

# **Développement d'un robot d'analyse de la locomotion et d'entraînement**

**Mémoire**

**William Bégin**

**Maîtrise en sciences cliniques et biomédicales de l'Université Laval offerte en  
extension à l'Université du Québec à Chicoutimi**  
Maître ès. Sciences (M. Sc.)

Québec, Canada

© William Bégin, 2021

# **Développement d'un robot d'analyse de la locomotion et d'entraînement**

**Mémoire**

**William Bégin**

Sous la direction de :

Martin Lavallière

# Résumé

Que ce soit à des fins d'amélioration de la performance ou de réhabilitation physique, l'analyse de la condition physique et de la locomotion occupe de nos jours une importante place dans le domaine des soins santé. La récente démocratisation des microcontrôleurs et de leur suite de capteurs a permis la venue de nouvelles modalités d'évaluation de la locomotion (p. ex. l'analyse cinématique avec systèmes inertiels). Ce mémoire présente le développement d'un véhicule autonome de petite taille permettant la capture vidéo d'un participant selon un suivi adaptatif utile à des fins d'analyse du mouvement et la réalisation d'évaluation de la condition physique en dehors des milieux de laboratoires et cliniques (p. ex. Salle instrumentée, tapis roulant). Le robot, fabriqué à faibles coûts, est basé sur la plateforme *Raspberry Pi*. De façon à conduire une collecte en milieu écologique, celui-ci est placé sur une piste d'athlétisme intérieure sur laquelle il peut se déplacer de manière autonome dans les corridors en suivant les lignes selon un algorithme de reconnaissance visuelle et d'automatisation. L'instrumentation embarquée du véhicule permet l'évaluation du participant en mode « *suivi* », c'est-à-dire en suivant la cadence en conservant une distance constante et sécuritaire. Le mode *meneur de train* » impose quant à lui une cadence au participant. Ce mode peut aussi être utilisé à des fins d'analyse de la performance ou d'entraînement. Au cours de ce document, les modalités de conception et de fabrication seront présentées. Les méthodes de vision par ordinateur pour la conduite autonome développée pour un ordinateur à faible coût et de faible puissance seront détaillées. Les résultats d'essais effectués sur une piste d'athlétisme intérieure permettant de démontrer les performances et limitations du véhicule sont présentés. Les résultats obtenus sont discutés. Finalement, de nouvelles approches pour l'évaluation de la condition physique seront proposées.

**Mots-clés** : Analyse de la locomotion, réhabilitation physique, contrôle autonome, conduite autonome, vision par ordinateur, Raspberry Pi, microcontrôleurs.

# Abstract

Whether for performance enhancement or physical rehabilitation, the analysis of fitness and locomotion occupies an important place in today's healthcare. The recent democratization of microcontrollers and their suite of sensors has allowed the emergence of new methods for evaluating locomotion (e.g., kinematic analysis with inertial systems). This dissertation presents the development of an autonomous vehicle of small size allowing the video capture of a participant according to an adaptive tracking that could be used for movement analysis and physical condition evaluations performed outside of laboratory environments and clinics (e.g., Instrumented room, treadmills). The robot, manufactured at low cost, is based on the Raspberry Pi platform. In order to conduct a video data acquisition in an ecological environment, it is placed on an indoor running track on which it can move independently in the corridors by following the lane lines according to a visual recognition and automation algorithm. The onboard instrumentation of the vehicle allows the participant to be evaluated in "follow-up" mode, that is to say by following the pace while maintaining a constant and safe distance with the assessed participant. The '*pacesetter*' mode, for its part, imposes a pace on the participant. This mode can also be used for performance analysis or training purposes. Throughout this document, the design and manufacturing methods will be presented. Computer vision methods for autonomous driving developed for a low cost, low-power computer will be detailed. The results of tests carried out on an indoor running track to demonstrate the performances and limitations of the vehicle are presented. Finally, new approaches for fitness assessment will be proposed.

