rapide les obstacles. Dans le contexte de la course, les obstacles sont les autres voitures, qui sont donc la plupart du temps également en mouvement. Pour gagner, il faut donc être en capacité d'effectuer des manoeuvres de dépassement rapides et optimales. Ayant déjà une connaissance du circuit et de la trajectoire optimale autour de celui-ci, la voiture suit cette dernière. Si un obstacle est détecté par le LiDAR et/ou la caméra, le contrôleur MPC formule une trajectoire optimale qui contourne l'obstacle tout en respectant les contraintes physiques (forme de la piste, limitations de mouvement de la voiture, etc).

3.2.3 Ne jamais être à l'arrêt

Malgré toutes les manoeuvres d'évitement d'obstacles, il arrive que la voiture se retrouve immobilisée, par exemple suite à une collision avec un compétiteur. Le règlement impose que chaque voiture immobile pendant plus que 3 secondes se voit disqualifiée et enlevée de la piste. Il est donc impératif de ne jamais se retrouver immobile. L'heuristique choisie est de forcer une marche arrière si la voiture est immobile pendant plus d'une seconde.

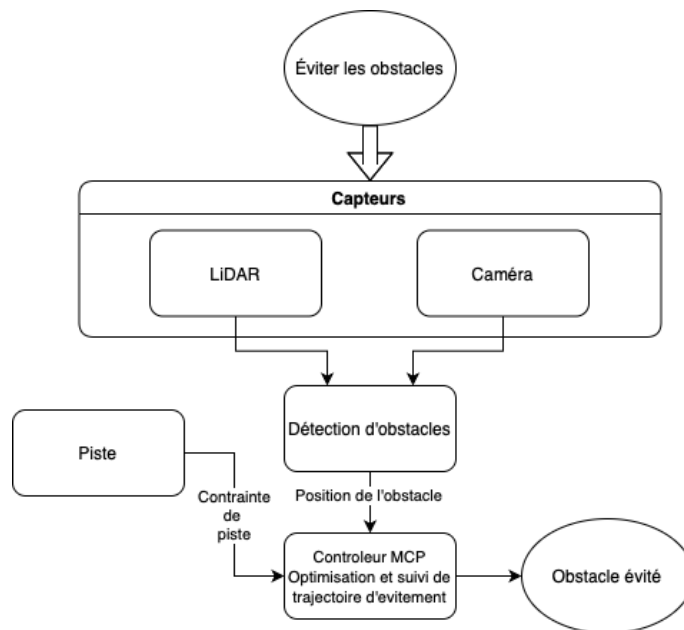


Figure 3.3 Evitement d'obstacles

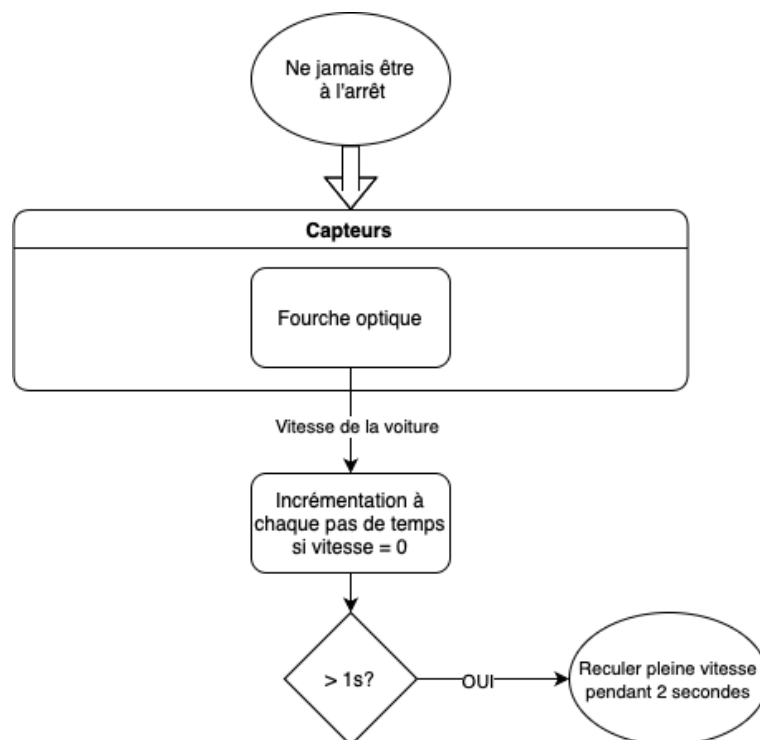


Figure 3.4 Ne pas rester immobile

3.2.4 Toujours être en communication

Également imposé par le règlement, il faut que le superviseur soit toujours en capacité de commander un arrêt d'urgence. Ainsi, si la communication entre l'ordinateur superviseur et la voiture est perdue, la voiture s'arrête automatiquement. Un délai de 2 secondes est admis pour résoudre des cas de connexion instables.

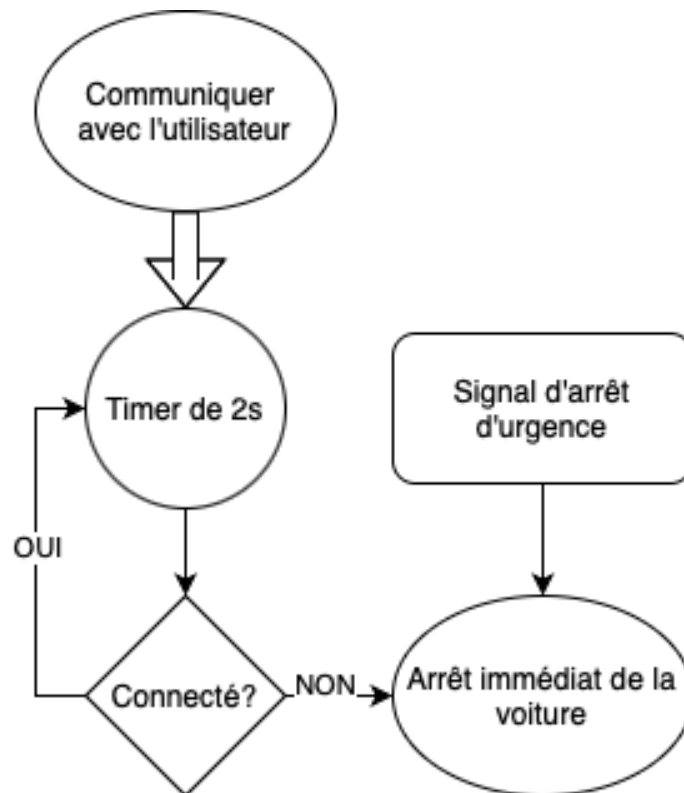


Figure 3.5 Toujours être en communication

3.3 Architecture fonctionnelle dynamique

3.4 Modes de fonctionnement

L'opération de la voiture passe par trois modes de fonctionnement distincts.

3.4.1 Cartographie de la piste

Il faut, avant la course, effectuer un tour de piste pour cartographier et enregistrer le parcours de celle-ci. Ce tour de piste est effectué sur piste vide de tout obstacle ou compétiteur.

