



<Project logo>

Project Challenge UTAC

I4 - PAP - Project_n°74 - 05/11/2020

6 novembre 2020

Synthèse .

La mobilité de demain s'oriente de plus en plus vers l'autonomie des véhicules. Un des point technique qui reste assez complexe est le manque de fiabilité dans l'interprétation des données, issues de divers capteurs, liées à l'environnement du véhicule. Il est nécessaire d'améliorer les systèmes de fusions de données pour pouvoir analyser la scène en prenant en compte plusieurs données simultanément afin de détecter les obstacles et de planifier sa trajectoire. Le but étant d'en faire une bonne interprétation pour pouvoir modéliser parfaitement l'environnement en temps réel.

Discussions

Ce modèle de RAPPORT ne comprend pas les éventuels jalons demandés par l'entreprise partenaire ni l'axe de valorisation choisi, qui doivent être formellement définis dans le cahier des charges préparé par l'équipe

Historique des modifications

Name	Date	Changes
Pierre MOREAU	03/11/2020	Partie 2, Partie3
Franck ZHANG	03/11/2020	Partie 3
Pascal CHEN	03/11/2020	Page de garde, Partie 5
Pierre MOREAU	04/11/2020	partie 3
Franck ZHANG	04/11/2020	
Pascal CHEN	04/11/2020	Partie 3.c, Partie 4
Pascal CHEN	08/12/2020	Traitement de méthode de tentacule

Sommaire

Historique des modifications	2
Sommaire	2
1 - Introduction	4
2 - Documentation et terminologie	4
2.1 - Document de référence	4
2.2 - Terminologie	4
2.2.1 - Termes	4
2.2.2 - Acronymes	4
3 - Description du projet	5
4 - Fonction et contribution des acteurs	12
5 - Environnement de test	12
5.1 - Environnement physique	12
5.2 - Environnement matériel	12
5.3 - Environnement logiciel	13
5.4 - Données	13
6 - Suite de tests	13
6.1 - Exigence 1	13
6.2 - Exigence 2	13
6.3 - Exigence i	14
7 - Conclusions & Perspectives	14

1 Introduction

Dans un véhicule autonome et connecté, la fusion multicapteur est une nécessité pour mener à bien les différentes tâches de perception. Le projet traitera de la détection et du suivi des obstacles. Ainsi que la remontée d'information grâce à une communication V2X.

Le projet consiste à rendre une voiture autonome. Celle-ci doit pouvoir assurer plusieurs fonctionnalités. Dans un premier temps, maintenir l'inter-distance avec le véhicule devant. Ensuite permettre une communication V2I et V2N. Adapter sa vitesse en fonction des panneaux rencontrés sur la route ou des différents ralentissements sur la route. Envoyer une alerte en cas de problème sur la route. Dans l'alerte, on doit pouvoir afficher l'obstacle, sa position et sa nature afin de pouvoir prévenir les autres usagers.

Dans le cahier des charges nous avons mis à disposition une partie énumérant tous les mots techniques utilisés dans le cahier des charges ainsi que leurs définitions situées dans le lexique. Une partie acronyme expliquant toutes les abréviations utilisées, une explication sera donnée pour chaque abréviations. Ainsi qu'une partie listant tous les documents pouvant être en relation avec le cahier des charges.

2 Documentation et terminologie

2.1 Document de référence

Document	Number	Attached?	Application
Document name	Code, number, version	Yes/No	The role of the document relative to the CDC

2.2 Terminologie

2.2.1 Termes

Clothoide	C'est une courbe plane dont la courbure en un point est proportionnelle à l'abscisse curviligne du même point.

2.2.2 Acronymes

V2I	Vehicle To Infrastructure	Des messages sont échangés entre le véhicule et l'infrastructure (en général l'équipement utilisateur UE) qui prennent en charge ce type de communication.
V2N	Vehicle To Network	Les messages sont échangés via EPS (Evolved Packet System) entre l'équipement utilisateur (UE) et le serveur d'applications qui prennent en charge la communication V2N13.

V2X	Vehicle To Everything	Des messages sont échangés entre le véhicule et n'importe quelles autres plateformes (infrastructure, réseaux, piétons, ..)
SAE	Society of Automotive Engineers	
N H T S A	(National Highway Traffic Safety Administration)	

3 Description du projet

A) État de l'art /Origine du projet

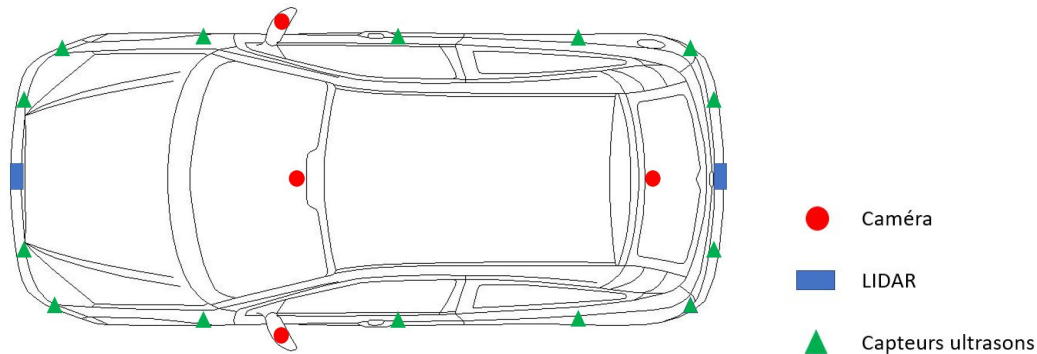
De nombreuses solutions répondent aux enjeux relatifs à la sécurité routière. C'est un domaine qui demande beaucoup d'intérêt dans la mesure où il s'agit de la vie d'hommes qui est en jeu. Aujourd'hui, de nombreuses entreprises se hâtent à développer un véhicule autonome et connecté:

- Tout d'abord nous pouvons parler de Waymo, qui est la voiture autonome de Google, la plus aboutie actuellement sur le marché international. Le système de pilotage automatique via une Intelligence artificielle utilise un LIDAR, des radars, une caméra 360°, un récepteur GPS ainsi que des capteurs sur les roues motrices. C'est une voiture qui a une autonomie de niveau 3. Les limites du projet sont tout d'abord que le véhicule est dans l'incapacité d'interpréter les signes d'un agent de la route, de la police.
- La France aussi a vu le développement de voitures autonomes sur son territoire tel que la voiture Drive 4U de Valeo qui est une voiture sans conducteur capable de rouler dans n'importe quelle ville. Le véhicule est équipé exclusivement de capteurs Valeo, comme le Radar MB79 et le ScaLa. De nombreux tests ont été fait dans la ville de Paris afin de mettre en situation ce véhicule. Néanmoins ce véhicule n'intègre pas la communication V2X dans le cadre de la signalisation d'obstacles ou de changements dans l'environnement du véhicule

.A travers ces solutions nous pourrions améliorer certains protocoles comme les aspects de la communication V2X présentés au-dessus et qui ne sont pas présents chez la concurrence. Ou encore les protocoles de détection d'obstacles et de planification de trajectoire en améliorant par exemple l'algorithme de nos solutions en portant un accent sur les librairies en libre service de ROS (= Robot Operating System).Par ailleurs, nous pouvons remarquer que l'utilisation de capteurs LIDAR et de caméras comme la solution de Google est la meilleure piste à tenir pour le développement de notre solution. A la différence de Valeo, nous utiliserons des capteurs de fabricants différents afin de pouvoir viser le produit qui propose un rapport qualité-prix qui nous intéresse.

B) Description du produit/solution

La solution sera sous forme d'un prototype de véhicule autonome composé de capteurs



Une voiture autonome est constituée de plusieurs systèmes embarqués comme des caméras, des capteurs ultrasons ou encore des LIDARs.

Pour qu'une voiture soit dite autonome il faut un certain nombre de capteurs, caméras ou encore lidar.

Pour cela il lui faudrait ainsi :

- 4 caméras
- 14 capteurs ultrasons
- 2 LIDARs

Les radars :

Un radar émet une onde afin de pouvoir mesurer la distance qu'il y a entre la voiture et l'objet. Le capteur ainsi que l'émetteur sont situés à côté l'un de l'autre. L'émetteur envoie une onde qui est réfléchiée sur un obstacle et captée par la suite par le capteur. Le temps qu'a parcouru l'onde est par la suite calculé afin de déterminer la distance qui sépare l'obstacle des radars. Il permet ainsi d'alerter le conducteur et ainsi freiner ou arrêter le véhicule en cas de potentiel contact avec l'obstacle.

