

DIGRACI PAUL
DEPARTEMENT IESE - POLYTECH GRENOBLE
Rapport de stage de 4^{ème} année.

ROBOT SUIVEUR DE LIGNE – WINTER CHALLENGE

RAPPORT UNIVERSITAIRE
ET
ANNEXE

Année universitaire 2018 - 2019
Mai – Août 2019

Table des matières

I – Introduction	3
II – UNESP / FEG	4
III – Réflexion préprojet.....	5
A – Technologie des capteurs.....	5
B – Architecture Mécanique.....	5
C – Architecture Electrique.....	5
D – Choix réalisés.....	5
IV – Le projet	6
A – Aspect Mécanique	6
B – Aspect Electrique et Electronique	6
C – Aspect Programmation.....	7
V – Le Stage.....	7
A - Réflexion.....	7
B – Programmation.....	8
C – Mécanique.....	10
D – Restructuration	12
E – Résultats.....	15
VI – Conclusion	16
XI – Annexe	17

I – Introduction

L'objectif premier du stage est de concevoir un robot suiveur de ligne capable de participer à la Winter Challenge, une compétition internationale au Brésil. Pour ce faire il doit être le plus rapide possible. Le robot doit suivre un circuit de ligne blanche sur fond noir connu uniquement le jour de la compétition. Plusieurs obstacles sont présents, des virages plus ou moins serrés, des croisements, zigzag... Le robot doit être entièrement autonome. Le règlement nous permet d'avoir trois essais, ne retenant que le meilleur temps, il y a donc un gros travail de gestion des vitesses. Il est bien entendu interdit de modifier la programmation entre les essais.

Le stage est une continuité du projet co et du projet de 4^{ème} année. Un important travail de recherche et de conception a été donc fait en amont. Le stage avait donc pour objectif d'optimiser le travail réalisé en France. Entre le moment de la prise de connaissance du sujet et la date de la compétition, environ 9 mois c'est écoulé.

Le cahier des charges comportait cinq règles qui sont les suivantes :

- Les dimensions maximales sont 250 mm en longueur, 250 mm en largeur et 200 mm en hauteur.
- Le robot doit être autonome. Il ne doit pas être contrôlé manuellement.
- Le robot doit suivre la ligne et en aucun cas se retrouver en dehors de celle-ci ou prendre de raccourcis.
- Le robot doit s'arrêter dans une zone d'un mètre après la première ligne d'arrivée
- Le robot doit pouvoir faire les trois essais à la suite sans être rechargé.

II – UNESP / FEG

Le stage a été réalisé dans la faculté d'ingénierie de Guaratinguetá (FEG), faisant partie de l'UNESP (Université publique de l'état de Sao Paulo). Agnelo Marotta Cassula a encadrés le projet sur place. Nous sommes 5 élèves de IESE à faire notre stage à la FEG, réparti en deux groupes, proposant des robots de différent type (Sumo et Suiveur de ligne).

Guaratinguetá est une ville de 120 000 habitants dans l'état de Sao Paulo. Elle se situe entre Rio de Janeiro et Sao Paulo. La FEG a été construite en 1967, elle possède neuf cursus techniques, tel que la mécanique, l'énergie, ou le génie électrique, pour une surface de 175 800 m² dont 14 500 m² de locaux. Environ 190 enseignant et 1 600 étudiants y travail.

L'emploi du temps des étudiants est complètement différent par rapport au notre. Effectivement, beaucoup d'élève travaille en parallèle de leurs études. Les cours sont donc adaptés à cette situation. Il y a des cours en début de matinée et en fin de journée, laissant donc les journées libres pour de possible jobs étudiants. Après les cours du soir, certains étudiants donnent des cours complémentaires, cela permet d'aider les plus jeunes élèves mais aussi de pallier le manque de professeur. Les activités de club se déroule durant leur temps libre, en particulier en fin d'après-midi.

Différents clubs existent au sein des universités. Ils sont entièrement gérés par les étudiants. Ils n'apportent aucun avantage dans leurs études (notes), mais peut être valorisable pour leurs futures recherches d'emploi. L'université ne leur fourni aucune aide financière. Les étudiants doivent donc trouver des partenaires pour réaliser les projets. Le club de robotique de la FEG a été fondé en 2011 et regroupe une trentaine de membre de tout niveau. Ils participent chaque année au Winter Challenge.

Nous avons pu remarquer que les clubs sont très organisés. En effet, il y a plusieurs groupes de travail en fonction des études réalisées ou envies. Certains travaillent sur l'aspect mécanique, électronique ou informatique. D'autres s'occupent de rechercher de possible nouveaux partenaires / sponsors, la communication a aussi son importance. Pour entrer dans ces associations il faut passer des tests et entretiens, enfin de vérifier si le candidat peut apporter quelque chose à celle-ci.

Comme nous, la FEG robotica propose chaque année plusieurs robots. Pendant notre stage, il y avait notamment plusieurs robots sumos de différentes catégories et un suiveur de ligne (avec leur équipe attitrée).

III – Réflexion préprojet

A – Technologie des capteurs

Plusieurs options s'offraient à nous. Des capteurs IR, utilisés par une majeure partie des robots participants à cette compétition. Un capteur optique (caméra), utilisée par l'équipe précédente. Nous avons eu la chance d'avoir un feed back important de l'équipe précédente. Nous avons donc décidé de conserver le même capteur. En effet celui-ci à l'avantage de permettre une meilleure anticipation des virages afin d'optimiser les trajectoires et vitesse de robots.

B – Architecture mécanique

En ce qui concerne la partie mécanique, nous avons décidé de complètement refaire celle-ci. Effectivement l'an passé, le suiveur de ligne était complètement surdimensionné. Nous avons donc décidé de faire un robot bien plus petit et maniable. Nous passons de 2 roues motrices avec une roue libre à 4 roues motrices. Le manque de puissance est compensé par un nombre plus important de moteur mais aussi pas la légèreté de la structure.

C – Architecture électrique

Nous avons pu voir les solutions utilisées par nos prédécesseurs, c'est-à-dire une batterie alimentant uniquement la raspberry (microcontrôleur) et une autre pour les moteurs. La puissance transmise étant gérée via un driver. Nous avons jugé ça bien trop imposant. Ainsi nous avons pris la décision de n'utiliser qu'une seule batterie. En utilisant une carte d'alimentation nous pouvons adapter la tension afin d'alimenter toute notre contrôler sans l'endommager.

D – Choix réalisés

Suite à un feed back important rapportant les défauts rencontrés mais aussi par une longue analyse du cahier des charges en nous basant aussi sur le robot précédent. Nous avons décidé de changer le système de direction, la motorisation et réfléchissant à un asservissement plus efficace. Nous gardons la caméra utilisée et nous comptons améliorer le traitement de l'image.