# Table des matières

Résumé .....	ii
Abstract.....	iii
Table des matières .....	iv
Liste des figures, illustrations.....	vii
Liste des tableaux .....	viii
Liste des abréviations, sigles, acronymes .....	ix
Remerciements.....	xii
Avant-propos .....	1
Introduction .....	2
Chapitre 1 Matériel et logiciels.....	6
1.1 Matériel.....	7
1.1.1 Survol du matériel .....	7
1.1.2 Ordinateur de bord .....	7
1.1.3 Générateur de signal.....	8
1.1.4 Détection de la position relative .....	9
1.1.5 Caméra de lecture de ligne .....	10
1.1.6 Caméra d'analyse .....	10
1.1.7 Traxxas 4-TEC 2.0 AWD.....	10
1.1.8 Assemblage .....	11
1.2 Survol de l'assemblage matériel.....	11
1.3 Logiciels et librairies .....	12
1.3.1 Raspberry Pi OS .....	12
1.3.2 Python 3.....	13
1.3.3 NumPy .....	13
1.3.4 OpenCV .....	13
1.3.5 SciPy.....	13
1.3.5 Matplotlib.....	13
1.4 Architecture du projet .....	13
1.4.1 Programme LiDAR.....	14
1.4.2 Programme de la caméra de ligne .....	15
1.4.3 Contrôle de vitesse .....	15
1.4.4 Contrôle de direction .....	15
1.4.5 Boucle de contrôle principale du véhicule .....	15
1.4.6 Programme de publication .....	15
1.4.7 Programmation du guidage.....	16
1.5 Assemblage électrique .....	16
1.6 Environnement d'évaluation .....	16
Chapitre 2 Vision par ordinateur .....	17
2.1 Structure de l'algorithme de vision numérique.....	17
2.1.1 Acquisition.....	17
2.1.2 Traitement.....	18
2.1.3 Discrétisation.....	18
2.2 Mécanique du traitement de l'image.....	18
2.2.1 Conversion du système d'encodage des couleurs.....	18

2.2.2 Déformation de l'image .....	19
2.2.3 Algorithme de tri de lignes.....	20
Chapitre 3 Asservissement .....	22
3.1 Contrôle de la trajectoire médio-latérale.....	22
3.2 Suivi longitudinal de la distance robot/patient.....	23
Chapitre 4 Real Application for a Low-Cost Low-Power Self-Driving 1/10 Scale Car .....	25
4.1 Résumé .....	25
4.2 Abstract .....	25
Introduction.....	26
Objective .....	26
Project Configuration .....	27
Mechanical components .....	27
Electronic components.....	27
Power .....	28
Software and project requirements .....	28
Project architecture .....	29
Computer Vision and Control Systems.....	30
Lane line detection .....	30
Algorithm 1: Line data sorting. ....	32
Steering control .....	32
Throttle control .....	32
Data collection.....	33
Results .....	34
Discussion .....	35
Cost.....	36
Safety .....	36
Conclusion.....	36
Acknowledgements .....	37
References .....	37
Chapitre 5 Résultats .....	39
5.1 Résultats de vision numérique .....	39
5.1.1 Fréquence d'analyse.....	39
5.1.2 Précision de lecture.....	39
5.2 Vitesses et performances .....	42
5.2.1 Caractérisation du comportement du moteur .....	42
5.2.2 Comparaison des résultats de vitesse .....	43
5.3 Résultats de régulation latérale en déplacement.....	43
5.3.1 Résultat du contrôle médio-latérale.....	43
5.3.2 Réglage du régulateur médio-latéral .....	45
5.3.3 Réglage du suiveur longitudinal .....	46
5.4 Évaluations comparative qualitative .....	48
5.4.1 Essais comparatif par vidéo .....	48
Chapitre 6 Discussion .....	49
6.1 Choix du matériel.....	49
6.2 Retour sur la programmation.....	51