Les caméras :

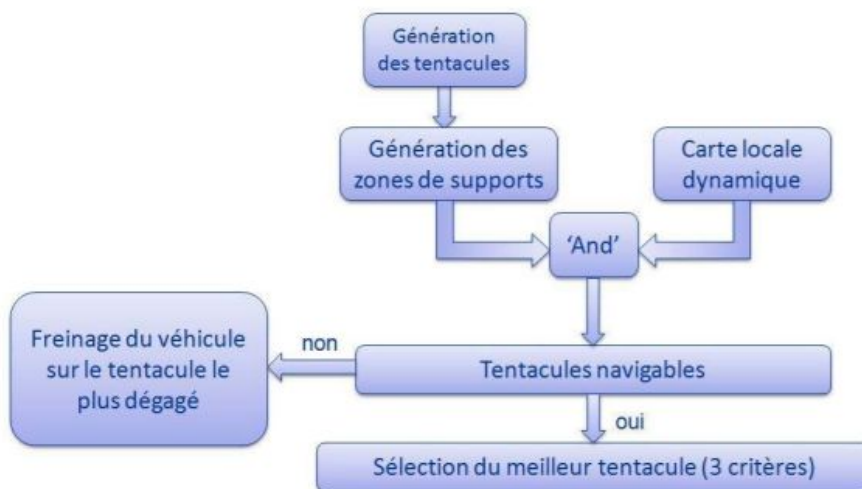
Toutes les voitures de nos jours sont pour la plupart équipées de caméra situées juste derrière le rétroviseur intérieur de la voiture. Cet appareil est constitué de deux objectifs afin de filmer la route en relief comme la vision humaine. Celui-ci prend énormément d'images par seconde (En France, la norme est de 25 images par seconde). Cette technologie permet une détection des obstacles mais pas seulement, il aide à se garer, détecter les routes ou encore les panneaux de signalisations ou les feux tricolores.

Le LIDAR :

Ce capteur se situe souvent dans la calandre du véhicule afin d'avoir une vision sur la route. C'est le même principe que le radar mais il utilise aussi les ondes électromagnétiques. Il peut mesurer la distance qu'avec un objet à la fois. Dans le domaine du véhicule autonome, il est utilisé pour pouvoir voir l'environnement où se situe la voiture. Il permet de détecter les obstacles, les informations que le LIDAR peut récupérer, peut-être ensuite envoyé sur un ordinateur de bord et ainsi peut influencer les trajets de la voiture.

C) Caractère innovant de la technologie/ solution

L'utilisation de modèle mathématique de planification de trajectoire tel que la **méthode des tentacules**. Le principe de cette méthode consiste à utiliser un ensemble d'antennes virtuelles appelées « tentacules » dans une grille d'occupation égo-centrée liée au véhicule, et qui représente l'environnement proche du véhicule avec la position des obstacles. Une fois les trajectoires virtuelles créées, une évaluation de ces dernières est effectuée afin de choisir la meilleure selon un critère défini. Le meilleur tentacule ou trajectoire de référence, sera exécuté par le véhicule à chaque pas de calcul. Cette méthode présente l'avantage d'être très réactive, permettant d'éviter des collisions et de circuler dans un environnement incertain sans besoin d'une grande quantité d'informations a priori. En outre, elle n'accumule pas les données et utilise une carte locale égo-centrée, évitant ainsi le problème du SLAM (Simultaneous Localization and Mapping). Par ailleurs, la méthode des tentacules est l'une des plus simples qui permet au véhicule de rouler en sécurité dans un environnement inconnu. Elle est aussi la plus rapide car il n'y a pas accumulation des données. En outre, elle permet une circulation sûre tout en étant capable de traverser des passages étroits. Nous la décrirons plus en détail dans le paragraphe suivant.



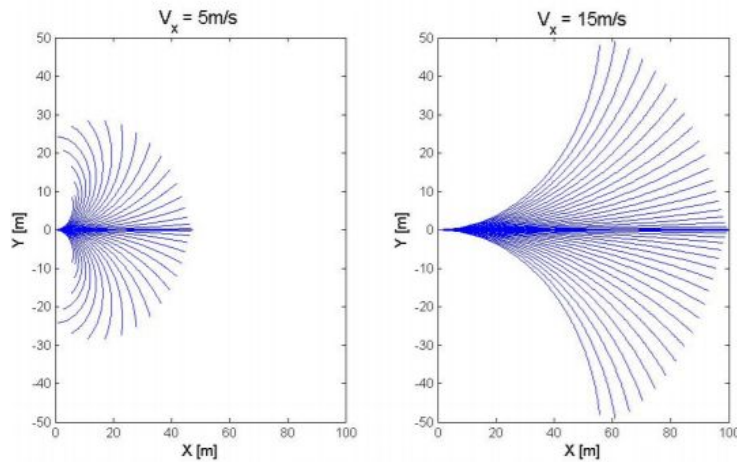
A chaque pas de calcul (toutes les 100ms), des tentacules en forme d'arcs de cercle sont générés. Une zone de support de chaque tentacule est générée permettant de s'assurer que le véhicule puisse circuler sur ce dernier en toute sécurité (sans risque de collision). Une fois les zones de supports générées, chaque tentacule est évaluée à l'aide d'une carte locale dynamique afin d'éliminer tout tentacule non navigable. L'ensemble des tentacules dont la longueur, au premier obstacle, est supérieure à la distance de collision est appelé « tentacules navigables ». Après cette évaluation, si plusieurs tentacules sont navigables, le meilleur tentacule est choisi suivant trois critères :

- le dégagement du tentacule (distance au premier obstacle) ;
- le changement de braquage ;
- et la trajectoire globale .

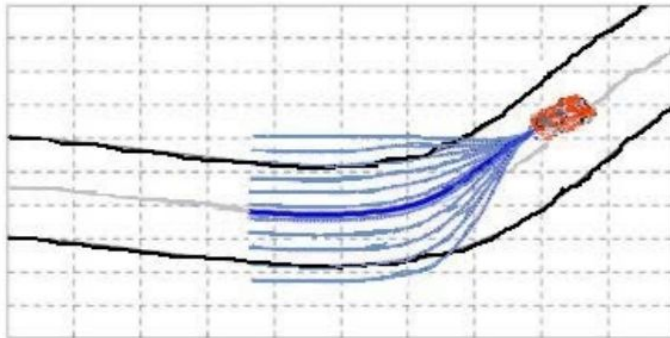
La navigation par cette méthode peut se décomposer en trois étapes : la génération des tentacules, la détermination des tentacules navigables et le choix du meilleur tentacule.

Toutes les tentacules sont représentées dans le système de coordonnées locales du véhicule. Ils commencent au centre de gravité du véhicule et ont la forme d'arcs de cercle. Dans la littérature plus récente, les tentacules sont générés suivant plusieurs formes : nous pouvons citer les tentacules parallèles à la route et les clothoïdes.

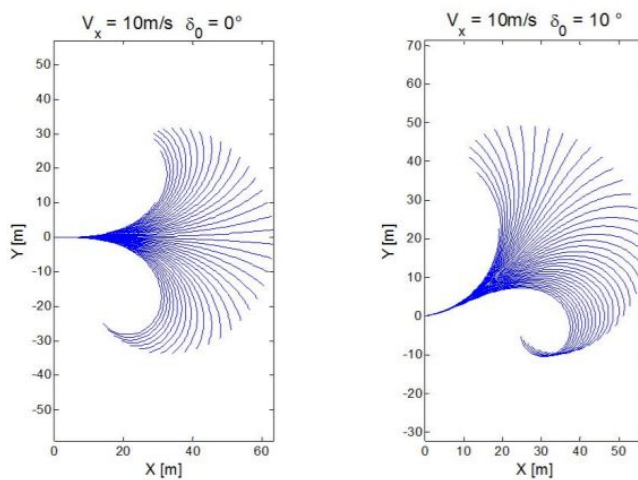
Tentacule en forme d'arc de cercle :



Tentacule parallèle à la route :



Tentacule en forme de clothoïdes :



Une fois les tentacules générés, une zone de support de chaque tentacule est déterminée. La zone de support est une zone autour du tentacule prenant en compte la longueur du véhicule et une distance de sécurité permettant de vérifier si le véhicule peut circuler sur le tentacule sans risque de collision. Elle est caractérisée par la distance de la zone de support. Après génération des zones de support, chaque tentacule est évalué en effectuant une correspondance entre la zone de support de ce dernier et la carte locale (obtenue par la fusion des informations de la perception et la localisation). Cette évolution permet de vérifier si le tentacule est

navigable ou non. Si le tentacule peut être parcouru jusqu'à la distance de collision, il est qualifié de « tentacule navigable ».