IV – Le projet

Avant d'être mon sujet de stage, le robot suiveur de ligne est aussi mon sujet de projet co et de projet de 4^{ème} année. Initialement je travaillais pour tous les robots, recherchant des possibles partenaires et le matériel désiré sur les sites web de vente autorisés par Polytech. J'ai pu réellement rejoindre l'équipe du suiveur de ligne par la suite. Nous étions 3 élèves, 2 en IESE 4 et 1 en IESE 3. Mon travail a donc été d'aider mes collègues dans deux secteurs, la réalisation mécanique et la programmation de notre suiveur. Nous les verrons ici de façon succincte.

A - Aspect mécanique

La conception structurelle reprend ce que l'on peut voir sur internet. Un aspect minimaliste, utilisant des tiges en laiton. Le robot a alors creux et possède un poids inférieur au 1kg. La fixation des tiges entre elles se fera par soudure (soudure à l'étain). Il est important d'estimer les dimensions et le poids de celui-ci pour choisir les moteurs et la batterie la plus adaptée.

Nous avons par la suite choisi des moteurs en recherchant une vitesse de 1m/s. En choisissant des roues de 20mm de diamètre nous déterminant une vitesse de rotation nominale de 500rpm. De plus nous voulons une accélération de 1m/s^2 , d'après nos calculs de vitesse, nous en déduisons que notre robot aura une vitesse maximale de 1.047m/s (sans glissement), et qu'il devra mettre 1.05s et 0.548m pour atteindre cette vitesse. De ceci nous déduisons que nos moteurs devront développer une force de 1N pour faire accélérer le robot, et par conséquent un couple de 0.02Nm. Pour répondre à ce besoin, nous avons décidé d'utiliser 4 motoréducteurs à courant continu, fonctionnant sous 6V, délivrant un couple de 0.08Nm et ayant une vitesse de rotation en sortie sans charge de 600rpm. Ces moteurs pesants chacun 9.5g, ils ajouteront donc un poids total de 38g au robot, ce qui est presque négligeable devant 1kg total.

B - Aspect énergétique et électronique

Les moteurs fonctionnent avec une tension de 6V, il nous est nécessaire de choisir une batterie délivrant une tension au moins supérieure. Nous décidons de concentrer nos recherches sur les Li-Po. En effet ils possèdent le meilleur rapport poids/puissance. Nous trouvons donc une batterie de 7.4V pour 1300mAh, ce qui est bien supérieur à la capacité nécessaire pour faire nos 3 tours.

La raspberry peut être alimentée par ses pins d'alimentation mais seulement pour une tension entre 4.7V et 5.3V et un courant entre 0.7A et 1.2A. Nous utilisons donc un régulateur de puissance, le L7805. Un dissipateur thermique peut être utilisé pour éviter les surchauffes.

Par la suite nous avons intégré un shield pour contrôler les moteurs. Les moteurs n'ont besoin que d'une direction. Effectivement le robot n'a pas besoin d'aller en arrière. Notre shield est donc assez simple. La gestion de la puissance se fera par l'utilisation de deux

PWM (un pour chaque côté). Nous utilisons pour se faire des transistors MOSFET, ils sont bien plus résistants que des transistors bipolaires, de plus ils nous fournissent un bien meilleur signal (on a moins de perte). Nous verrons ce circuit en détail dans une autre partie, celui-ci ayant été modifié durant le stage.

C - Aspect programmation

Concernant la programmation nous avons décidé de reprendre une partie du programme précédant. Le langage utilisé est Python. Il est plutôt simple à prendre en main, de plus il permet l'utilisation de la bibliothèque de vision par ordinateur OpenCV. Au niveau pratique nous utilisant aussi VNC, ce qui nous permet d'accéder au bureau de la raspberry via nos propres ordinateurs. Cela nous évite notamment de connecter un clavier, une souris et un écran.

OpenCV est un outil puissant. Il nous permet de reconnaître les différents éléments sur la piste et les recréer virtuellement. C'est nécessaire pour pouvoir faire des calculs d'angle par exemple. L'algorithme est similaire à ce qui a été fait l'année dernière. Un gros travail a donc été fourni pour comprendre toutes les fonctions existantes et leur utilité dans notre projet. Des fonctionnalités ont été ajoutées, tel que la gestion de la luminosité, évitant les problèmes de reflet et différentes fonctions d'affichage des données.

Avant le début de mon stage (étant le seul à partir dans l'équipe suiveur de ligne), le robot était fini au niveau mécanique et électrique. Le travail restant étant de finir la programmation de celui-ci.

V – Stage

A – Réflexion

Le stage m'a appris une chose. Quoi que l'on puisse prévoir, rien ne se passe comme prévu. L'important est donc de savoir rebondir sans perdre espoir. Effectivement de nombreux problèmes sont survenus lors de mon stage.

Dès le début nous avons appris que la Winter Challenge n'aura pas lieu durant notre séjour. Nous avons donc espoir d'avoir des compétitions locales pour remplacer celui-ci. De plus le voyage a été éprouvant pour notre suiveur de ligne. En effet la structure minimaliste n'a pas supporté la brutalité du voyage en soute.

Après avoir fait connaissance avec les membres de la FEG robotica. Mon travail a donc été dans un premier temps de resouder la structure et certains moteurs. Une partie assez laborieuse, les moteurs étant difficile à souder. L'étape suivante a été d'effectuer des tests pour vérifier l'intégrité physique dans un premier temps, puis fonctionnel dans un second

temps. C'est durant ces tests qu'un nouveau problème est survenu. La raspberry suite à un reboot a généré un signal non désiré ce qui a conduit à la chute de robot ainsi qu'à la destruction du contrôleur. La structure bien trop fragile n'a pas supporté le choque. Un nouveau questionnement a donc été nécessaire.

- Comment acquérir une nouvelle raspberry ?
- Comment améliorer la structure afin d'éviter de nouveaux problèmes structurels ?

En ce qui concerne l'obtention d'une nouvelle raspberry, plusieurs choix sont possibles. La première étant d'en acheter une sur place, le problème étant le prix et le temps de livraison bien supérieur à celui de la France. J'ai alors demandé à M. Eon de m'en rapporter une. En effet, il devait passer nous voir dans 1 semaine. La programmation a donc été mis de côté durant cette période.

Après une longue réflexion, une solution simple ne nécessitant pas une modification complète est ressortie. En ne prenant en compte que les parties fragiles ayant le plus de contrainte, l'idée de fixation des tiges inférieures en fixant les moteurs sans collage ni soudure m'est devenu évident. Ainsi j'ai commencé à modéliser différentes pièces en 3D via le logiciel SolidWorks.