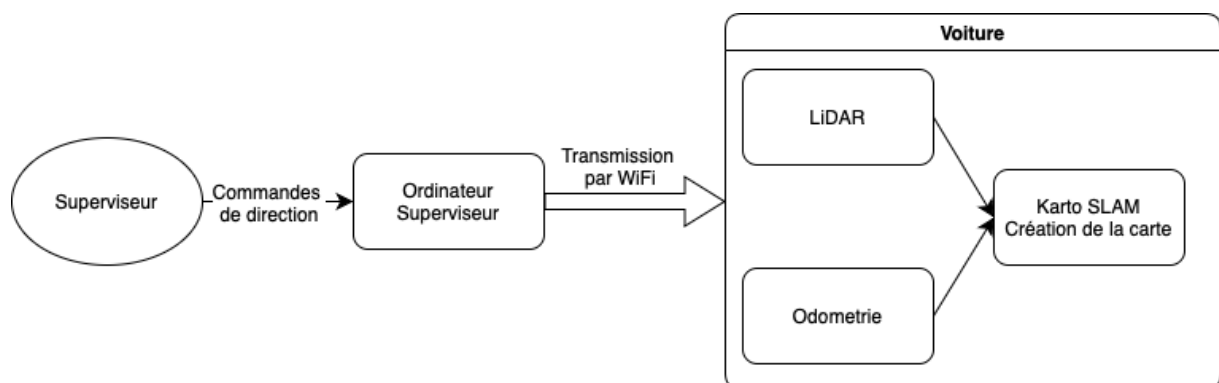


Figure 3.6 Cartographie

3.4.2 Génération de la trajectoire

Une fois la carte effectuée, une optimisation de la trajectoire est effectuée sur l'ordinateur superviseur, plus puissant que les ordinateurs embarqués sur la voiture. Cette opération prend environ 5 minutes, mais n'est effectuée qu'une seule fois.

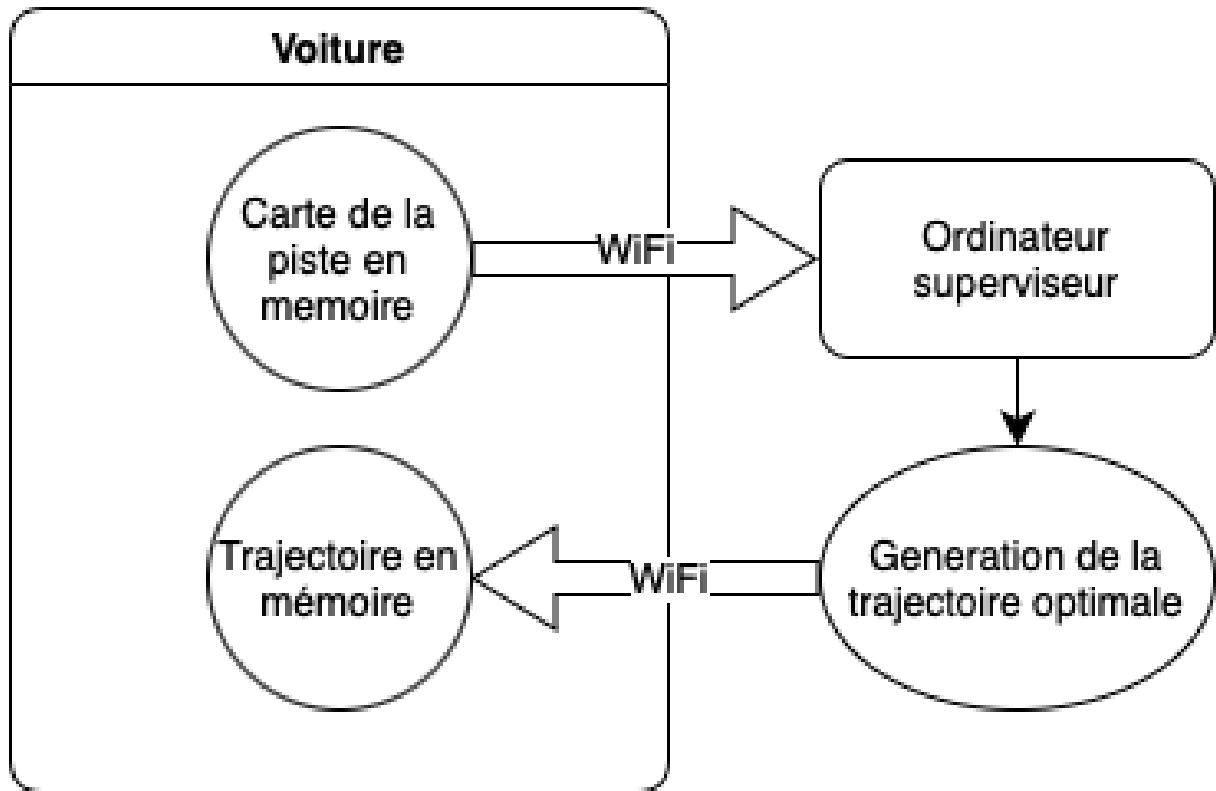


Figure 3.7 Generation de la trajectoire optimale

3.4.3 Course

Finalement, le mode de fonctionnement qui est au centre de l'enjeu de la voiture autonome, est celui de parcourir le circuit en un temps minimal. Tous les capteurs de la voiture sont utilisés pour réduire au maximum les erreurs de positionnement et trouver la meilleure trajectoire à emprunter.