Pour décider du "meilleur" tentacule, toutes les tentacules navigables sont évaluées. Dans l'algorithme de base, trois valeurs sont calculées et combinées de manière linéaire pour choisir le tentacule optimal [Hundelshausen et al., 2008] :

- le premier critère V_{libre} permet d'évaluer si le tentacule est libre d'obstacle ;
- Le deuxième critère V_{lisse} permet d'évaluer si le braquage est lisse (ou brusque) pour suivre le tentacule. Ce paramètre se calcule à partir du braquage courant ;
- Le troisième critère $V_{trajectoire}$ permet d'évaluer si le tentacule se rapproche de la trajectoire globale.

Ces trois critères permettent ainsi d'éviter les obstacles, d'avoir une conduite confortable tout en suivant la trajectoire globale. Une fois ces critères estimés, ils sont combinés avec pondération de la manière suivante :

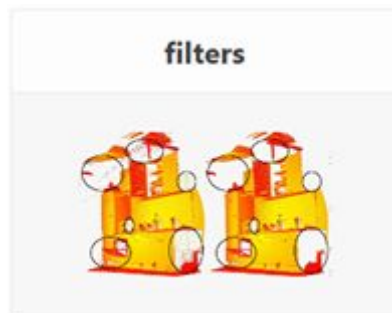
$$V_{tentacule} = a_1 * V_{libre} + a_2 * V_{lisse} + a_3 * V_{trajectoire}$$

où a_1 , a_2 et a_3 sont des constantes. La combinaison de ces critères fait apparaître le problème de pondération qui devient très difficile et délicat car dépendant du contexte de conduite. Les coefficients de pondération sont nécessairement modifiés selon le contexte.

D) Mise en place du projet dans un environnement de simulation

Dans le cadre de notre projet, nous allons mettre en place à travers ROS un environnement de simulation dans lequel nous ferons évoluer le robot HUSARION. Ce dernier sera équipé des mêmes composants que la voiture présentée plus haut mais en nombre réduit ; de plus il sera aussi capable de planifier la trajectoire et éviter les obstacles. Ci-dessous nous allons vous présenter les aspects du robot HUSARION relatif à la mise en place de notre projet ECar :

- Récupération de la data des capteurs présent HUSARION (lidar, capteurs ultrasons, caméras) :



La bibliothèque `pcl_filters` contient des mécanismes de suppression des valeurs aberrantes et du bruit pour les applications de filtrage des données des nuages de points 3D.

Un exemple de suppression du bruit est présenté dans la figure ci-dessous. En raison d'erreurs de mesure, certains ensembles de données présentent un grand nombre de points d'ombre. Cela complique l'estimation des caractéristiques 3D des nuages de points locaux. Certaines de ces valeurs aberrantes peuvent être filtrées en effectuant une analyse statistique sur le voisinage de chaque point, et en ajustant celles qui ne répondent pas à certains critères. La mise en œuvre de l'élimination des aberrations dans le PCL est basée sur le calcul de la distribution des distances entre points et voisins dans l'ensemble de données d'entrée. Pour chaque point, la distance moyenne entre le point et tous ses voisins est calculée. En supposant que la distribution résultante est gaussienne avec une moyenne et une norme.



La bibliothèque `pcl_keypoints` contient des implémentations de deux algorithmes de détection de points clés dans les nuages de points. Les points clés (également appelés points d'intérêt) sont des points dans une image ou un nuage de points qui sont stables, distinctifs, et peuvent être identifiés en utilisant un critère de détection bien défini. En général, le nombre de points d'intérêt dans un nuage de points sera beaucoup plus petit que le nombre total de points dans le nuage, et lorsqu'ils sont utilisés en combinaison avec des descripteurs de caractéristiques locales à chaque point clé, les points clés et les descripteurs peuvent être utilisés pour former une représentation compacte - mais descriptive - des données originales.



La bibliothèque `pcl_kdtree` fournit la structure de données kd-tree, en utilisant FLANN, qui permet des recherches rapides sur le plus proche voisin.

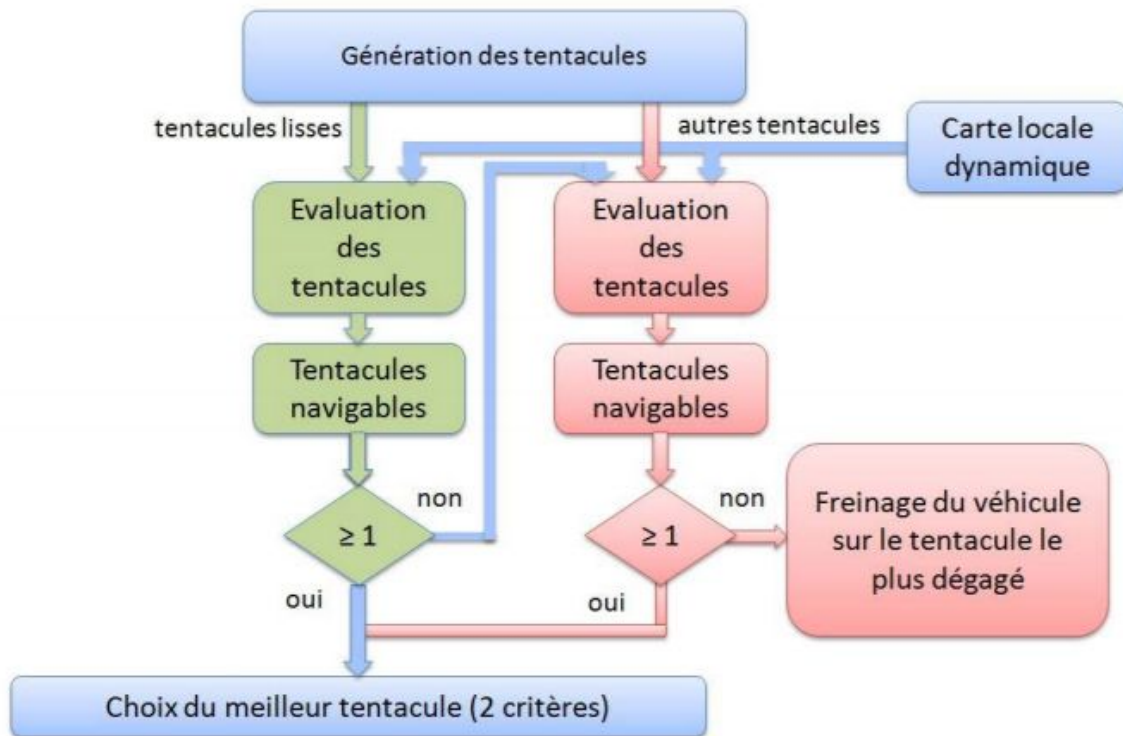
Un arbre Kd (arbre à k dimensions) est une structure de données à partitionnement spatial qui stocke un ensemble de points à k dimensions dans une structure arborescente qui permet des recherches efficaces de distance et de plus proche voisin. La recherche du plus proche voisin est une opération essentielle lorsqu'on travaille avec des données de nuages de points et peut être utilisée pour trouver des correspondances entre des groupes de points ou des descripteurs d'éléments ou pour définir le voisinage local autour d'un ou plusieurs points.