6.3 Contrôle .....	51
6.4 Portée dans l'analyse de la condition physique .....	52
6.6 Applications futures .....	53
6.6.1 Test-Léger-Boucher .....	53
6.6.2 Test de marche de 6 minutes (6 Minute Walk Test).....	54
Conclusion .....	56
Bibliographie .....	57
Annexe A Connection au robot via le protocole SSH.....	59
Annexe B Cahier de dessins.....	61
Annexe C Plan du Pavillon sportif de l'UQAC.....	66
Annexe D Montage électrique du circuit .....	67
Annexe E Cinématique du véhicule .....	68
2.1 Modèle de la bicyclette .....	68
2.2 Modèle adapté.....	70
2.2.1 Utilisation du modèle .....	73

# Liste des figures, illustrations

Figure 1 – Version finale assemblée de VAALERIE équipée pour la conduite autonome .....	6
Figure 2 - Raspberry Pi 4, 4 GB RAM (image tirée de <a href="https://www.buyapi.ca/product/raspberry-pi-4-model-b-4gb">https://www.buyapi.ca/product/raspberry-pi-4-model-b-4gb</a> ).....	8
Figure 3 – Le module PCA-9685 d'Adafruit ( <a href="https://learn.adafruit.com/16-channel-pwm-servo-driver/faq">https://learn.adafruit.com/16-channel-pwm-servo-driver/faq</a> ) .....	9
Figure 4 - Le capteur de distance par laser LIDAR-Lite V3 de Garmin ( <a href="https://buy.garmin.com/fr-CA/CA/p/557294">https://buy.garmin.com/fr-CA/CA/p/557294</a> ).....	10
Figure 5 – Diagramme de bloc de l'architecture du matériel électronique du projet .....	11
Figure 6 - Architecture du projet en mode « suivi » du participant. Chaque case représente un objet programmé en Python 3.7. ....	14
Figure 7 - Représentation visuelle de la distribution chromatique selon l'encodage HSV .....	19
Figure 8 - Image originale utilisée pour la calibration.....	20
Figure 9 - Représentation de la dérive $\beta$ de la ligne à 4 m, soit l'angle maximal perçu. ....	21
Figure 10 - Structure du régulateur de direction .....	23
Figure 11 - Erreur perçue en fonction du temps sur un essai statique au centre du corridor .....	40
Figure 12 - Erreur mesurée selon la position en fonction du temps.....	41
Figure 13 - Filtre d'erreur par moyenne mobile sur 5 points. Données brutes en bleu, signal transformé en rouge.....	42
Figure 14 - Diagramme en violon représentant l'influence de l'utilisation d'une moyenne mobile sur 5 valeurs sur la variation de la lecture de la position du véhicule par rapport au centre de la voie comparée aux valeurs réelles sans traitements. ....	42
Figure 15 - Vitesse réelle mesurée en fonction de la commande (pulse) transmise par le système de contrôle. ....	43
Figure 16 - Écart-type de l'erreur positionnelle mesurée en fonction de la vitesse du véhicule. Données recueillies sur 186.7m pour chaque groupe de vitesse avec une fréquence d'analyse à 40 Hz .....	44
Figure 17 - Valeurs individuelles des éléments du contrôleur PID pour le contrôle médio-latéral sur un essai de 186.7 mètres à 6 m/s. ....	46
Figure 18 - Position réelle du véhicule pour un essai réel sur piste.....	46
Figure 19 - Schématisation du suiveur longitudinal .....	47
Figure 20 - Analyse de vitesse d'un patient avec augmentation continue de la vitesse sur 160 m.....	47
Figure 21 - Comparaison des images obtenues lors de tests cliniques et en milieu écologique sur un essai de course à 16 km/h .....	48
Figure 22 - Instructions de vitesse en fonction du temps pour un exemple de test Léger-Boucher réalisé par VAALÉRIE. ....	54
Figure 23 - Heat map de la vitesse d'un patient en m/s lors d'un tour de piste de 180 m sur un test de marche de 6 minutes. ....	55