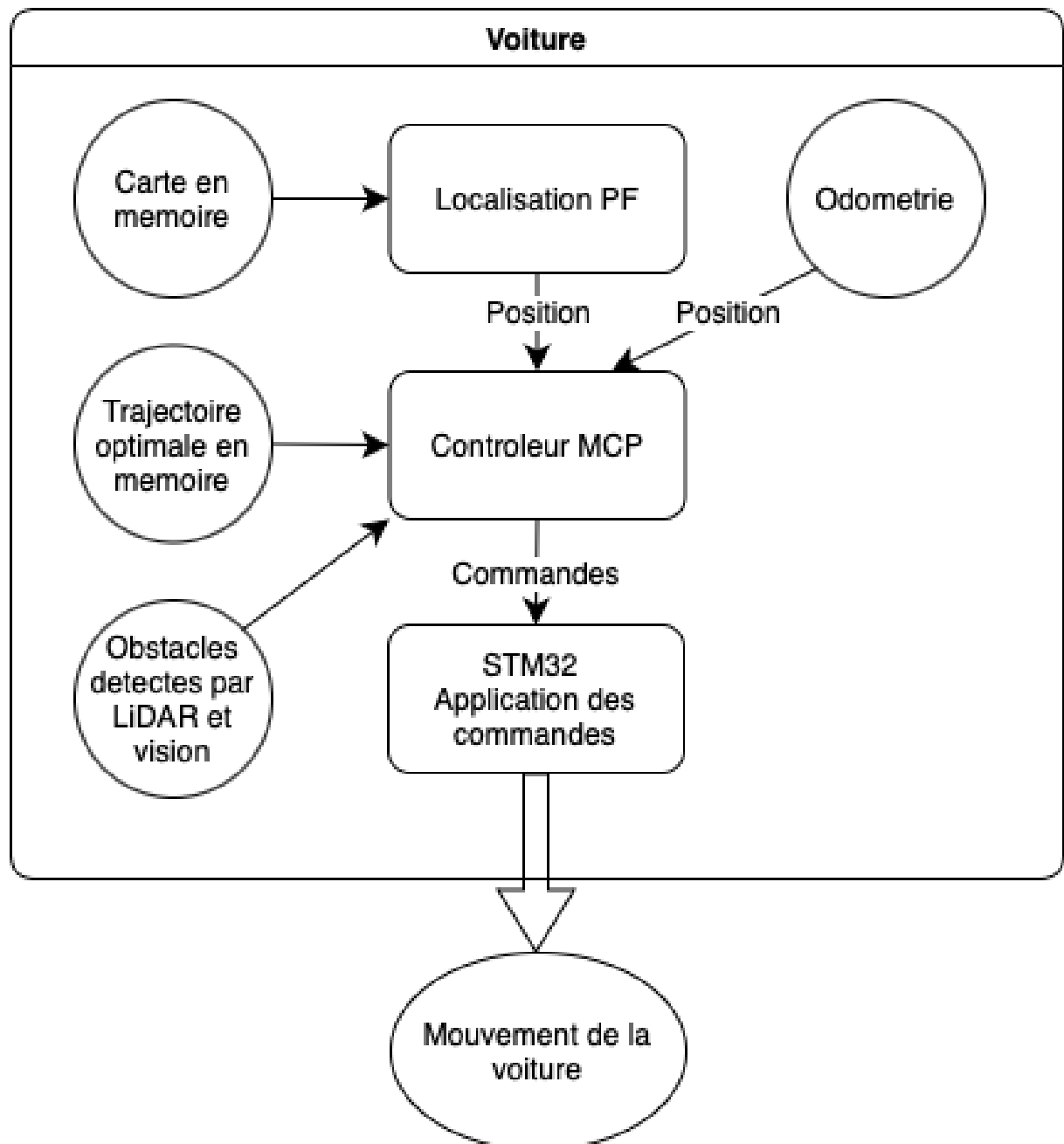


Figure 3.8 Course

4 Architecture physique

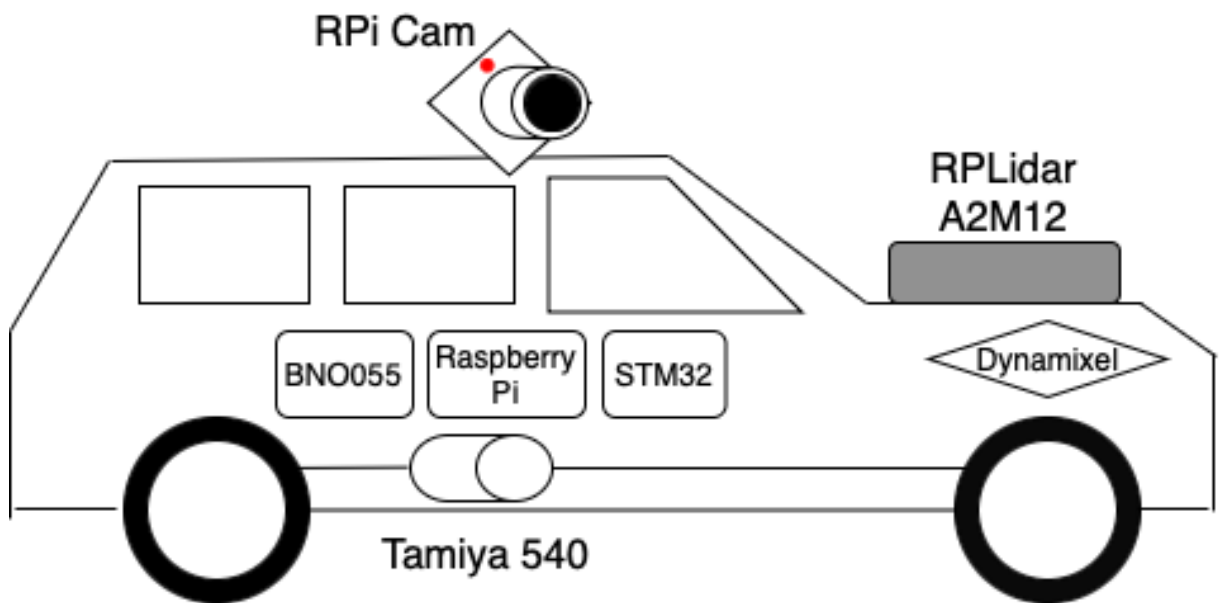


Figure 4.1 Composants principaux

Composant	Fonction	Contexte
Tamiya TT02	Chassis de la voiture	Voiture radiocommandée, base de la voiture autonome.
Tamiya 540	Moteur de propulsion	Moteur de base de la TT02. Contrôlé par un ESC.
Tamiya TBLE-04S	Electronic Speed Controller (ESC)	Convertit un signal PWM en tension fournie au moteur pour le contrôle de la vitesse.
Dynamixel AX-12A	Servomoteur	Sert à contrôler l'angle de direction de la voiture.
STM32L432KC	Microprocesseur	S'occupe des systèmes bas-niveau : récolte des données des capteurs, envoi des données au Servo et à l'ESC, etc
Raspberry Pi	Single Board Computer	S'occupe des calculs haut-niveau, comme la localisation, la formulation de commande, etc

Composant	Fonction	Contexte
RPLidar A2M12	LiDAR	Utilisé pour cartographier et se localiser dans la piste.
RPi Cam V2	Camera	Utilisée pour détecter les obstacles.
BNO055	IMU	Utilisée pour l'odométrie de la voiture.
Fourche Optique	Capteur de vitesse	Fabrication ENS Saclay, mesure la vitesse de rotation de l'arbre de transmission.
6s 5,000 mAh LiPo	Batterie	Alimente toute la voiture en 22V
Pololu D24V22F5	Step-Down 5V	Passe les 22V de la batterie en 5V pour alimenter l'électronique.

La partie « voiture qui avance » est standardisée entre les compétiteurs pour ne pas avoir de différences de vitesses causées par autre chose qu'une différence d'heuristique et d'exécution de dite heuristique.