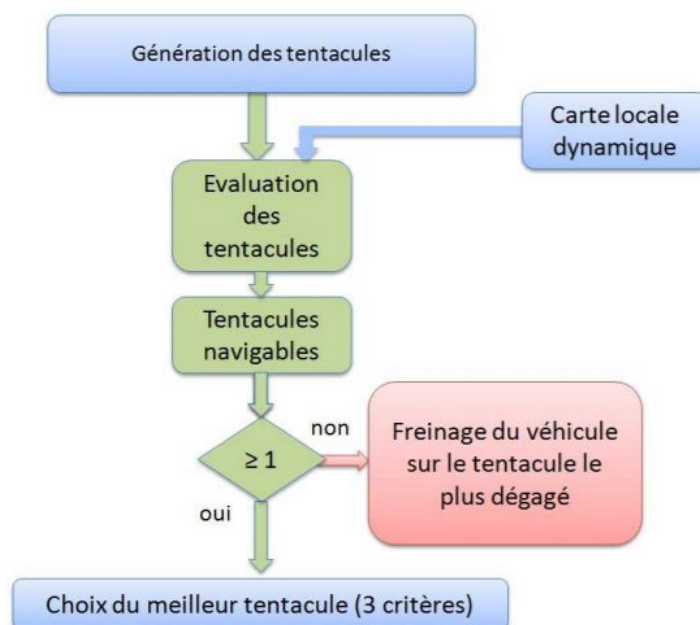
La bibliothèque `pcl_visualization` a été construite dans le but de pouvoir prototyper et visualiser rapidement les résultats d'algorithmes opérant sur des données de nuages de points 3D. La bibliothèque propose des routines `highgui` similaires à celles d'OpenCV pour l'affichage d'images 2D et le dessin de formes 2D de base à l'écran :

- Traitement de données avec ROS

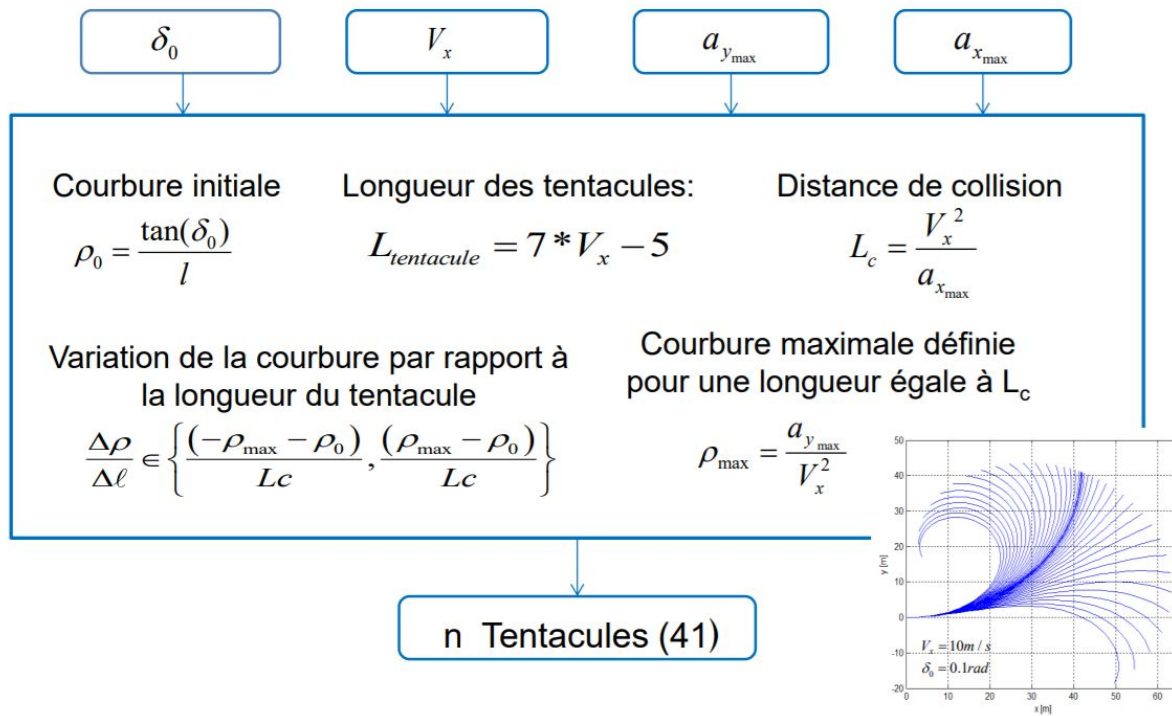
- Algorithme de la planification trajectoire de la méthode tentacule en forme d'arc de cercle :



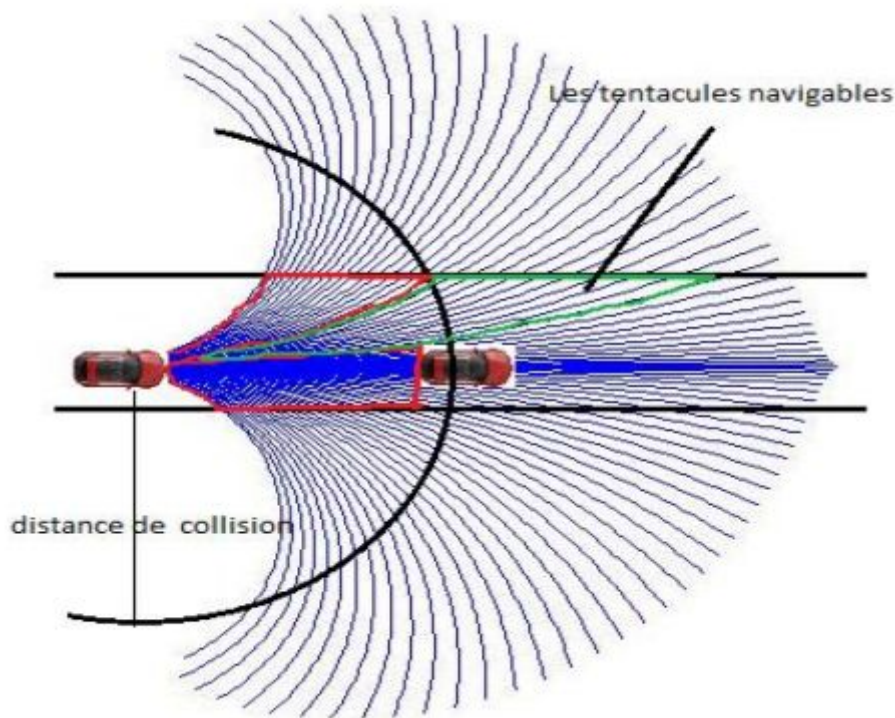
- Algorithme de la planification trajectoire de la méthode tentacule sous forme de clothoïde :



- Formules pour trouver un tentacule navigable



où δ_0 est le braquage initial, V_x est la vitesse du véhicule, $a_{y_{\max}}$ est l'accélération latérale maximale et $a_{x_{\max}}$ est la décélération longitudinale maximale

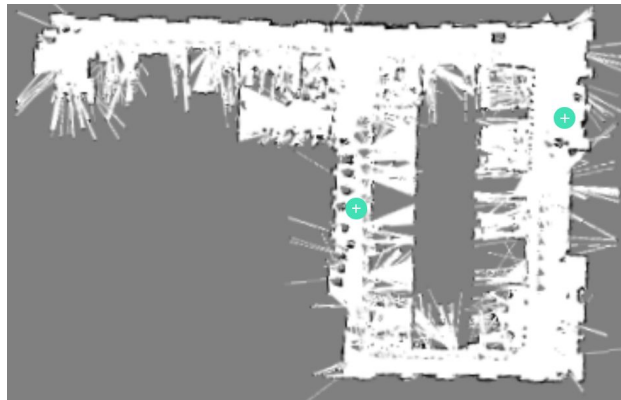


Distance de collision $L_c = \frac{V_x^2}{a_{x_{\max}}}$

- Changement de trajectoire du robot (bibliothèques ROS qui permettent les mouvements)
 - Librairies
 - Planification de mouvement : **MovIt**
 - Contrôle du robot : **KLD, iTask, SoT**
 - Boucles de perception-action
 - 1 s - 5 mn, boucle lente dans le cadre de la planification de mouvements réactifs dans des environnements complexes
 - Rosparam
 - Paramètres de stockage de données (dont les trajectoires de mouvements)
 - Elément <joint>
 - Décrit la cinématique et la dynamique d'un joint et spécifie les limites de sécurité
 - Attributs : **name & type**
 - Name spécifie un nom unique pour le joint
 - Type spécifie le type de joint (2 utiles pour les trajectoires)
 - *floating* : mouvements suivants les 6 degrés de libertés
 - *planar* : permet des mouvements dans un plan perpendiculaire à l'axe
-
- Construction la map de simulation (utilisation des bibliothèques pour construire l'environnement)

Pour la construction de l'environnement ou le robot se retrouvera deux simulateurs peuvent être utilisés. Le GMapping ou Gazebo.

GMapping :



Cette technique de cartographie a été intégrée en 2010 sur ROS. Cette méthode est la plus utilisée aujourd'hui. Cette technologie utilise l'odométrie, ce qui permet d'estimer la position du robot en mouvement.