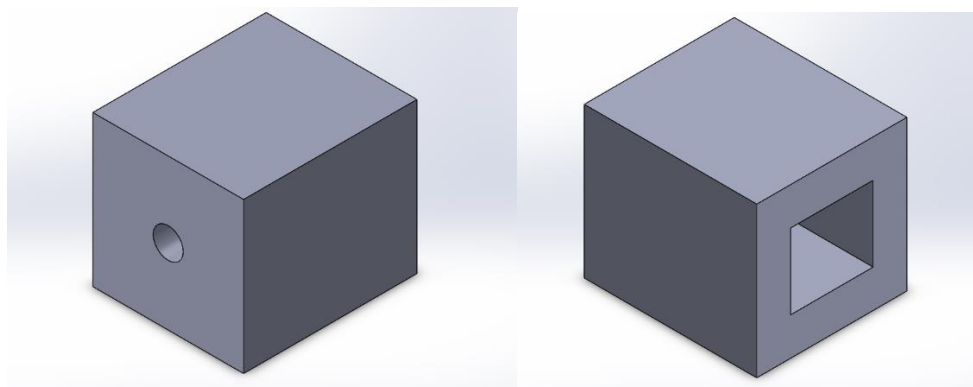


Figure 1 & 2 : Support moteur

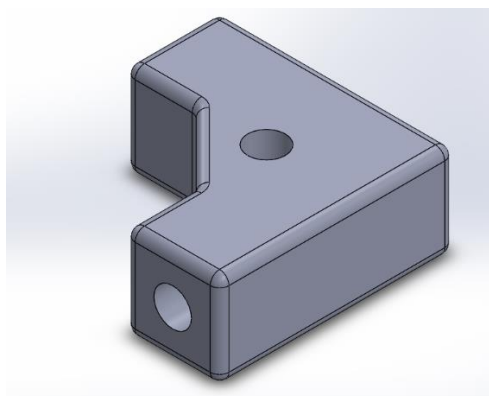


Figure 3 : Fixation des tiges

Après discussion avec mes collègues français, le modèle a été validé. Un étudiant brésilien ayant accès à l'imprimante 3D, m'a proposé de s'occuper de l'impression. Il ne manquait plus qu'à attendre patiemment.

Peu de temps après, j'ai pu recevoir une nouvelle raspberry. Il était donc temps de s'occuper de mon travail initial, finir le programme. Mais avant toute chose, une réinstallation de tous les outils et bibliothèques est nécessaire. Nous en arrivons donc à un nouveau problème. Certaines installations de modules ne s'effectuent pas correctement. En effet des erreurs apparaissent. Ce n'est qu'après de nombreuses heures de recherche, tout en essayant de trouver l'origine des problèmes que j'ai pu avancer. En effet, des problèmes de certification de connexion empêchaient de télécharger les éléments. Une solution simple a pu alors être trouvée, « PYTHONHTTPSVERIFY=0 ». Cette commande permet d'éviter la vérification, ce qui me débloque entièrement.

B – Programmation

La plupart des problèmes réglés, cela me permet enfin de pouvoir commencer la programmation. Dans un premier temps, j'énumère tout ce qui a été fait. Ainsi j'identifie ce qui me reste à faire.

J'identifie notamment un réel problème lors des croisements. Le programme actuel ne permet pas d'identifier les croisements, créant des angles de trajectoire totalement aberrant. L'idée est donc d'utiliser la comparaison d'aire. Effectivement, comme on peut le voir sur le schéma ci-dessous on peut créer des rectangles englobants la totalité des lignes (en vert). Celle-ci est bien supérieure à l'aire des lignes (en rose).

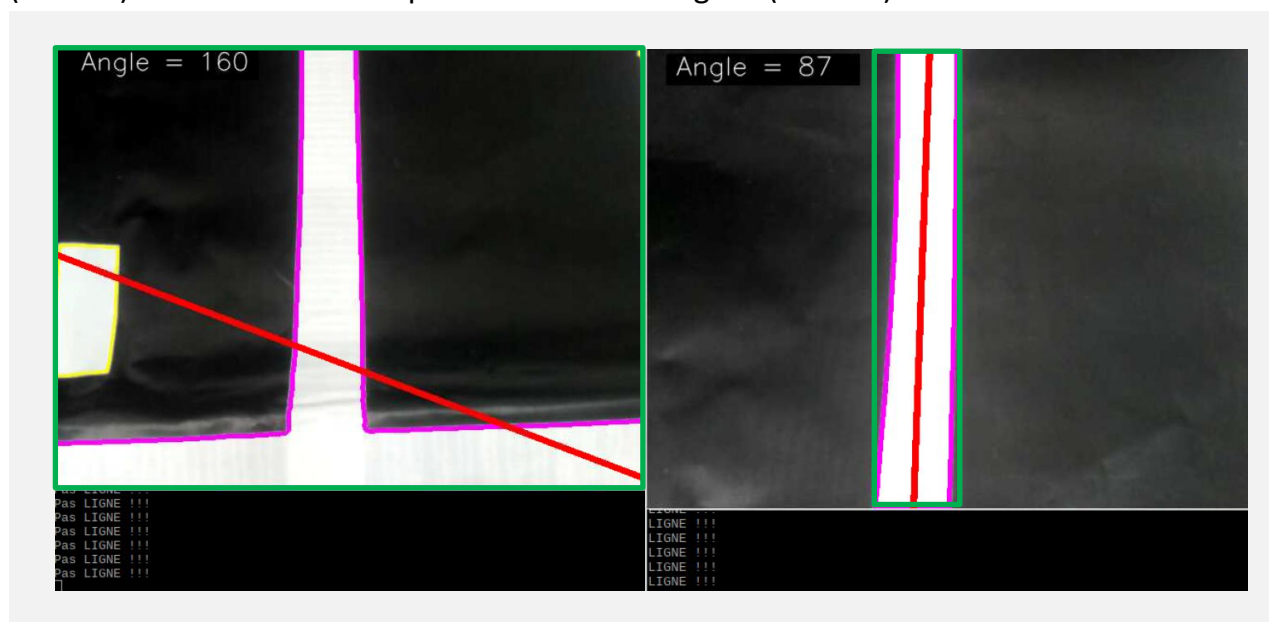


Figure 4 : Comparaison des aires

Cette solution peut en revanche poser problème dans les virages. Mais c'était sans compter sur l'utilisation des pastilles de virage. Je n'aurais qu'à activer ou non ce comparatif.

Un autre problème lié au croisement concerne les pastilles. La caméra arrive à détecter les pastilles de signalisation de virage de l'autre voie. De la même façon, j'effectue une comparaison. Les pastilles étant des rectangles et non des carrés, il est assez simple de savoir le sens de celles-ci en fonction du rapport longueur / largeur.