4.1 Protocoles de communication

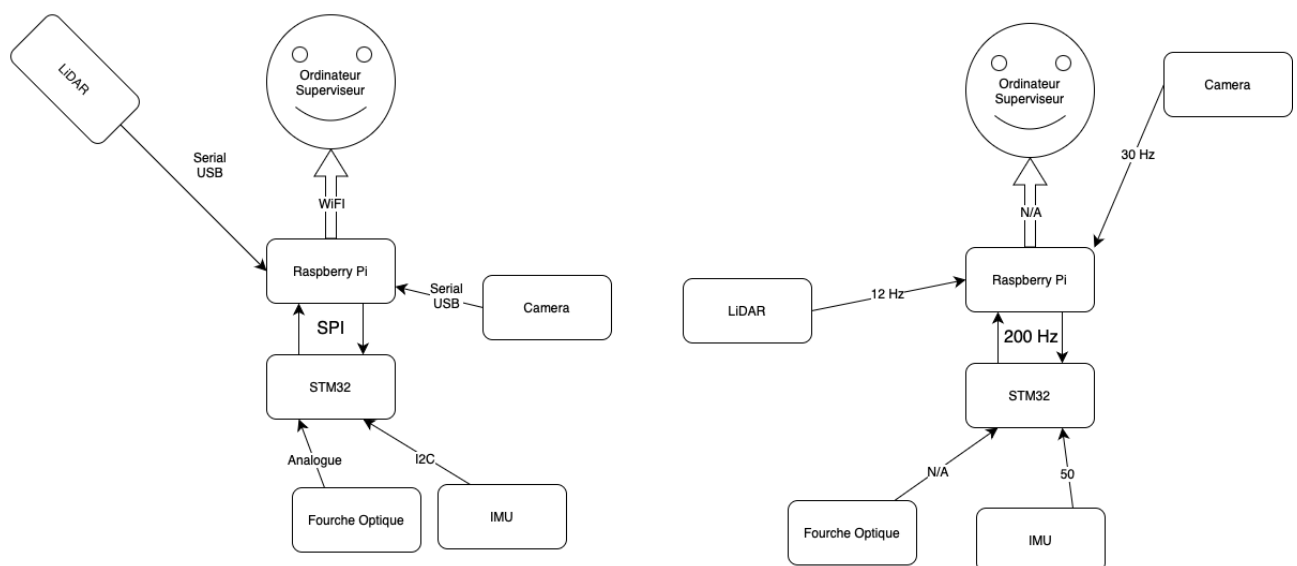


Figure 4.2 Protocoles de communications (gauche) et fréquences de communication (droite)

5 Architecture logicielle

La voiture tourne sous ROS1. L'architecture logicielle est donc composée de « noeuds », qui sont reliés ensemble par le roscore.

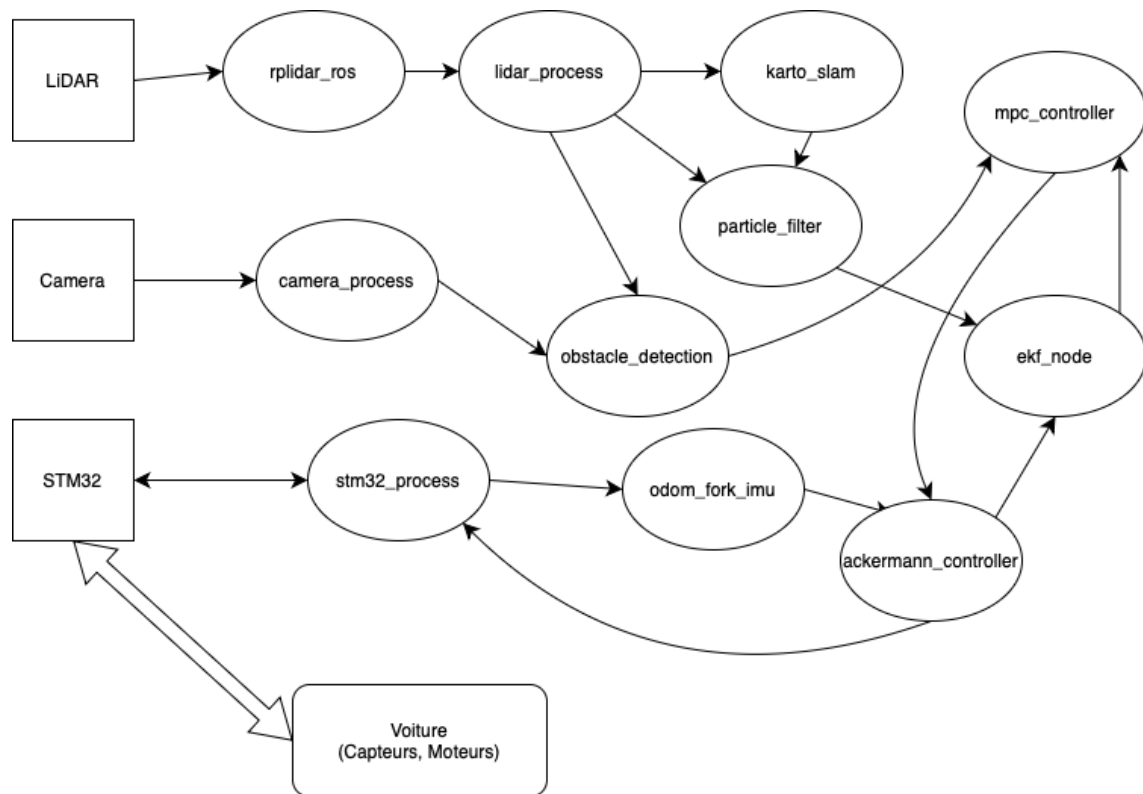


Figure 5.1 Architecture des noeuds ROS

5.1 SLAM

La voiture utilise la SLAM (Simultaneous Localization and Mapping) pour cartographier et se localiser dans la piste. Cet algorithme utilise les données de l'odométrie et du LiDAR pour créer une carte de l'environnement et s'y retrouver si la carte est déjà existante.

5.2 MPC

Pour éviter les obstacles de la manière optimale, la voiture utilise un algorithme MPC. Cet algorithme prend en compte les contraintes du système, tel que les équations de mouvement de la voiture en fonction de sa vitesse, adhérence des pneus, etc mais également la forme de la piste, l'angle de braquage maximal, la position des obstacles, pour minimiser une fonction coût. Ici, la fonction coût inclut la déviation de la trajectoire initiale, la distance