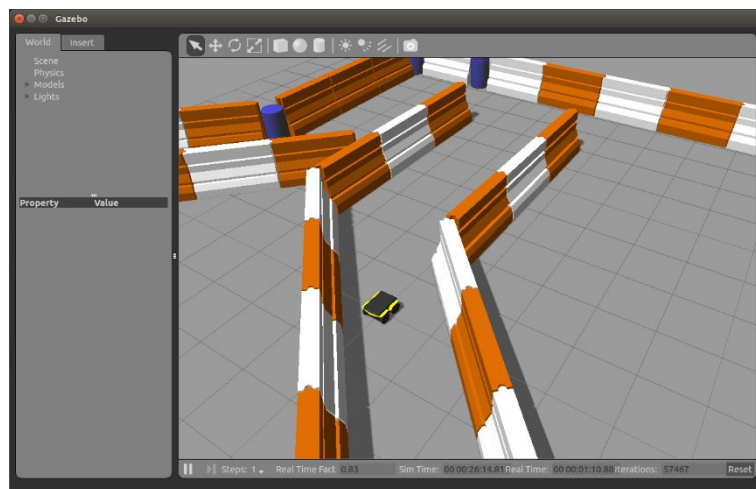
Lorsque l'on arrête le capteur LIDAR du robot ainsi que l'odométrie permettant de le localiser dans l'espace, cela engendre des erreurs, la carte devient floue ou encore pourrait se dupliquer. S'il y a un problème de LIDAR cela n'affectera pas la cartographie de l'environnement.



Alors que s'il y a un défaut lié à l'odométrie du robot cela rendra la représentation de son environnement compliqué.

Le GMapping est utilisé lorsque le robot ne dispose pas d'énormément de ressources.

Gazebo :



C'est un simulateur 3D qui permet de représenter des robots articulés dans des environnements complexes intérieurs ou extérieurs. Ce simulateur est un programme « open source » cela signifie qu'il peut être modifié. Celui-ci est souvent utilisé dans la recherche robotique et dans l'intelligence artificielle. Nous pouvons l'utiliser sur ROS et même utiliser les API (interfaces de programmation), elles permettent de contrôler le robot lors des simulations. Le logiciel permet de faire des simulations réalistes. Le robot peut pousser des objets ou encore rouler sur le sol. Gazebo utilise plusieurs moteurs physiques tels qu'ODE (Open Dynamics Engine) ou encore Bullet.

Le logiciel peut ajouter et simuler des capteurs au robot comme les capteurs de force, de contact, laser et des caméras.

Pour pouvoir créer un environnement de simulation, nous devons créer un fichier ayant pour extension « .world », le langage utilisé de ce fichier est le XML et doit contenir toutes les informations nécessaires pour générer l'environnement. (gravité, ombre, ciel, ...).

E) Liberté d'exploitation, risque de contrefaçon

Cette technologie n'est pas un brevet protégé, alors n'importe qui pourrait développer à leur guise pour une bonne intention. C'est pour cela que le risque se porte en individuel; c'est-à-dire que les causes d'ennuis ou détruire un bien public involontairement sera attribué directement aux développeurs de cette technologie.

F) Aspect réglementaire

Quand on parle de l'automatisation de la conduite, on a tendance à penser que c'est tout ou rien. Or justement, il y a 5 niveaux différents pour caractériser un véhicule autonome. Ils répondent à un classement opéré par la SAE (Society of Automotive Engineers) aux Etats-Unis :

Autonomie de niveau 0

Le conducteur fait tout, aucune fonction n'est automatisée.

Il peut disposer de lane departure warning : si le véhicule franchit la ligne blanche, un signal sonore retentit.

Autonomie de niveau 1

Le conducteur a toujours le contrôle du véhicule : il peut disposer d'ABS, d'anti-patinage, etc...

Autonomie de niveau 2

Le véhicule peut être doté de divers dispositifs d'aide à la conduite tel que Park assist qui permet à la voiture de se garer automatiquement : le système aide à chercher une place de parking adaptée au gabarit de la voiture et effectue une manœuvre automatique une fois l'emplacement détecté, aussi bien en créneau qu'en bataille. Le conducteur doit seulement gérer l'accélération et le freinage (il peut donc retirer les mains du volant).

Certains dispositifs de Park assist sont même feet off , le conducteur peut lever les pieds, voire être à l'extérieur du véhicule pendant que le véhicule se gare.

Le traffic jam chauffeur ou traffic jam assistant: ce dispositif contrôle la vitesse de la voiture et la distance avec la voiture située devant. Il peut être utilisé lorsque le trafic est dense sur autoroute quand la vitesse est au delà de 60 km/h

Autonomie de niveau 3

La voiture de Google, Google car est un véhicule qui présente une autonomie de niveau 3. Ce véhicule est pourvu d'un capteur sur le toit (son prix avoisine 80 000 dollars) dont le fonctionnement est basé sur la technologie laser : il détecte ainsi l'environnement de la voiture à 360 degrés.

Les Tesla sont également des véhicules conçus comme autonomes de niveau 3.

Le système conduit et le conducteur supervise le système mais il n'est pas obligé de superviser en permanence, il peut faire autre chose que conduire : il doit tout de même rester installé au poste de conduite puisqu'il doit pouvoir reprendre le contrôle de la voiture si besoin.

Autonomie de niveau 4

La machine fait tout, la voiture est complètement autonome, elle peut même circuler seule, sans conducteur ni passager et aller chercher des passagers par exemple.

1) En France

La Convention de Genève signée en 1949 et celle de Vienne en 1968 par de nombreux pays n'autorisent pas la circulation de véhicules autonomes de niveau 3 et 4.

En effet, l'article 8 de la convention de Vienne précise qu'il faut un conducteur dans un véhicule et que ce conducteur doit avoir en permanence le contrôle de son véhicule.

L'adoption en juin 2020 de directives détaillées par le forum mondial pour l'harmonisation des réglementations sur les véhicules, qui se tient à l'Unece, Commission économique des Nations unies pour l'Europe, concerne la conduite autonome de niveau 3.

Aujourd'hui en France, nous en sommes au stade 2 du véhicule autonome ce qui consiste à avoir un dispositif d'aide à la conduite tel que la lane assist (l'aide au maintien de la voie) ou le park assist (l'aide au stationnement).

Si la France semble accuser un certain retard en termes de réglementation concernant les voitures autonomes par rapport à ses voisins européens, elle est encore plus en retard par rapport à d'autres pays tels que les Etats-Unis ou encore la Chine et le Japon.

2) A l'international

Le 22 Septembre 2016, le NHTSA (National Highway Traffic Safety Administration), l'administration en charge de la sécurité et des infrastructures routières américaine, a publié un ensemble de règles spécifiques concernant les voitures autonomes à destination des constructeurs automobiles. Le NHTSA se base sur les différents niveaux d'autonomie des véhicules établis par le SAE international, pour donner des directives précises aux constructeurs. Il leur est demandé d'effectuer une évaluation de leurs véhicules autonomes sur 15 points précisément décrits dans le guide. Ils concernent principalement la sécurité des usagers.

Seuls les véhicules ayant satisfait les 15 recommandations pourront effectuer des tests sur la voie publique en vue d'une prochaine commercialisation du véhicule. Les recommandations concernent notamment :

- L'enregistrement et le partage de données,
- Le respect de la vie privée des utilisateurs,
- La sécurité du système,
- La cybersécurité du véhicule,
- La coordination homme-machine,
- La protection des utilisateurs en cas de collision,
- La formation et l'entraînement des consommateurs à l'usage de tels véhicules,
- L'enregistrement et la certification des équipements,
- Le comportement post-accident,
- Le respect des lois étatiques, fédérales et locales,
- Les considérations éthiques,
- La conception opérationnelle de chaque système,
- Le repli en cas de problème,
- La méthode de validation.

Cependant, le NHTSA précise que ces recommandations à destination des fabricants n'ont pas vocation à former un premier corps de lois adopté au niveau fédéral. Pour l'instant, il ne s'agit que d'assurer la sécurité. Sous le système fédéral américain, à défaut de loi fédérale, chaque État est libre d'adopter ses propres règles.

C'est ainsi que la Californie, la Pennsylvanie, l'Arizona, le Nevada, la Floride et plus récemment l'Etat de New York (une quinzaine d'états au total pour l'instant) autorisent depuis le printemps 2011 les tests de voitures autonomes sur leurs autoroutes mais aussi en plein cœur des villes.