## Liste des tableaux

Tableau 1 - Table des propriétés des différents véhicules de petite taille capable de conduite autonome. Tiré de Openbot: Turning Smartphones into Robots (Müller & Koltun, 2020). .....	5
Tableau 2 - Liste des composants requises pour VAALERIE .....	7
Tableau 3 - Spécification du Raspberry Pi 4 4 GB RAM.....	8
Tableau 4 - Liste des librairies utilisées et leur version.....	12

## Liste des abréviations, sigles, acronymes

CM	Centre de masse
CSI	Camera Serial Interface
CV	Computer Vision
eMMC	Embedded MultiMedia Card
EEPROM	Electrically-erasable Programmable Read-only Memory
GB	Gigabit
HSV	Hue Saturation Value
I2C	Inter-Integrated Circuit
IA	Intelligence artificielle
LiDAR	Light Distance and Ranging
ML	Machine Learning
OS	Operating System
PD	Proportionnel Dérivé
PID	Proportionnel Intégral Dérivé
PWM	Pulse Width Modulation
ROI	Region of Interest
ROS	Robot Operating System
RP4	Raspberry Pi 4
RPOS	Raspberry Pi Operating System
SBC	Single Board Computer
SE	Système d'exploitation
SSH	Secure Shell
UQAC	Université du Québec à Chicoutimi
VAALÉRIE	Véhicule autonome d'analyse de la locomotion équipé d'un référentiel intelligent embarqué



*« I've seen things you people wouldn't believe. Attack ships on fire off the shoulder of Orion. I watched C-beams glitter in the dark near the Tannhäuser Gate. All those moments will be lost in time, like tears in rain. Time to die. »  
(Rutger Hauer as Roy Batty, Blade Runner)*

# Remerciements

J'aimerais tout d'abord remercier mon directeur, M. Martin Lavallière pour son support, mais aussi pour sa latitude dans mes idées folles (et parfois les siennes). La direction de qualité offerte lors des deux dernières années aura permis d'achèvement d'un travail dont je suis fier et dont j'en retire une connaissance dépassant largement mes espérances. Un remerciement tout spécial doit être également offert à mon collègue et ami de longue date Steven Duquette sans qui la programmation de ce projet aurait probablement une forme bien différente aujourd'hui. De plus, je souhaiterais profiter de ces lignes pour remercier mes parents, Corinne et François, qui m'ont offert le cadeau parfois empoisonné de la curiosité et du devoir accompli. Votre support s'est fait ressentir à chaque instant de ce fabuleux projet.

# Avant-propos

Ce mémoire est composé d'un article rédigé et publié lors de mes études de maîtrise en sciences clinique et biomédicale réalisées à l'Université du Québec à Chicoutimi. Il est présenté dans sa version publiée dans une revue scientifique revue par les pairs. Dans cet article, j'ai effectué la préparation des montages expérimentaux, effectué les tests sur le terrain et procédé à l'extraction des données en plus d'effectuer les analyses statistiques. J'ai de plus effectué la rédaction de cet article en tant que premier auteur. Le manuscrit initial et sa version finale ont ensuite été révisés par tous les auteurs du papier.

Article 1 : Bégin, W., Duquette, S., & Lavallière, M. (2020). *Real Application for a Low-Cost Low-Power Self-Driving 1/10 Scale Car*. *Journal of Robotics and Automation*, 4(1), 195-201. doi:10.36959/673/370