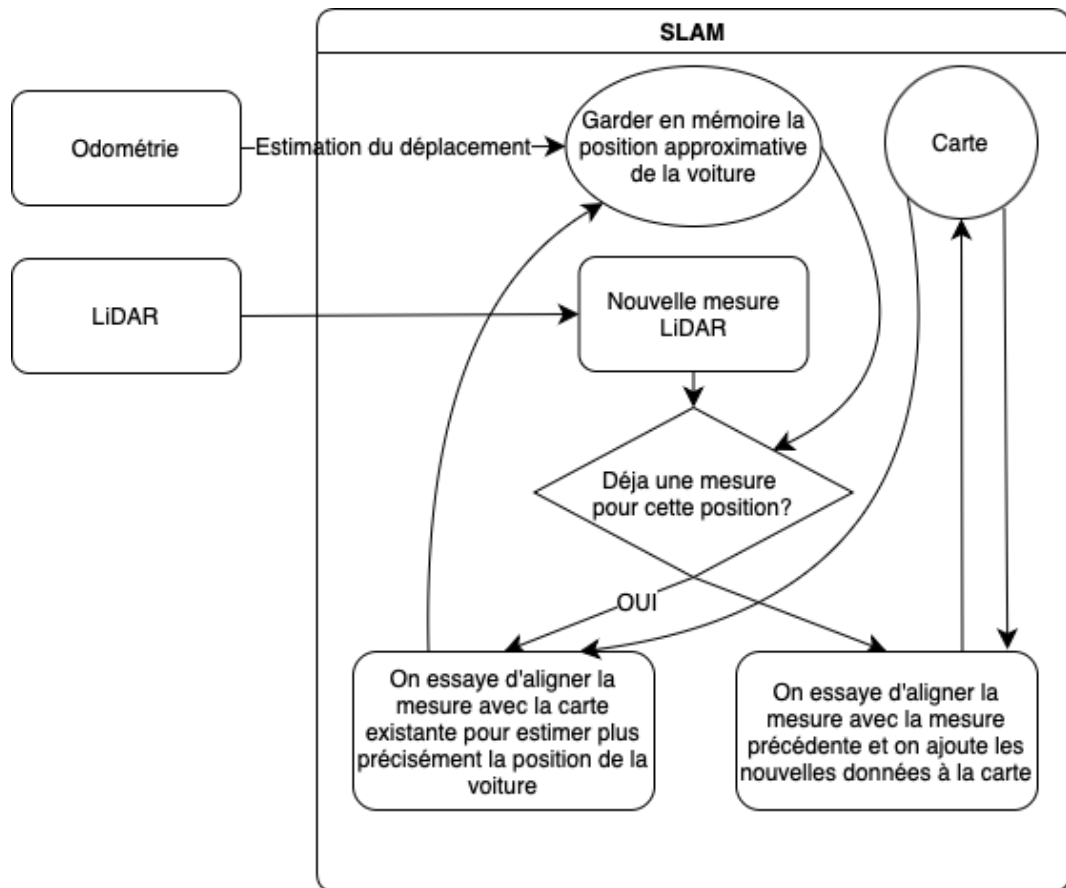


Figure 5.2 Simplification de l'algorithme réalisant de la SLAM

parcourue, et la magnitude des changements de direction. Cette combinaison de facteur sera donc réduite au maximum.

Sur la figure 5.3, les trajectoires possibles sont celles que la voiture peut physiquement effectuer (il y en a 5, émanant de la voiture). Les deux trajectoires en rouge continues sont impossible car elles ne respectent pas les contraintes de non-sortage-de-piste et/ou de non-collision-avec-obstacles. La trajectoire rouge pointillée est la trajectoire optimale. La trajectoire optimale est la verte, car elle minimise la fonction coût (arbitraire, ici).