Les tests devront être faits par un conducteur expérimenté et accrédité par le fabricant. Par contre, en cas de commercialisation, les conducteurs des véhicules de niveaux 4 et 5 n'auraient pas nécessairement besoin d'être titulaires du permis de conduire. Ces véhicules devraient alors faire l'objet d'une immatriculation spécifique.

Les voitures autonomes sont en test depuis mi 2013 sur les autoroutes de Singapour et au Japon. A cette époque, il n'existait pas de législation spécifique en la matière. Les constructeurs qui testaient leur voiture autonome sur la voie publique de ces deux pays avaient simplement obtenu une autorisation par les autorités locales.

Depuis, au Japon, un plan national pour le développement des véhicules autonomes a été adopté en 2015. Cependant, ce plan ne définit que les acteurs de l'industrialisation des voitures autonomes : il prévoit une association entre le secteur automobile, le secteur électronique et les grandes universités du pays pour concevoir un véhicule sûr et totalement indépendant.

La Chine, qui a également autorisé de nombreux constructeurs nationaux à tester leurs véhicules autonomes sur la voie publique, n'a toujours pas adapté sa législation à ces nouvelles technologies. Elle ne prévoit pas non plus de cadre réglementaire spécifique aux expérimentations.

La Corée du Sud, quant à elle, voit plus grand mais surtout différemment. Aucun cadre réglementaire à l'horizon mais plutôt la création d'une nouvelle ville sur le modèle de la Silicon Valley » entièrement dédiée à la conception, la fabrication et l'expérimentation des voitures autonomes : K-City. La création de cette ville permettrait au pays de développer rapidement et en condition réelle la technologie nécessaire à l'autonomie des véhicules de niveau 5 sans devoir attendre les adaptations législatives nécessaires pour des tests sur la voie publique.

G) Étude de marché

Encore en phase de tests, les voitures connectées et autonomes pourraient représenter un marché de 55 millions de véhicules vendus d'ici 20 ans. Selon une étude du think-tank numérique français Idate, ce sont encore une fois les pays de l'Asie-Pacifique qui seront en tête des ventes et devraient demeurer à la première place d'ici 2040, tous niveaux d'autonomie confondus.

Les constructeurs nord-américains arrivent à la seconde place « mais devraient connaître une croissance moins rapide que leurs voisins asiatiques. Ils sont pourtant toujours leaders des initiatives de tests et ont déjà expérimenté des véhicules autonomes dans 35 villes. L'écosystème européen arrive en suite du classement avec 33 villes testées ».

Les véhicules pleinement autonomes, c'est-à-dire de niveau 4 et 5, ne devraient pas arriver sur le marché avant 2025-2030. Mais l'accélération du marché sera alimentée par deux canaux. Les constructeurs historiques multiplient les initiatives dans le domaine, dernier exemple en date l'alliance entre BMW et Daimler. Mais aussi et surtout les géants de la tech comme Google, Apple, Tesla et Baidu.

Autonomous-Ready Vehicles Net Additions, 2018-2023

Use Case	2018	2019	2020	2021	2022	2023
Commercial	2,407	7,250	10,590	16,958	26,099	37,361
Consumer	134,722	325,682	380,072	491,664	612,486	708,344
Total	137,129	332,932	390,662	508,622	638,585	745,705

Source: Gartner (November 2019)

H) Positionnement sur le marché et avantages concurrentiels/ Coûts/ avantages environnementaux

Le projet est un défi pour créer un véhicule autonome, il sera considéré comme la plus haute classe d'automobile. Mais il est encore incomplet en termes d'IA car la détection de ce qui constitue la nature de l'obstacle ne peut pas être parfaitement effectuée et il y a toujours un problème de machine morale. Notre partie du projet ne traite pas ces problèmes, et se contente de simuler la détection d'obstacles.

I) Programme de travail et dépenses

a) Etape technique

b) Jalon du marché

c) Sécurité juridique (Ip).

J) Qu'allez-vous faire avec le produit/solution du projet ? / Qu'est-ce que le produit/solution à vie ?

Where possible, provide graphical/ schemas

4 Fonction et contribution des acteurs

a) Contexte

Name	Specialization
------	----------------

Justine REYNAUD	OCRES (Connected Objects, Networks and Services)
Elyes ZAHAR	SI (computer system)
Lucas RIETSCH	SI (computer system)
Franck ZHANG	FINANCE
Pascal CHEN	SE (Embedded system)
Pierre MOREAU	SE (Embedded system)

b) Fonction et contribution de chaque acteurs

Elyes : Chef de projet

Justine : Planification des étapes du projet

Lucas : Recherche technique

Franck : Recherche technique

Pascal : Recherche technique

Pierre : Budgétisation

c) Motivation et engagements personnels

Pascal : La découverte de l'automatisme m'a toujours fasciné, j'aimerais apprendre davantage sur le concept de l'intelligence artificielle. La voiture autonome me paraît un bon début pour faire progresser ma connaissance.

Pierre : Étant un passionné des voitures et surtout des nouvelles technologies embarquées dedans. J'ai pour but de me spécialiser dans les véhicules connectés et autonomes. Le fait de pouvoir faire ce projet me donnera un premier aperçu sur les véhicules autonomes et de leur fonctionnement.

Lucas : Ayant toujours été très futuriste, et passionné par le domaine de l'automobile, ce projet est l'occasion pour moi d'améliorer mes connaissances sur les avancées de l'automobile autonome, connectée et intelligente. Il me permet également d'en savoir plus sur les algorithmes mis en œuvre pour mettre en sécurité les usagers du véhicule mais aussi les piétons et autres véhicules qui interagiront avec ce dernier.

Justine : Le domaine des véhicules autonomes est un domaine qui m'a toujours captivé mais aussi pour lequel je n'avais jusque là, très peu de connaissances. Ce PPE est donc pour moi l'occasion de découvrir les principes et le fonctionnement d'un système de véhicule autonome, de nouvelles technologies et surtout de nouvelles compétences de programmation.

Elyes : Un des principaux enjeux du 21ème siècle est de fournir à l'homme un environnement sécurisé. C'est dans cette optique que s'intéressait au enjeu tel que la sécurité routière fût moteur de ma motivation. De plus, notre véhicule nous permettra d'acquérir de nouvelles compétences comme la programmation ROS, ce qui accentue ma considération pour ce projet.

5 Environnement de test

Notre projet consiste à faire une maquette robotique contenant une intelligence artificielle ainsi qu'une géolocalisation pour définir une destination et de modifier la trajectoire du prototype au fur à mesure des rencontres d'obstacles et d'ajuster la vitesse de déplacement en respectant les vitesses maximales autorisées.

5.1 Environnement physique

Il est nécessaire de simuler une voie routière comme afficher les panneaux de signalisation du code de la route. Notre projet a lieu sur une maquette de voie routière. Ainsi l'observation de notre prototype est faite grâce aux différentes caméras disposées sur celui-ci et avec nos yeux également.

5.2 Environnement matériel

L'intelligence artificielle est le principal technologique qui nous sera fourni pour reconnaître la nature de l'obstacle et comprendre les signalisations des panneaux comme la limitation de vitesse ou une ligne de stop. Puis l'assistant de navigation GPS ou un système de cartographie sera intégré dans notre projet également pour se localiser en temps réel et aussi pour définir la destination.

5.3 Environnement logiciel

Il n'y a pas de logiciels autres que ceux que nous utilisons pour programmer notre prototype.

5.4 Données

Afin de pouvoir faire fonctionner notre maquette de voiture autonome, une base de données sera nécessaire pour la maquette. Afin de pouvoir implémenter les différentes structures, panneaux et obstacles que la maquette sera susceptible de rencontrer sur la route.

6 Suite de tests

6.1 Exigence 1

No.	Description	Execution scenario	Expected results	OK/NOK
1	Capacité à rouler tout droit	Faire parcourir le robot d'un point A à un point B	Parcours du point A au point B de manière rectiligne et avec une vitesse uniforme	
2	Capacité à prendre des virages	Faire faire au robot un virage de 90°	Virage réussi	
3	Capacité au véhicule d'accélérer et décélérer	Faire parcourir le robot sur une ligne droite de A à C en passant par B. A->B : accélération du robot. B->C : décélération du robot	Augmentation de la vitesse entre A et B. Réduction de la vitesse de B à C.	