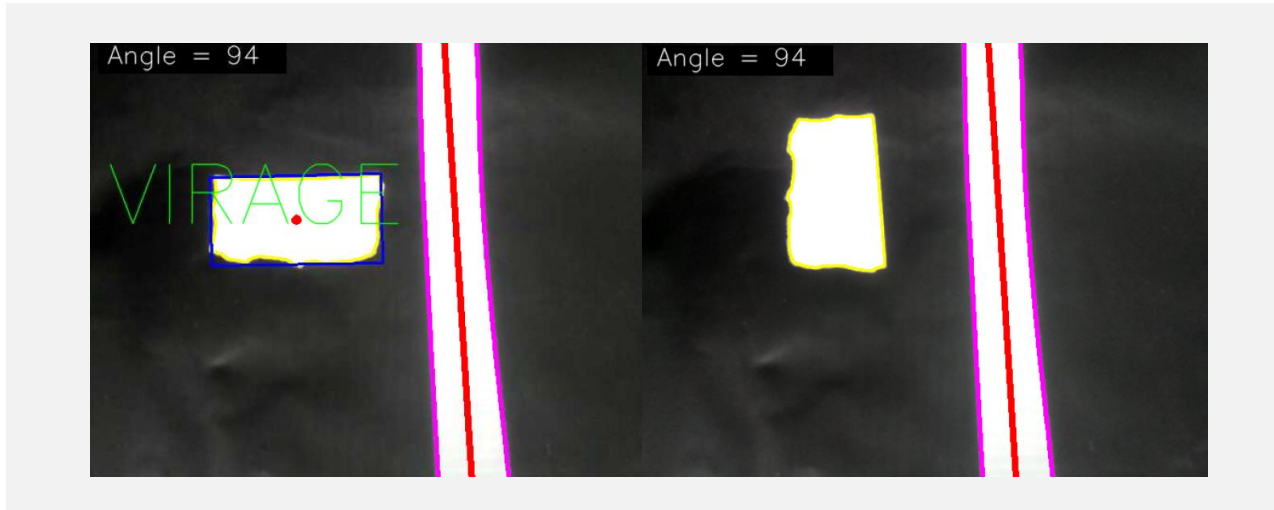


Figure 5 : Détection des pastilles

La partie traitement de l'image est ainsi achevée. Seul l'assemblage de toutes ces fonctions est alors à faire. Nous entrons dans une autre partie, la gestion des moteurs.

Plusieurs solutions sont possibles. Dans un premier temps l'idée de gérer les moteurs avec des fonctions simples à mettre en œuvre est privilégiée.

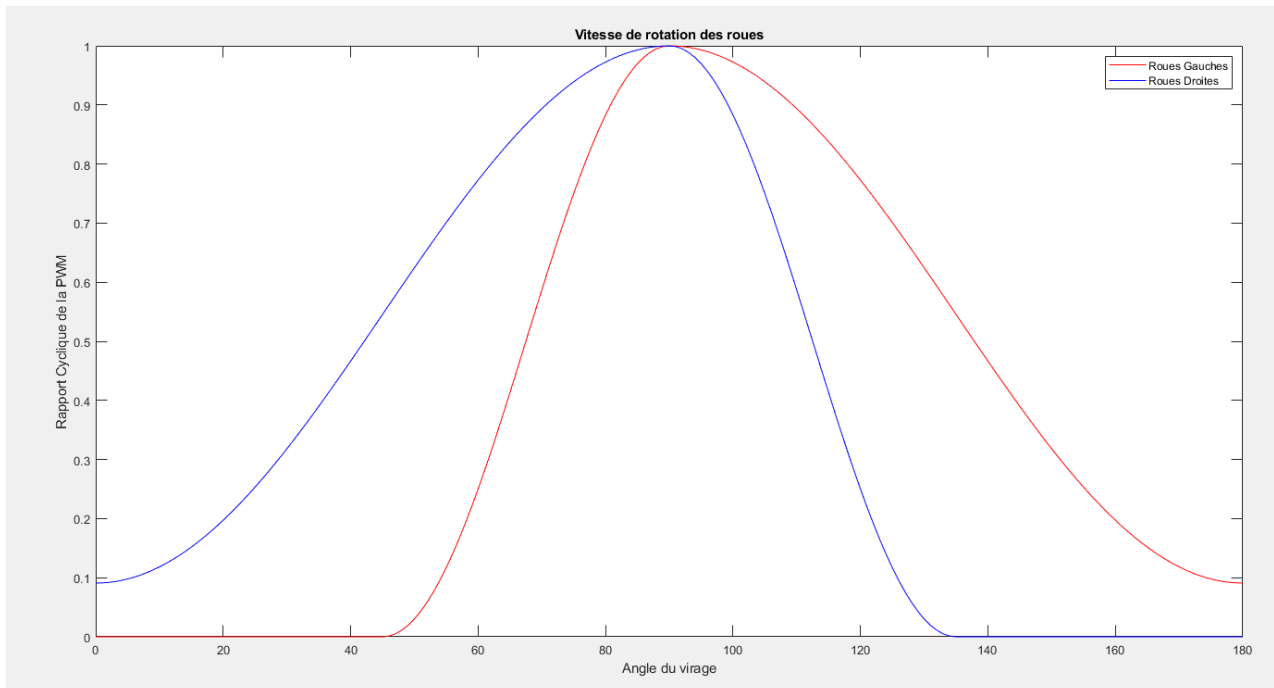


Figure 6 : Vitesse de rotation des roues

Les fonctions les plus élégante afin d'éviter tout changement brutale de vitesse est l'utilisation des fonctions cosinus sous conditions.

$$PWM_{gauche}(angle) = \begin{cases} Si\ angle < 45\ alors = 0 \\ Si\ angle > 90\ alors = \cos(angle * 2 + (1 + 2 * x)) / (2 + x * 2) \\ Sinon = \cos(angle * 4 + 1) / 2 \end{cases}$$

Ici X représente la PWM minimale pour tourner (ci-dessus nous voyons que sa valeur est 0.1). La fonction est du côté droit est symétrique à celle de gauche.

Un paramètre que je dois prendre en compte est la vitesse du robot. En effet si le robot se déplace trop rapidement, alors ces fonctions devront être divisé par une constante Y. Le but étant de diminuer la valeur de la PWM maximal.

Concernant le programme en tant que tel, il se découpe en plusieurs blocs. L'ensemble des algorithme sont dans les annexes. L'annexe 1 correspond à l'ensemble du programme. On peut observer différents processus détaillé en annexes 2, 3 et 4. Ainsi ils permettent de parer aux différentes éventualités que l'on peut rencontrer sur la piste (les croisement, les virages (pastilles), l'arrêt de fin ...) ce que le programme de base (sans les processus) ne permet pas. Celui-ci permettant juste de suivre une ligne. Le code est disponible sur GitHub : github.com/PaulAroide/Liska

C - Mécanique

A ce moment-là, pas mal de temps c'est écoulé. Mes pièces n'étant toujours pas faite malgré les nombreuses relances. Un choix a dû être fait. Celui de réaliser mes pièces à la main à l'aide de thermoplastique (PCL) et de serflex que l'on avait à disposition.