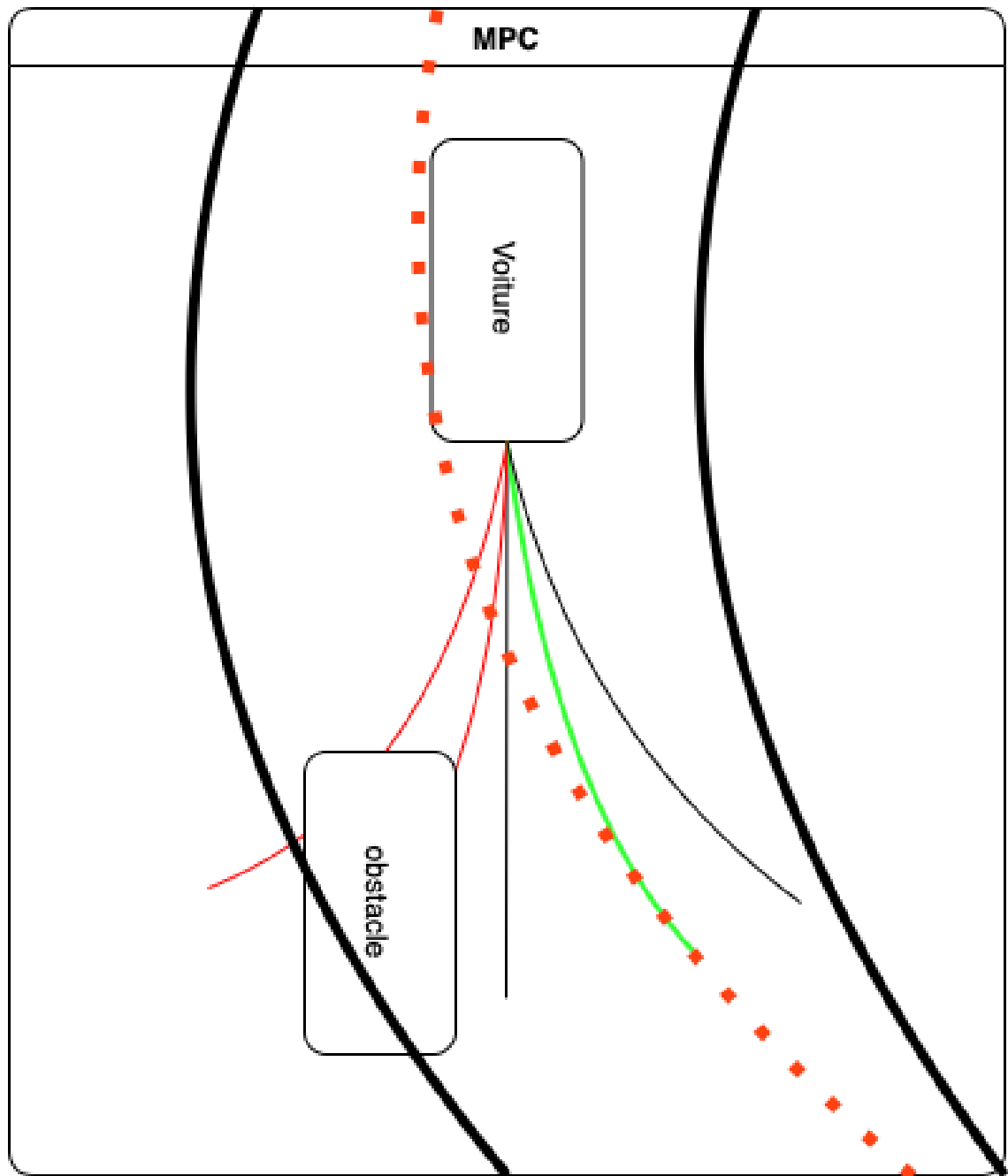


Figure 5.3 Illustration du MPC

6 Architecture organique

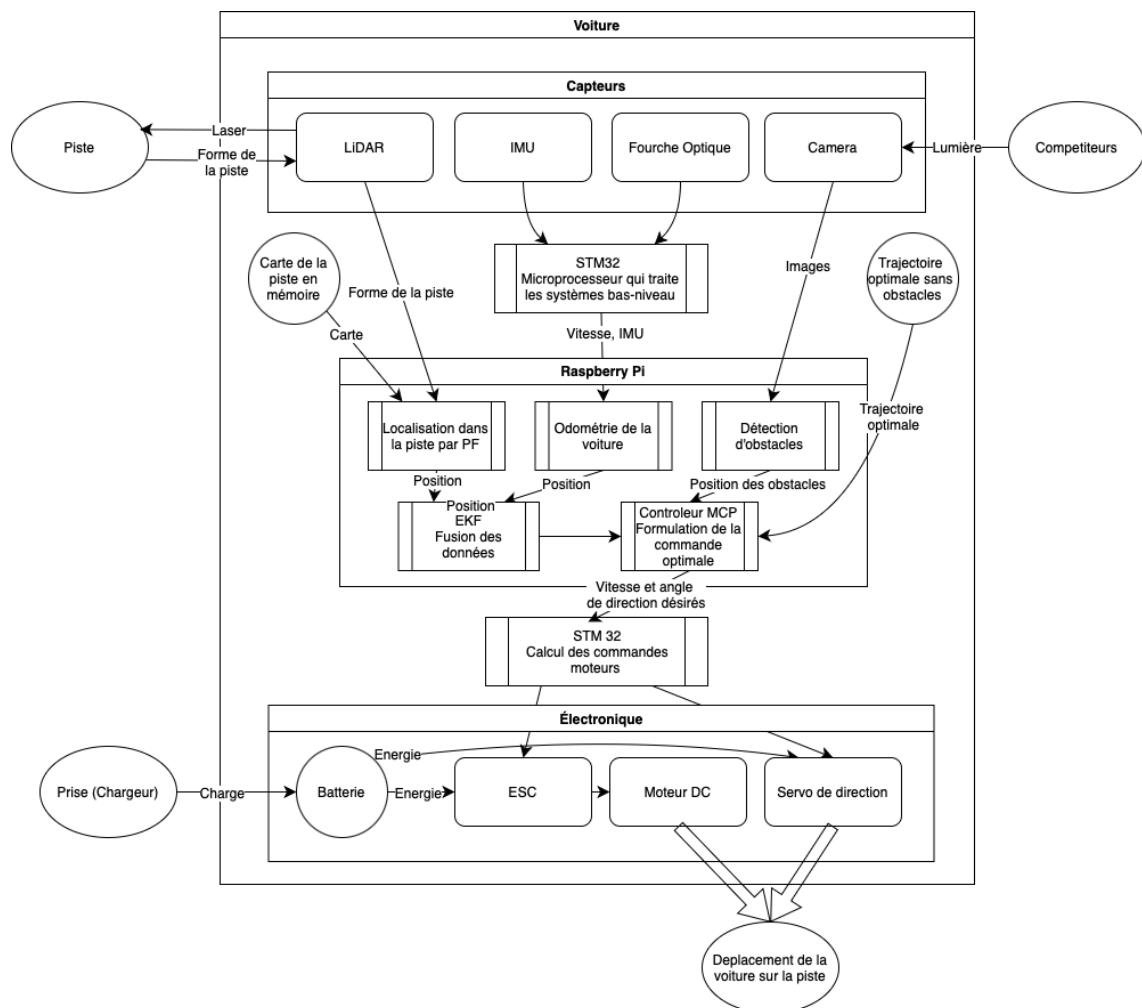


Figure 6.1 Architecture organique complète de la voiture

6.1 Organisation des composants

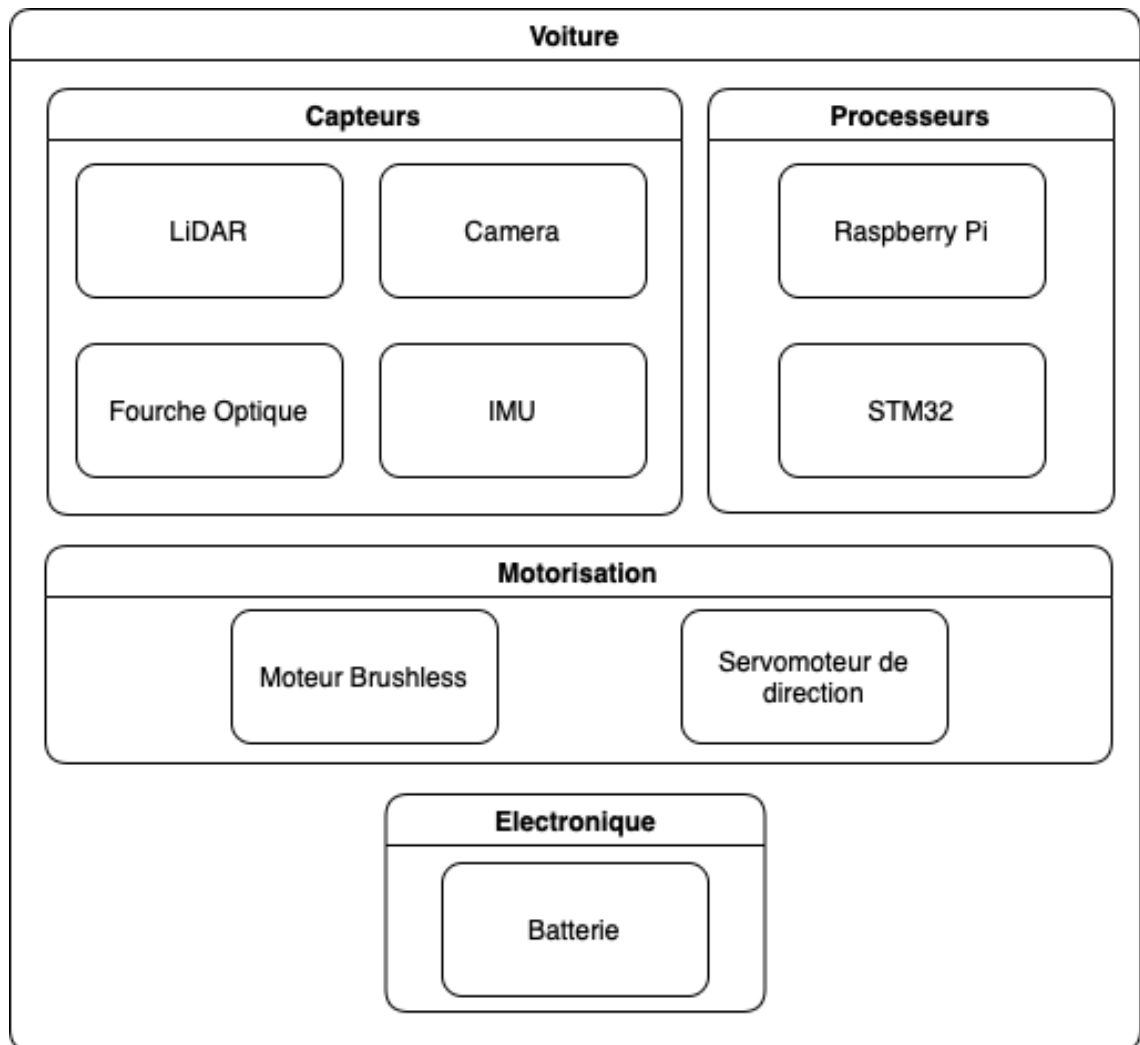


Figure 6.2 Organisation des composants

Les composants détaillés sont listés dans la section Architecture physique, ici les sous-ensembles sont l'objet du diagramme.