6.2 Exigence 2

No.	Description	Execution scenario	Expected results	OK/NOK
4	Capacité à reconnaître les panneaux de signalisation	Afficher un panneau de stop devant le robot	Panneau détecté par le système du véhicule	
5	Capacité à détecter un obstacle	Mettre soudainement un obstacle devant le robot	Obstacle détecté via fenêtre de simulation ROS	
6	Capacité à esquiver un obstacle	Mettre un obstacle sur la voie du robot	Detourner l'obstacle	

6.3 Exigence 3

No.	Description	Execution scenario	Expected results	OK/NOK
7	Planifier sa trajectoire en fonction de la signalisation	Faire parcourir le robot sur un circuit type "urbain"	Changement de trajectoire suite à une indication sur le chemin	
8	Capacité à planifier sa trajectoire en continue	Faire parcourir le robot sur un circuit type "urbain"	Changement de trajectoire en direct en fonction de l'environnement qui l'entoure	

7 Conclusions & Perspectives

La voiture autonome est un grand enjeu du 21^{ème} siècle. Plusieurs grands groupes se sont déjà lancés dans la course comme Tesla, Uber, Google, BMW, Renault, et pleins d'autres. Dû aux récents événements qui ont frappé le monde de l'automobile autonome, plusieurs accidents ont été recensés durant cette dernière décennie comme par exemple en mars 2018, un véhicule Uber a percuté une femme de 49ans suite à une erreur de détection de la part du logiciel embarqué. Ceci a eu pour effet d'un retour en arrière des technologies développées jusqu'à présent, malgré les justifications des constructeurs à ce sujet, la voiture autonome n'est pas encore prête à voir le jour.

C'est dans ce contexte-ci que nous nous sommes intéressés à ce projet car en tant que futur ingénieur, c'est de notre responsabilité d'être acteur de cette transition technologique. Les véhicules autonomes auront un impact fondamental sur l'évolution de la vie quotidienne. A travers ce projet nous espérons pouvoir élargir notre spectre de connaissances notamment en développant nos compétences de programmation, en apprenant l'utilisation de l'outil de simulation de robotique ROS et en comprenant le fonctionnement global du véhicule autonome des capteurs Lidar aux caméras binoculaires.

On aimerait pouvoir contacter des constructeurs automobiles et des fournisseurs d'applications GPS afin de pouvoir implémenter notre solution dans leurs produits

Explicatin datasheet :

ROSbot 2.0

ROSbot est une plate-forme de robot mobile autonome 4x4. Ce robot est propulsé par ROS.

Il possède :

- Une plateforme mobile à 4 roues motrices contient des moteurs DC (à courant continu) avec des encodeurs et la structure est en aluminium.
- Une caméra 3D Orbbec Astra RGB-D
- Un capteur inertiel MPU 9250 (accéléromètre + gyroscope)
- Un panneau à l'arrière fournissant des interfaces pour l'ajout de modules supplémentaires
- Un contrôleur CORE2-ROS avec Asus Tinker Board et avec Rockchip RK3288 jusqu'à 1,8 GHz, 2 Go de RAM DDR3 et 32 Go de MicroSD.
- Un scanner laser RPLIDAR A2



Front

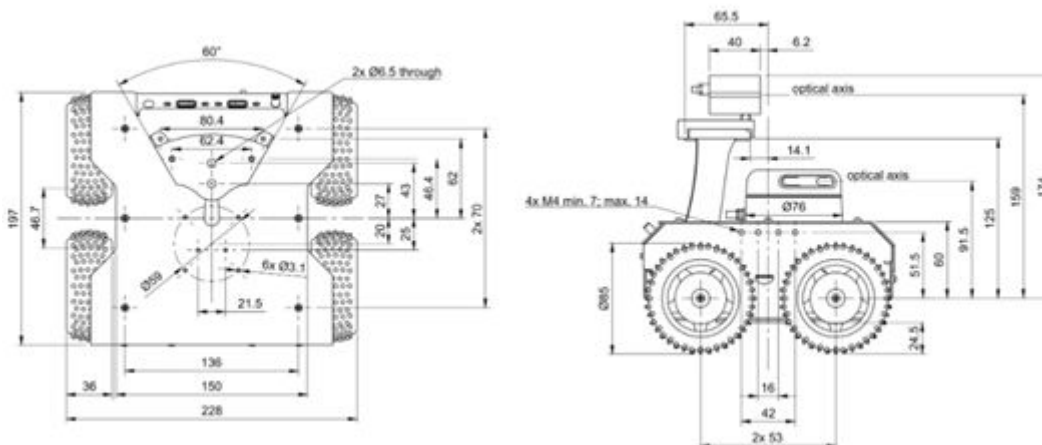


Back

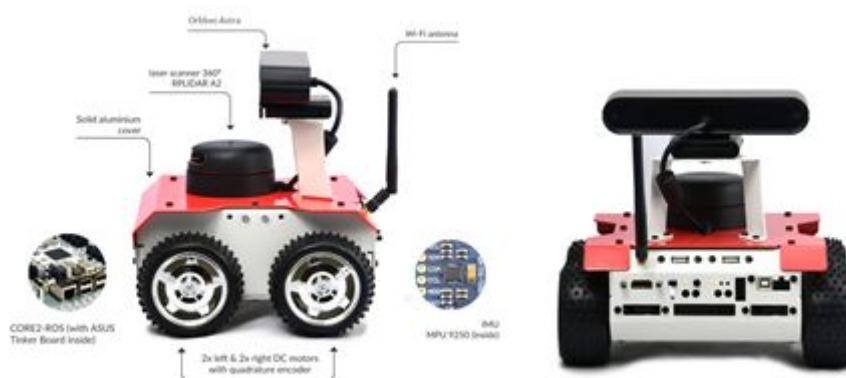


Perspective

Guide hardware :



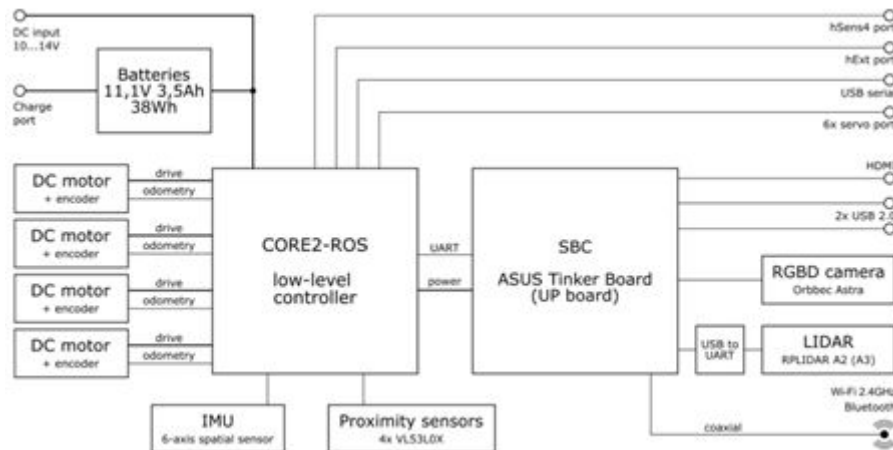
Attributs	Description
Dimensions avec la caméra et LiDAR	200 x 235 x 220 mm / 7.87 x 9.25 x 8.66in [LxWxH]
Dimensions sans la caméra	200 x 235 x 146mm / 7.87 x 9.25 x 5.74in [LxWxH]
Dimensions sans la caméra et sans LiDAR	200 x 235 x 106mm / 7.87 x 9.25 x 4.17in [LxWxH]
Poids	2.84kg / 100oz (avec caméra et LiDAR), 2,45kg / 86oz (sans caméra et LiDAR)
Diamètre de la roue / Passage de roue / Empattement	85mm / 22mm / 105mm
Matériel du châssis	Plaque en aluminium peint par poudrage, épaisseur 1.5mm
Vitesse de translation maximale	1.0 m/s
Vitesse de rotation maximale	420 deg/s (7.33 rad/s)
Capacité de charge maximale	10kg / 352oz
Vie de la batterie	1.5h – 5h



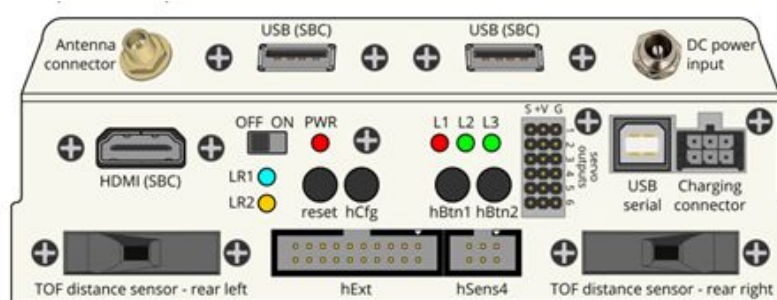
Description des composants :