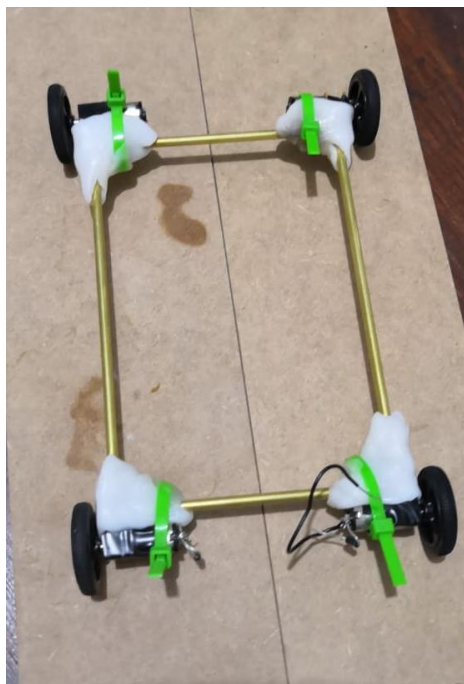


Figure 7 : Montage de la partie inférieure

Cette solution est certes moins élégante mais est tout aussi efficace. Le montage de l'intégralité de la structure a donc été fait rapidement. Le résultat correspond à ce qui avait été imaginé, robuste et simple à démonter.

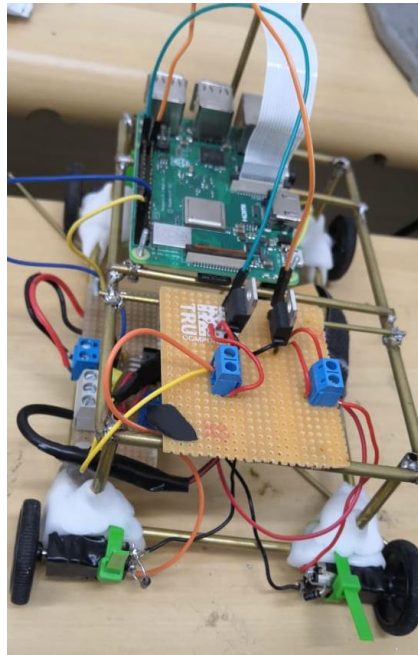


Figure 8 : Nouvelle structure du suiveur de ligne

Suite à une période de tests, plusieurs problèmes d'ordre physique apparaissent. En effet pour pouvoir effectuer des virages, le suiveur doit avoir des écarts de vitesse de rotation entre les deux côtés doivent être important, sans quoi sa trajectoire est trop proche de la ligne droite. J'ai donc modifié les fonctions de commande afin de parer ce problème. Par conséquent, il possède une vitesse de déplacement trop important pour s'adapter au reste du parcours. Ainsi je détermine que l'utilisation de 4 roues rend le robot bien trop rigide et donc pas assez manœuvrable. Je remets donc au cause nos choix fait durant le projet d'utiliser 4 roues. Comment gagner de la manœuvrabilité quitte à perdre de la stabilité ?

D - Restructuration

Je reviens sur notre choix de ne pas utiliser de bille multidirectionnelle. Celui-ci a été fait afin de ne pas avoir trop d'instabilité, ce qui était l'un des problèmes de mes prédécesseurs. Ainsi je décide de remplacer deux roues par une bille. Mais contrairement à l'année dernière, je la place à l'arrière du suiveur. Effectivement, la perte de stabilité est bien moins importante en la plaçant à l'arrière, cela permet de garder un certain contrôle.

Par manque de temps (les temps de livraison étant très long), j'ai dû concevoir ma propre bille multidirectionnelle. Pour ce faire, j'ai entouré une bille en verre par du PCL

(thermoplastique). Le résultat est certes moins efficace que ce que l'on peut trouver sur le marché mais elle suffit pour permettre un fonctionnement correct du suiveur.

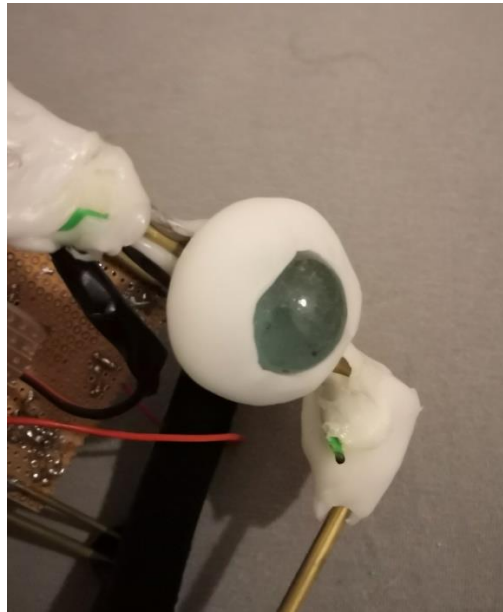


Figure 9 : Bille multidirectionnelle

Une nouvelle batterie de tests a donc été nécessaire. Les résultats sont bien plus satisfaisants que précédemment. Effectivement le robot effectue des virages sans grande vitesse. Malgré tous ces progrès un point est à souligner. Les roues motrices patinent souvent lors des virages. Cela peut être dû à deux choses. La première étant un manque de couple, les moteurs ne sont sûrement pas adaptés à cette nouvelle architecture. La deuxième cause peut être dû à une mauvaise répartition du poids. Ainsi les roues n'adhèrent pas suffisamment au sol.

Un autre problème est aussi à noter. Le temps de réaction est trop lent. Ce qui induit la prise en compte du virage après celui-ci. Deux causes peuvent expliquer ce dysfonctionnement. Un trop grand nombre d'opérations / calculs (superflus) ou un manque d'énergie. En effet, j'ai pu remarquer que le temps de démarrage de la raspberry dépend de l'état de charge de la batterie mais aussi de la façon dont elle est alimentée. J'ai alors dans un premier temps réduit le nombre de calcul en retirant plusieurs éléments d'affichage. De très faibles améliorations ont pu être remarqué, mais n'étaient pas assez significatif pour permettre le bon fonctionnement. J'ai donc pris la décision de rajouter une deuxième batterie, une batterie externe (utilisée pour recharger les téléphones). Les résultats ont été immédiats. De plus celle-ci peut rajouter la masse nécessaire pour corriger le déséquilibre.