6.2 Flux MEI de la voiture

La voiture étant autonome, les flux matière sont inexistants, si l'on ne considère pas « rouler sur la piste » comme un flux matière. Les flux informations n'incluent pas la piste que l'on pourrait rajouter sur la base qu'elle est détectée par le LiDAR.

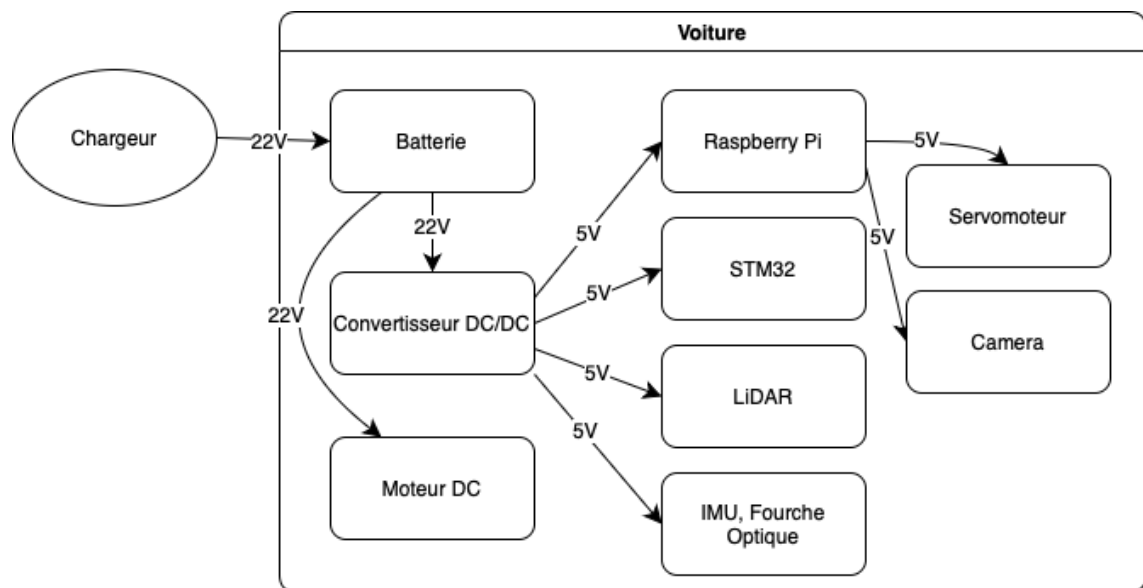


Figure 6.3 Flux énergie de la voiture

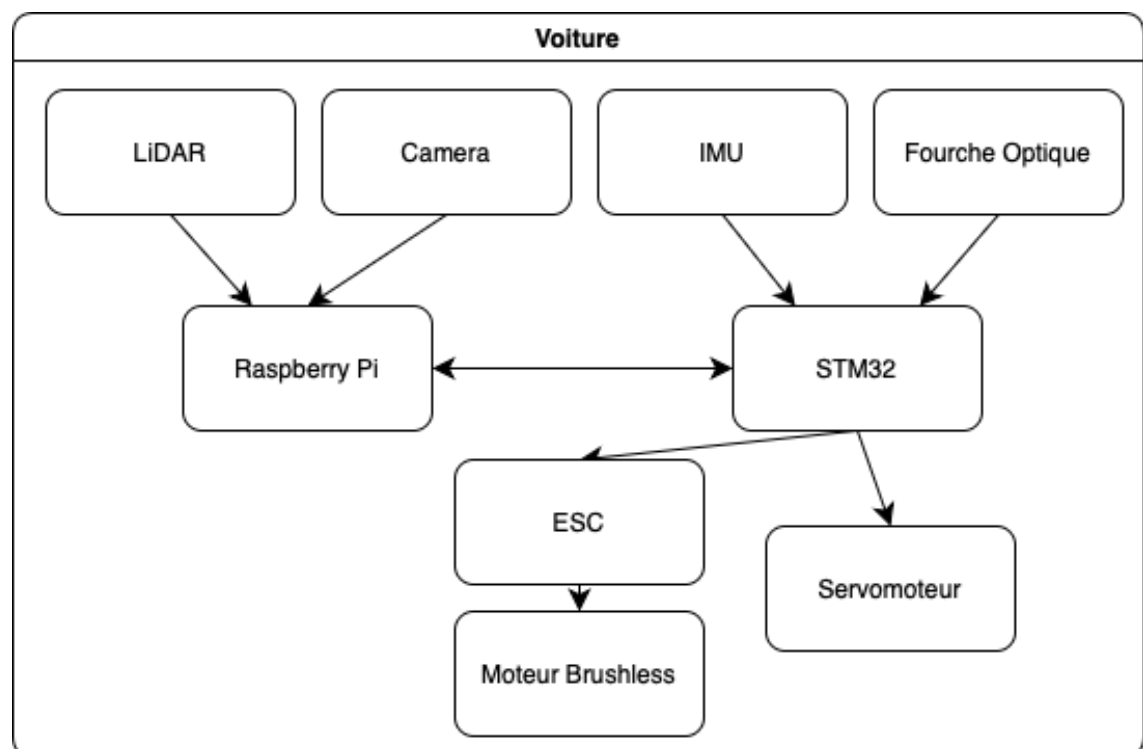


Figure 6.4 Flux information dans la voiture

7 Dimensionnement du système

7.1 Dimensions physiques

Dimension	Valeur	Contexte
Masse de la voiture	2.3kg	Poids de la voiture. Plus il est faible, plus la voiture accélère rapidement.
Taille de la voiture	0.257 x 0.19m	La taille est fixée par le règlement.
Vitesse maximale	10 m/s	La vitesse maximale est fixée par le règlement imposant un moteur et une tension maximale.
Taille de la batterie	5,000mAH	La taille maximale de la batterie est fixée par le règlement.
Autonomie de la voiture	20 tours	Plus la batterie est chargée, plus la voiture accélère rapidement.
Cout	1,200 euros	La voiture est transmise d'année en année et le budget n'est pas un facteur limitant.

Le but du projet n'est pas de modifier la voiture, et de tout de façon les règles ne permettent pas de gagner un avantage matériel sur les compétiteurs. Le but est d'optimiser l'heuristique et l'optimisation de la voiture.

7.2 Performance

L'année dernière, la voiture obtenu le 2e meilleur temps de la course. Cette année, l'ajout du MPC et de la détection d'obstacles devrait la rendre plus rapide. Cependant, il est compliqué de mesurer les temps sans faire des tours de pistes à l'ENS, ce qui n'est possible qu'en mars.