# Introduction

L'analyse de la locomotion est un élément primordial que ce soit à des fins de réadaptation, d'amélioration de la performance ou de développement d'équipement sportif et clinique. Comme la locomotion engendre implicitement un déplacement, il est à ce jour toujours aussi difficile d'acquérir de l'information sans en modifier sa nature. Au cours des dernières décennies, le monde de la recherche a vu naître de nombreux outils permettant de quantifier la locomotion avec moins d'impact. Plusieurs approches différentes permettent de quantifier la locomotion humaine. L'une d'elles consiste en l'évaluation cinématique dans laquelle on étudie les composantes en position, vitesse et accélération de points fixes du corps humain. De nombreux systèmes sont à ce jour disponibles permettant la réalisation d'évaluations cinématiques de la locomotion. Parmi ceux-ci, on compte la présence de systèmes optoélectroniques. Ces systèmes consistent en un volume de contrôle instrumenté à l'aide d'une matrice de caméras infrarouges. Ces caméras, équipées d'émetteurs infrarouges en périphérie de la lentille permettent la captation en temps réel du mouvement de réflecteurs sphériques de petite taille placés sur le(s) patient(s) dans la zone d'évaluation. Une approche logicielle permet le suivi de ces marqueurs sous forme de matrices quadridimensionnelles  $(x, y, z, t)$  pouvant être réinterprétées pour des analyses ultérieures. Ces systèmes ont l'avantage d'être extrêmement précis (Merriault, Dupuis, Boutteau, Vasseur, & Savatier, 2017) et d'être peu invasifs pour le patient. En utilisant les données recueillies avec ces appareils, il est possible d'obtenir les diagrammes de phases aux différentes articulations dans le plan ou dans l'espace conformément aux travaux réalisés par de nombreux chercheurs (Lamine, Bennour, Laribi, Romdhane, & Zaghloul, 2017). Néanmoins, ces systèmes sont coûteux et limitent le domaine d'analyse à l'environnement dans lequel ils sont installés. Par ailleurs, à moins d'être jumelée à un autre appareil, cette approche ne fournit aucune information cinétique sur le patient en évaluation. Veilleux, Raison, Rauch, Robert, and Ballaz (2016) ont démontré la faisabilité de l'utilisation combinée d'un tel outil avec des appareils de mesures et une programmation efficace afin de quantifier le déplacement spatio-temporel et les réactions aux appuis lors d'une marche dite normale.

Plus récemment, l'utilisation de technologies portables comme les systèmes à centrales inertielles ont permis la mise en pratique de collectes cinématiques à l'extérieur des murs des laboratoires et des cliniques. Validées par la littérature (Lebel, 2017), les centrales permettent la décomposition du mouvement en réseau (similairement aux points d'évaluations des systèmes optoélectroniques infrarouges présentés plus haut), lorsqu'utilisées en groupes aux articulations. La miniaturisation des systèmes embarqués des deux dernières décennies aura grandement aidé à l'acceptation clinique de ces outils. Permissif sur le plan de la zone d'évaluation, l'utilisation de réseau de centrales inertielles demande toutefois une lourde tâche en ce qui a trait à la programmation. Bien qu'aucune publication n'ait été effectuée à ce sujet par ceux-ci, le groupe de recherche Ing Réadapt ULaval planche actuellement au développement d'un système mettant à profit les centrales

inertielles permettant ainsi l'analyse sportive en milieu écologique et l'évaluation de la condition motrice en réadaptation pour une vaste clientèle (<https://www.youtube.com/watch?v=z94avivk4Q8&t=1s>). Toutefois, Cuesta-Vargas, Galán-Mercant, and Williams (2010) ont démontré que l'utilisation de ces systèmes pour des évaluations de mouvements plus complexes comme la marche induit une plus grande incertitude que pour des mouvements plus simples en raison du nombre de capteurs devant se positionner adéquatement sur les participants.

L'augmentation drastique des performances des ordinateurs et de la popularité de l'intelligence artificielle (IA), plus particulièrement l'apprentissage machine (AM) (considéré comme une sous-catégorie d'IA), depuis la dernière décennie a permis le développement de nouvelles modalités d'analyses dans le domaine de l'évaluation physique. Toshev and Szegedy (2014), entre autres, ont démontré la faisabilité de l'utilisation d'un réseau de neurones convolutif afin d'estimer la position segmentaire d'une personne sans l'utilisation de marqueurs placés au préalable sur une personne en déplacement. Cette approche non invasive a permis la classification correcte, mais non quantifiée des différents membres du corps. D'autre part, la méthode itérative appliquée à un réseau convolutif avancée par Carreira, Agrawal, Fragkiadaki, and Malik (2016) a permis la définition correcte en pourcentage des points clés (de l'anglais Percentage of Correct Keypoints of humanpose – PCKh) allant de 66.9 % pour les chevilles et 95.5 % pour la tête. Par ailleurs, comme démontré par Ramakrishnan, Kim, and Reed (2019), des images de surveillances en milieux écologiques peuvent être mises à profit afin de reconnaître des patrons de marche atypiques. Dans le secteur privé, Proplayai (PitchAI, Hamilton, Canada) propose des évaluations par reconnaissance visuelle spécifique aux joueurs de baseball à des fins d'entraînement. L'avantage premier de cette approche est l'absence totale de contact avec le patient, évitant ainsi toute modification au patron de locomotion.