Les tests étant satisfaisant, de nouvelles fonctions ont dû être élaborer. Celles-ci prenant en considération les résultats obtenues.

$$Moteur_{Gauche}(angle) = \begin{cases} Si\ angle \leq 50\ alors\ 0 \\ Si\ 50 < angle < 90\ alors\ (\cos((angle - 90) * 4.5) + 1) * 0.05 \\ Si\ 90 \leq angle < 130\ alors\ (\cos((angle - 130) * 4.5) + 1) * 0.05 + 0.1 \\ Si\ angle \geq 130\ alors\ 0.2 \end{cases}$$

$$Moteur_{Droit} = Moteur_{Gauche}(180 - angle)$$

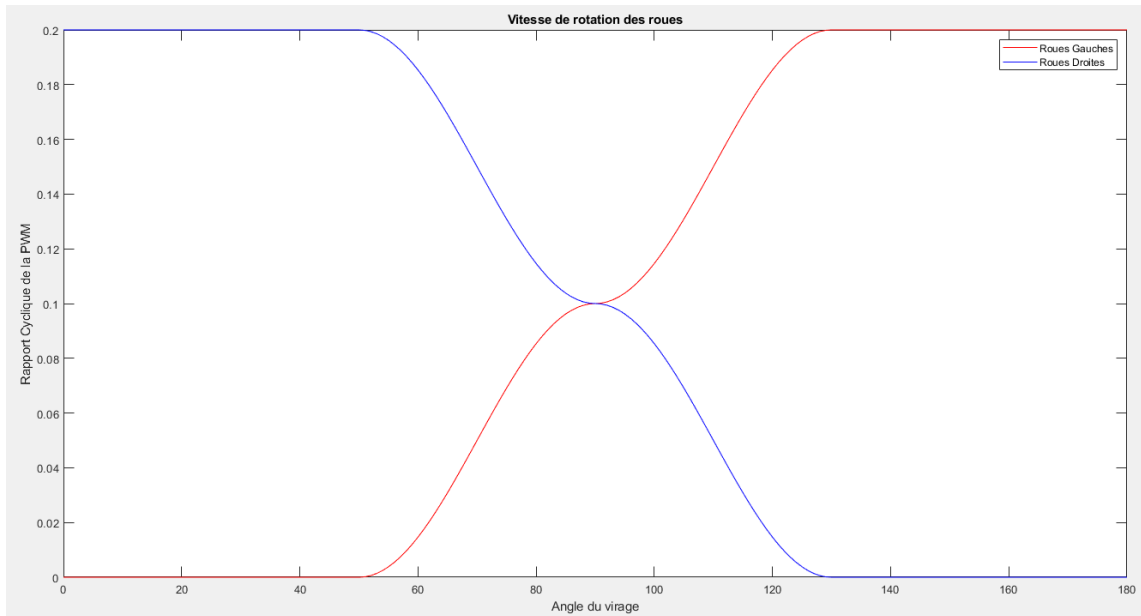


Figure 10 : Courbes de contrôle des moteurs

Une question que l'on m'a alors posée : « Pourquoi faire des courbes et non des droites ? ». Une question légitime et intéressante, mais ayant une réponse simple. L'utilisation de ces courbes permettent d'avoir des tangentes nulles aux points clés. Evitant donc tout à-coup. On peut voir que la puissance délivrée durant les virages est supérieure à celle en ligne droite. Cela est nécessaire car en ligne droite le suiveur utilise les deux moteurs, de plus il ne doit pas aller trop vite afin d'éviter les problèmes de réactivité. N'oublions pas que le but premier est d'effectuer des tours de circuit, le score au chrono est passe alors au second plan.

Par la suite on m'a suggéré d'utiliser des capacités polarisées. Ainsi on peut linéariser le signal, rendant le démarrage des moteurs plus rapide par la décharge des capacités. J'en utilise 2. L'objectif étant d'avoir une valeur de C suffisamment importante tout en minimisant l'impédance. Comme nous pouvons le voir sur la figure X, les valeurs ont un rapport d'un facteur 100.

$$Y_{equ} = Y_{C1} + Y_{C2} = \frac{1}{j\omega(C_1 + C_2)} \sim \frac{1}{j\omega C_1} \text{ (Car } C_1 \gg C_2)$$

$$\frac{1}{R_c} = \frac{R_{C1} + R_{C2}}{R_{C1}R_{C2}} \sim \frac{1}{R_{C2}} \text{ (Car } R_{C1} \gg R_{C2})$$

Nous voyons que l'on a toujours une valeur de capacité équivalente proche de C1 mais que la résistance interne est minimisée par l'ajout de la deuxième capacité C2. A noter que les valeurs des Rc ont une relation avec les valeurs de C. En effet plus C est grand, plus Rc l'est aussi.

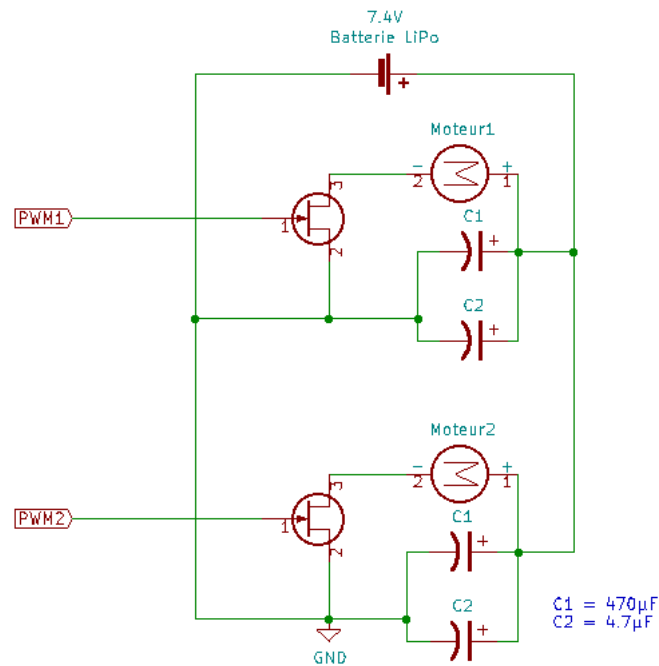


Figure 11 : Schéma du circuit électrique du suiveur de ligne

E - Résultats

Les progrès dans ce projet on était remarquable. J'ai pu passer d'un robot bien trop fragile par sa conception/assemblage et bien trop rigide par ses 4 roues à un robot plus réactif mais surtout bien plus maniable. Mais je me heurte à un problème structurel de taille. Comment avoir de meilleures performances, une plus grande facilité de contrôle. En effet, quelques soucis subsistent, notamment au niveau du contrôle qui doit être minutieux. Ainsi si la puissance apportée au moteur est trop faible celui-ci n'avance pas (ou très peu), si la puissance est trop forte, il sera emporté par son élan et réagira trop tard à l'information reçue. La marge recherchée est dans le cas de ce robot plus faible (entre 15 et 25% pour le rapport cyclique), tout en sachant que celle-ci se déplace en fonction de l'état de charge de la batterie des moteurs. C'est donc par manque de temps que je n'ai pas pu trouver avec précision des fonctions de contrôle optimal (les fonctions trouvées actuellement étant proche de celle recherchées). Le robot n'étant pour le moment incapable de faire le circuit correctement.