Composants	Quantité	Description
Capteur de distance infrarouge	4	VL53L0X Time-of-Flight distance sensor with up to 200 cm range
CORE2	1	Contrôleur en temps réel basé sur le microcontrôleur STM32F407.
Moteur DC	4	Moteur Xinhe XH-25D, Moteur utilisé : RF-370, 6VDC nominal, 5000 tr / min, vitesse à vide sur l'arbre de sortie : 165 tr / min, couple de décrochage : 2,9 kg * cm, courant de décrochage : 2,2 A, rapport de vitesse : ~ 34 (exact ratio est 30613/900), encodeur : magnétique, 48ppr, 12 pôles
Capteur IMU	1	Sortie numérique accéléromètre 3 axes / Capteur gyro / magnétomètre avec MPU-9250
Caméra RGBD	1	Orbbec Astra RVB avec taille d'image 640x480 et taille d'image profondeur 640x480.
Batteries	3	Batteries rechargeables protégées Li-Ion 18650, capacité de 3500 mAh, tension nominale de 3,7 V. Remarque : l'appareil peut être expédié de manière interchangeable avec des batteries similaires.
Antenne	1	Connectée directement au module Wi-Fi ASUS Tinker Board. Utilise un câble RP-SMA (m) <-> I-PEX MHF4 pour connecter l'antenne au SBC.
SBC	1	ASUS Tinker Board avec 2 Go de RAM, Rockchip RK 3288 avec 4x 1,80 GHz comme CPU et un ARM Mali-T764 MP2 comme GPU et 32 Go MicroSD. Le SBC fonctionne sur un système d'exploitation basé sur Ubuntu, personnalisé pour utiliser ROS.
LIDAR	1	RpLidar A2, 360 degrés et jusqu'à 8 m de portée

Diagramme



Description du panneau arrière

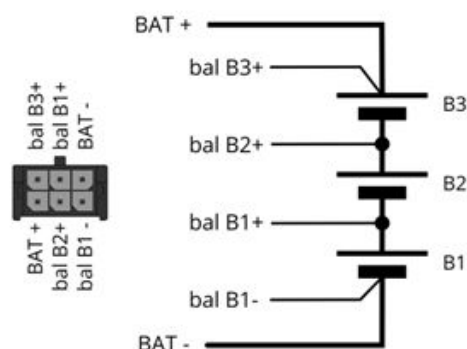


Composants	Quantité	Description
Connecteur d'antenne	1	Prise RP-SMA de l'antenne Wi-Fi. Requis pour la connectivité Wi-Fi.
USB	2	Ports hôtes USB 2.0 de SBC.
HDMI	1	Sortie HDMI de SBC.
Interrupteur	1	Active ou désactive complètement ROSbot.
LEDs	6	LR1 (jaune), LR2 (bleu), L1 (rouge), L2 (vert), L3 (vert), PWR (rouge)
Bouton de réinitialisation	1	Bouton utilisé pour réinitialiser CORE2.

hBtn	2	hBtn1, hBtn2 – boutons programmables.
Sorties pour servo	6	Sorties servo avec PWM
Port série	1	Port série USB utilisé pour déboguer le micrologiciel sur le contrôleur CORE2-ROS.
Connecteur de charge	1	Connecteur à 6 broches pour charger les batteries Li-Ion internes.
Entrée d'alimentation DC	1	DC pour travailler avec une alimentation externe 12V. Utilisez l'alimentation fournie avec le chargeur ou tout autre 12V, min. Bloc d'alimentation 5 A avec fiche 5,5 / 2,5 mm (centre positif).
Capteur de distance Time-of-Flight	2	Capteur de distance Time-of-Flight VL53L0X avec une portée jusqu'à 200cm
hExt	1	12xGPIO, 7x ADC, SPI, I2C, UART
hSense	1	4 xGPIO, ADC, UART

Source de courant

Le ROSbot est alimenté par une batterie interne rechargeable Li-Ion contenant 3 cellules Li-Ion, connectées en série. Ce type de connexion est appelé «3S ». Le schéma ci-dessous explique comment les cellules sont câblées entre elles et avec le connecteur de charge (côté ROSbot).



BAT + et BAT- sont les connexions d'alimentation et les fils « bal Bx » sont utilisés pour surveiller la tension sur chaque cellule. Il est fortement recommandé de maintenir des tensions égales sur chaque cellule pendant le

processus de charge. Le chargeur inclus avec ROSbot peut charger les batteries de la manière décrite et, grâce à cela, la longue durée de vie du jeu de batteries est possible.

La tension nominale de chaque cellule est de 3,7 V mais la plage utile est de 3,2 V à 4,2 V.

Il y a un indicateur de décharge si le bon micrologiciel est préchargé sur le contrôleur interne (CORE2).

La LED1 indique l'état d'alimentation :

- La LED1 est allumée si le robot est allumé.
- Lorsque la LED1 se met à clignoter, c'est que la charge de la batterie est insuffisante.

Attention : toujours s'assurer que le micrologiciel utilisateur contient toujours la fonction qui surveille le niveau de tension d'alimentation. Une décharge totale de la batterie peut réduire son cycle de vie. La décharge à une tension inférieure à 3,0 V / cellule peut également déclencher la protection contre les décharges excessives. Si la tension est trop basse, éteignez le ROSbot et chargez les batteries dès que possible.

Pour charger ROSbot

Le kit chargeur comprend :

- Un chargeur Redox Beta
- Un adaptateur secteur AC / DC 100 ... 240V à 12V 5A avec fiche 5,5 / 2,5 mm sur le côté 12V
- Un câble pour connecter le chargeur avec le port de charge de ROSbot.
- Un guide de charge rapide

Software

Le logiciel pour ROSbot peut être divisé en 2 méthodes :

- Un micrologiciel de bas niveau qui fonctionne sur le contrôleur en temps réel (CORE2). Il peut être développé à l'aide de l'IDE de Visual Studio Code.
- OS basé sur Ubuntu 16.04, qui fonctionne sur le SBC (ASUS Tinker Board ou UpBoard) et contient tous les composants nécessaires pour commencer à travailler avec ROS immédiatement. La carte microSD avec OS est incluse avec chaque ROSbot. Le système d'exploitation a été modifié pour rendre le système de fichiers insensible aux coupures de courant soudaines.

Pour notre projet nous allons opter pour la deuxième méthode

ROS API

Vous trouverez ci-dessous les sujets et services disponibles dans ROSbot:

Topic	Message type	Direction	Node	Description
/imu250	rosbot_ekf/Imu	publisher	/serial_node	Raw IMU data in custom message type
/range/fl	sensor_msgs/Range	publisher	/serial_node	Front left range sensor raw data
/range/fr	sensor_msgs/Range	publisher	/serial_node	Front right range sensor raw data
/range/rl	sensor_msgs/Range	publisher	/serial_node	Rear left range sensor raw data
/range/rf	sensor_msgs/Range	publisher	/serial_node	Rear right range sensor raw data
/joint_states	sensor_msgs/JointState	publisher	/serial_node	Wheels rotation angle
/battery	sensor_msgs/BatteryState	publisher	/serial_node	Battery voltage
/buttons	std_msgs/UInt8	publisher	/serial_node	User buttons state, details in User buttons section
/pose	geometry_msgs/PoseStamped	publisher	/serial_node	Position based on encoders
/odom/wheel	nav_msgs/Odometry	publisher	/msgs_conversion	Odometry based on wheel encoders
/velocity	geometry_msgs/Twist	publisher	/serial_node	Odometry based on encoders
/imu	sensor_msgs/Imu	publisher	/msgs_conversion	IMU data wrapped in standard ROS message type
/odom	nav_msgs/Odometry	publisher	/rosbot_ekf	Odometry based on sensor fusion
/tf	tf2_msgs/TFMessage	publisher	/rosbot_ekf	ROSbot position based on sensor fusion
/set_pose	geometry_msgs/PoseWithCovarianceStamped	subscriber	/rosbot_ekf	Allow to set custom state of EKF
/cmd_vel	geometry_msgs/Twist	subscriber	/serial_node	Velocity commands
/config	rosbot_ekf/Configuration	service server	/serial_node	Allow to control behaviour of CORE2 board, detaild in CORE2 config section