Une seconde approche à l'évaluation, soit l'approche cinétique, tend à décrire la locomotion via la quantification des forces et moments présents dans le patron locomoteur. Plusieurs outils allant de surfaces instrumentées aux tapis de course sensibles à la pression permettent de recueillir des informations spatiales, physiques et temporelles sur la marche dans le but d'en déterminer ses paramètres cinétiques. Webster, Wittwer, and Feller (2005), entre autres, ont démontré la validité et l'utilité d'un de ces outils dans un cadre d'analyse clinique. Bien que ces outils soient établis depuis plusieurs années dans certains milieux cliniques et de recherche pour l'analyse de la démarche, ceux-ci sont dispendieux et leur utilisation est limitée à des espaces d'analyses restreints (Zhou & Hu, 2008). De nouveaux produits plus portables rendent désormais possible la réalisation d'évaluation en milieux plus écologiques (ex. en milieu de travail). Leur utilisation est toutefois limitée à la longueur des surfaces d'analyses elles-mêmes, ne permettant donc pas l'obtention de modèles déployés sur une distance excédant la longueur du tapis. Par ailleurs, les systèmes actuels ne permettent que la collecte bidimensionnelle de la locomotion en évaluant les points de contact au sol.

L'approche cinétique et cinématique ne présentent pas à elles seules une finalité quant à la description de la locomotion humaine et doivent généralement être utilisées conjointement afin de réaliser une étude complète du mouvement. À ce jour, la littérature fait état de peu de solutions viables (Müller & Koltun, 2020) capables d'effectuer l'analyse terrain de la locomotion, c'est-à-dire dans des environnements moins restreints que ceux offerts par les laboratoires et les cliniques.

De cette brève revue, deux constats peuvent être tirés. Dans un premier temps, les systèmes actuels sont principalement utilisables dans des environnements contraignants comme les laboratoires de recherche et les cliniques, limitant ainsi la portée des collectes de données à l'extérieur de ces milieux. Par ailleurs, de nombreux systèmes comme les tapis roulants instrumentés ne permettent généralement pas la réalisation d'évaluations sur une clientèle pathologique ou vieillissante. Le second constat est que bon nombre de systèmes nécessitent une approche pouvant perturber le patron de locomotion par l'utilisation de centrales de capteurs et de réflecteurs devant être placés sur les participants.

Ces constats tendent à démontrer que les méthodes actuelles rendent l'évaluation de la condition physique dans des cadres réels à l'extérieur des laboratoires ne modifiant pas les patrons de locomotion sont déficientes. Dans le but de pallier cette problématique, un projet de développement d'un outil permettant l'évaluation de la locomotion à l'extérieur des murs des laboratoires a été mis sur pieds. Le cadre de ce projet est de réaliser un outil capable de recueillir des données cinématiques d'un participant sans interférer physiquement avec son patron locomoteur. Le projet a également pour objectif de mener des évaluations de clientèle pathologiques dans un environnement sécuritaire. Le projet doit être également réalisé à faible coût.