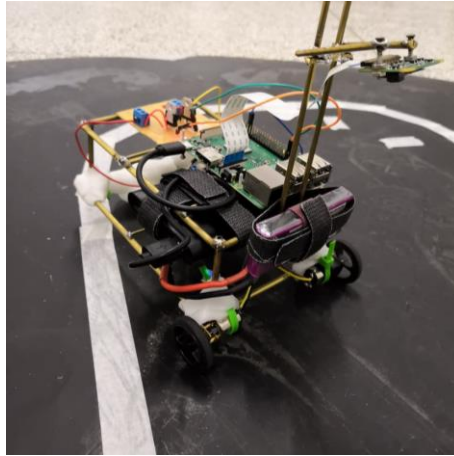


Figure 12 : Version de fin de stage du suiveur

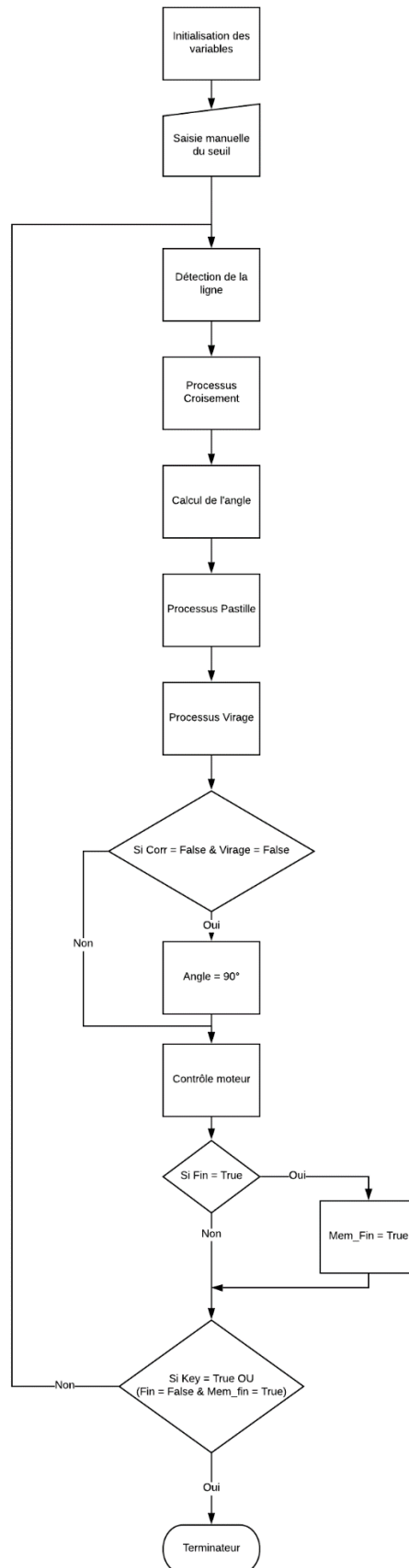
Une solution existe pour augmenter la marge de fonctionnement, Il faut augmenter la largeur du suiveur. Ainsi durant les virages il faudra une vitesse de rotation (des roues) plus grande pour tourner aussi vite qu'avant. La marge est donc plus grande, ce qui nous permet d'utiliser plus de puissance. En ayant une marge d'utilisation plus grande nous augmentons la précision de robot. Cette solution est principalement inspirée du robot suiveur de ligne de la FEG Robotica. Celui-ci est très fin (<5cm de haut) mais est bien plus large. J'ai pu observer une manœuvrabilité et une stabilité supérieure à ce que j'ai pu produire. A noter qu'eux aussi utilise une bille multidirectionnelle à l'arrière avec deux roues motrices à l'avant.

VI – Conclusion

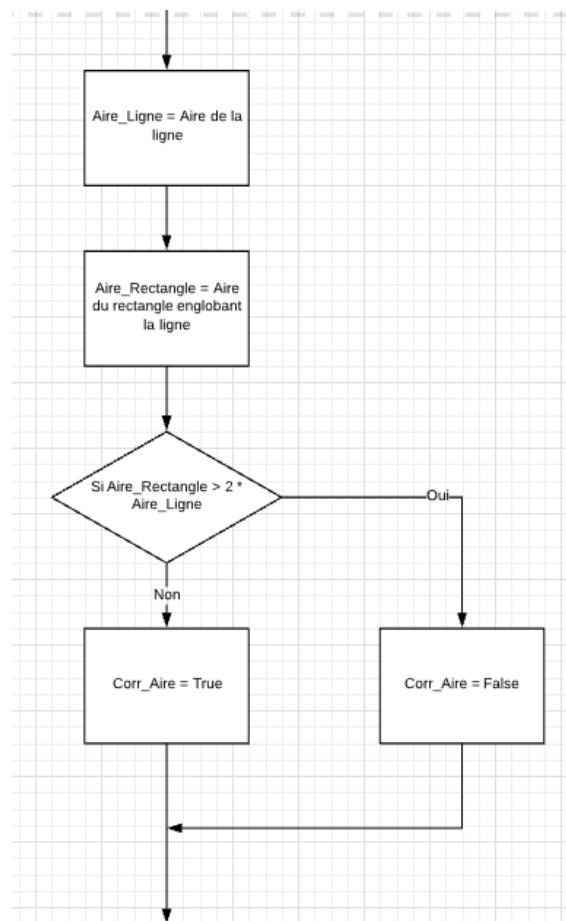
Le stage a été une opportunité incroyable pour apprendre à gérer un projet en autonomie. Être seul sur un projet a été une épreuve de gestion du temps et des ressources. Cela a aussi été une incroyable découverte. En effet j'ai pu découvrir une autre culture, qui semblait semblable à la notre mais qui en réalité est bien différent. J'ai donc pu côtoyer et essayer de comprendre comment ils s'organisent pour réaliser leurs différents projets.

Concernant le robot suiveur de ligne, un certain nombre de problèmes sont survenus. La plupart étaient mécaniques et non anticipés. Beaucoup de temps a été consacré à leur résolution. J'ai pu prendre conscience de l'importance de la gestion du temps, tout particulièrement dans ce genre de situation. L'évolution du suiveur est importante, changeant radicalement de principe de direction. Malgré le manque de temps et de moyen (La FEG Robotica n'ayant pas énormément de matériel à disposition et des temps de livraison très long, supérieur au mois) je n'ai pas pu finir complètement le robot. Ce n'est pas pour autant une défaite. Le nouveau modèle a des performances bien plus impressionnantes que ce que j'ai pu avoir avant. De plus, j'ai pu déterminer les pistes d'amélioration nécessaires à la finalisation du projet. C'est donc sur une demi-victoire que je finis ce stage.