D'autre part, n'ayant pas encore finalisé les algorithmes de détection d'obstacles par vision, il nous est difficile de quantifier leur performance. Nous envisageons cependant

l'ajout d'un deuxième Raspberry Pi, dédié à la vision, pour alléger la charge du premier et/ou augmenter la performance de la détection d'obstacles. Pour l'heure, nous travaillons principalement en simulation, et n'avons pas encore de résultats meilleurs qu'en utilisant les algorithmes des années passées, si ce n'est dans le cas de figure où un obstacle bloque la trajectoire, où l'algorithme de l'année passée se retrouve donc bloqué.

7.3 Modes de dysfonctionnements

Il se peut qu'un ou plusieurs composants de la voiture ne fonctionnent pas correctement lors de la course. Des solutions de replis sont donc implémentées, pour maximiser les chances de terminer la course. On distingue une anomalie sur quelques pas de temps d'une anomalie longue durée.

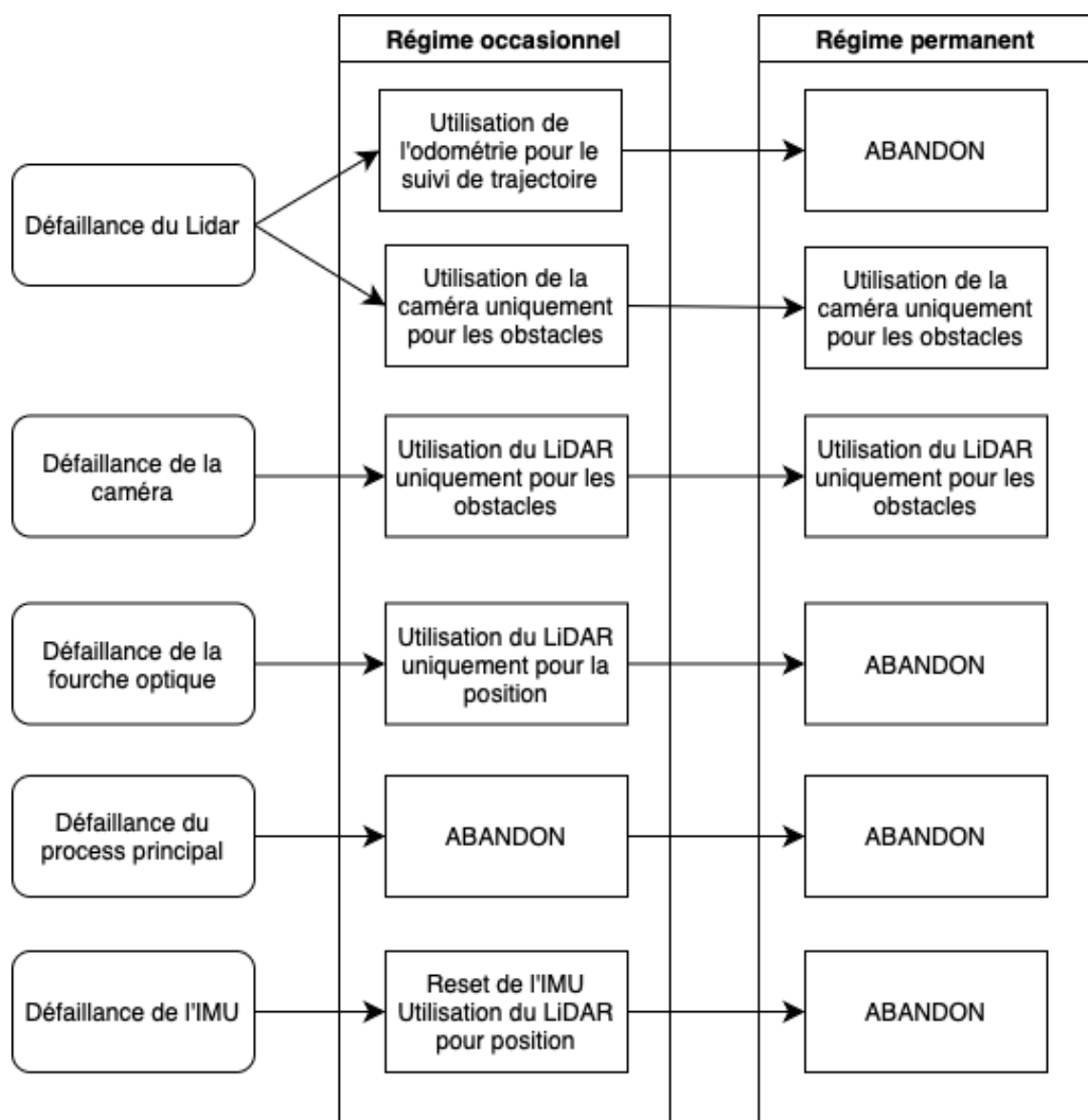


Figure 7.1 Solutions de replis

Les systèmes tels que ceux utilisés sur une voiture autonome ont la particularité de devoir émettre une commande à intervalle régulier, il n'est donc pas possible d'attendre qu'une fonction qui bugerait se termine, ou qu'une erreur de programmation fasse planter tout le système. Il faut donc prendre soin à bien traiter chaque erreur possible de manière appropriée, pour permettre à la voiture de continuer sa course ou, le cas échéant, de s'arrêter de manière « sure ». Bien que l'on soit sur des échelles réduites, les voitures peuvent quand même causer de conséquents dégâts, et il est donc primordial de considérer sérieusement les possibles anomalies et la façon dont elles sont traitées.

8 Conclusion

La voiture autonome est un système compliqué, évoluant dans un environnement hautement dynamique, utilisant un ensemble de technologies avancées pour fonctionner correctement. L'utilisation de ROS permet de faciliter l'intégration de ces différentes briques, mais cela n'économise que peu de temps dédié à l'optimisation et l'harmonisation des algorithmes sous-jacents.

La décision d'implémenter une détection des obstacles par vision, cette année, vient combler les manquements du LiDAR pour cette tâche. La performance de notre implémentation n'est pas encore quantifiable, et de nombreux tests à la fois en simulation et en vrai sont nécessaires pour correctement mesurer son efficacité.

De nombreux points peuvent encore être amélioré, incluant mais pas limité à l'amélioration de la liaison entre LiDAR et caméra, qui est encore en phase de prototypage, et l'implémentation d'une batterie de tests unitaires pour valider le bon fonctionnement de la voiture quand on modifie la structure du code.

Le travail effectué permet d'avoir une fondation solide sur lesquelles on peut facilement venir ajouter des briques de technologies encore plus poussées, permettant aussi bien à notre équipe qu'aux équipes futures de s'essayer aux nouvelles technologies d'aujourd'hui.

Au final, la course n'est qu'un objectif secondaire face à la quantité d'enseignements que l'on peut tirer d'un travail sur ce projet. La rigueur demandée quand à l'organisation n'est que fruit des projets passés, où dite rigueur n'était pas imposée, et où elle aurait été bienvenue.