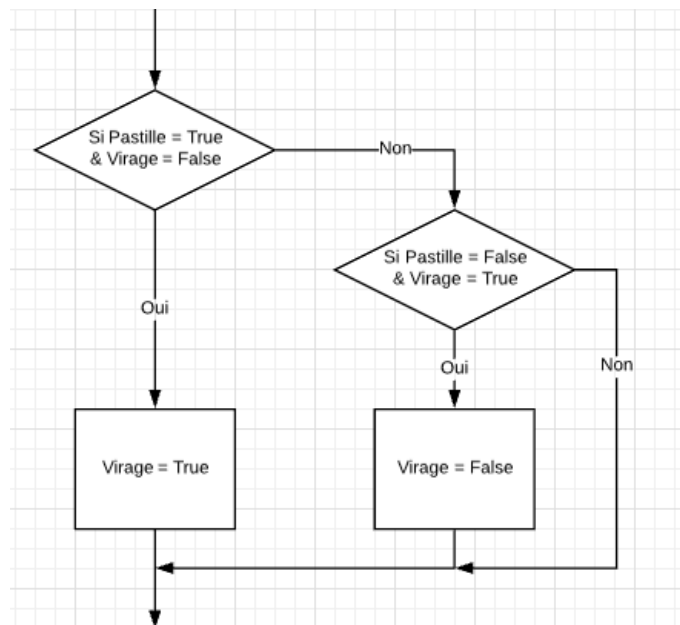
XI – Annexe



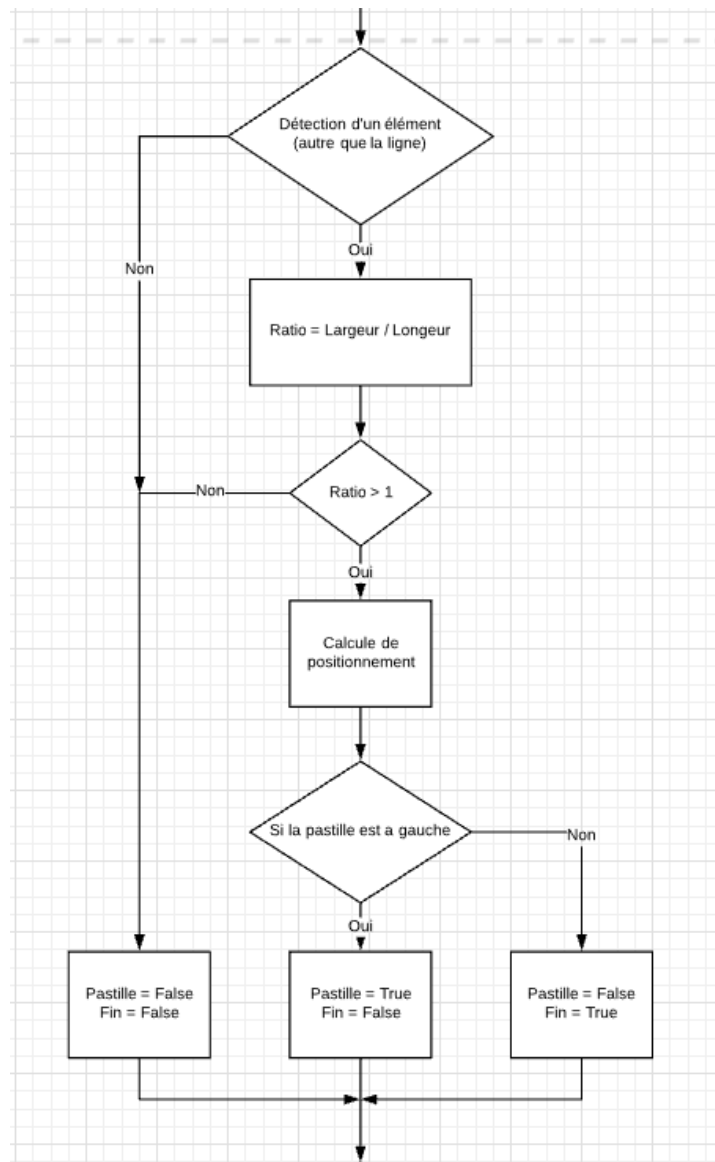
Annexe 1 : Algorithme du programme principal



Annexe 2 : Processus Croisement



Annexe 3 : Processus Virage



Annexe 4 : Processus Pastille

DOS DU RAPPORT

Etudiant : Paul DIGRACI

Année d'étude dans la spécialité : 4^{ème}

Entreprise :
Adresse complète :

FEG Robotica UNESP
Av. Dr. Ariberto Pereira da
Cunha, 333 - Parque Alberto
Bayington,
Guaratinguetá - SP, 12516-410

Téléphone (standard) :
Responsable administratif :
Téléphone :
Courriel :

+55 (12) 3123-2800
Mauro Hugo Mathias - Directeur

Tuteur de stage :
Téléphone :
Courriel :

Agnelo Marotta Cassula

agnelo@feg.unesp.br

Enseignant-référent :
Téléphone :
Courriel :

Eon David

david.eon@univ-grenoble-alpes.fr

Titre : Compétition de Robotique – WINTER CHALLENGE - Conception d'un robot suiveur de ligne par reconnaissance d'image.

Résumé :

Dans la continuité de mon projet universitaire de quatrième année de la filière IESE, je suis allé à Guaratinguetá au Brésil pour participer à des petites compétitions de robotique (remplaçant la Winter Challenge déplacée en septembre). Un important travail a été fait en amont du stage, notamment au niveau de la conception mécanique et électrique. Cela permet de se concentrer sur la programmation et la correction d'inconnues. En effet le robot doit suivre une ligne blanche sur fond noir le plus rapidement possible. Le circuit n'est connu que le jour J et comportant des croisements, virages serrés et zigzag. Différentes technologies permettent de réaliser un suiveur de ligne, la plus courante étant les capteurs infra-rouges. Dans la continuité de l'équipe précédente (équipe de 2018), nous avons utilisé une caméra. L'objectif étant de prévoir les trajectoires en amont, nous permettant d'optimiser sa trajectoire. Ce projet m'a permis d'apprendre des notions qui m'étaient inconnues tel que la reconnaissance d'image (avec l'utilisation de OpenCV), tout en mettant en pratique des éléments connus, notamment la programmation en python et la conception électronique. De plus, j'ai pu travailler en coopération avec l'équipe des robots sumo (de Polytech) et avec les étudiants de la FEG Robotica. Cette expérience est très enrichissante et elle me permet de découvrir comment les brésiliens travaillent sur leurs différents robots.