De manière à concentrer la portée de cette recherche, l'environnement de contrôle est limité aux pistes d'athlétisme puisque celles-ci permettent la réalisation d'évaluation sur des distances prolongées similairement à un tapis roulant sans toutefois induire les notions de sécurité qui leur sont rattachés. Dans le but de limiter au maximum la modification des patrons de locomotions lors des évaluations, l'approche préconisée dans ce projet est la capture vidéo permettant une analyse par reconnaissance visuelle. Plutôt que de fixer une caméra pour recueillir les informations visuelles désirées à un point sur la piste, basé ainsi selon un référentiel global, le point de référence de la caméra est fixé sur le participant (un référentiel mobile). Un système mobile autonome de type suiveur est donc développé. Le véhicule électrique de petite taille, appelé V.A.A.L.É.R.I.E. (Véhicule Autonome d'Analyse de la Locomotion Équipé d'un Référentiel Intelligent Embarqué) développé dans le cadre de ce projet permet la réalisation de cette tâche. Basé sur une plateforme de véhicule radiocommandé, le véhicule est modifié et instrumenté d'un ordinateur de bord, d'un capteur de distance et de caméras afin de suivre, précéder ou imposer un pas de déplacement aux participants. V.A.A.L.É.R.I.E. utilise comme références latérales les lignes de la piste d'athlétisme reconnues par un algorithme de reconnaissance visuelle pour se diriger en conservant une trajectoire au centre du corridor. Un capteur de distance est utilisé pour assurer un suivi de distance constant du participant en évaluation. Une caméra placée sur le véhicule permet l'acquisition

d'images pouvant être utilisées pour l'analyse de la locomotion. Le véhicule est également utilisé afin d'imposer le pas de course, lui permettant d'être utilisé dans une série de tests, dont le test Léger-Boucher) durant lesquels la vitesse doit être contrôlée. De nombreux projets ayant pour but l'atteinte de la conduite autonome sont décrits dans la littérature. Dans le cadre de leurs travaux, Müller and Koltun (2020) ont établi une liste exhaustive des capacités, coûts et performances de différents véhicules autonomes formant une catégorie dans laquelle le projet VAALÉRIE peut s'inscrire. Le **Erreur ! Source du renvoi introuvable.**, tiré de leurs travaux à libre accès, présente les différents véhicules développés dans les dernières années par différents groupes de recherches et organisations.

Tableau 1 - Table des propriétés des différents véhicules de petite taille capable de conduite autonome. Tiré de Openbot: Turning Smartphones into Robots (Müller & Koltun, 2020).

Platform	Retail Cost	Setup Time [h]	Size [cm]	Weight [kg]	Speed [m/s]	Battery [min]	Actuation	Odometry	Camera	LiDAR	IMU	GPS	WiFi	Bluetooth	3G/4G/5G	Speaker	Microphone	Display	Compute	Ecosystem
AutoRally [12]	\$10,000	100	100x60x40	22	25	20+	BLDC+Servo	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	Mini-ITX PC	ROS
F1/10 [25]	\$3600	3	55x30x20	4.5	18	20+	BLDC+Servo	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	Jetson TX2	ROS
RACECAR [19]	\$3400	10	55x30x20	4.5	18	20+	BLDC+Servo	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	Jetson TX1	ROS
BARC [13]	\$1000	3	54x28x21	3.2	-	20+	BLDC+Servo	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	Odroid XU-4	ROS
MuSHR [36]	\$900	3	44x28x14	3	11	20+	BLDC+Servo	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	Jetson Nano	ROS
DeepRacer [1]	\$400	0.25	-	-	6	15+	BDC+Servo	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	Intel Atom	Custom
DonkeyCar [31]	\$250	2	25x22x12	1.5	9	15+	BDC+Servo	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	Raspberry Pi	Custom
Duckiebot [28]	\$280	0.5	-	-	-	-	2xBDC	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	Raspberry Pi	Custom
Pheeno [38]	\$270	-	13x11	-	0.42	300+	2xBDC	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	ARM Cortex-A7	Custom
JetBot [24]	\$250	1	20x13x13	-	-	-	2xBDC	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	Nvidia Jetson	Custom
Create-2 [7]	\$200	-	34x34x9	3.6	-	-	2xBDC	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	Custom
Thymio II [29]	\$170	-	11x11x5	0.46	0.14	-	2xBDC	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	Microcontroller	Custom
AERobot [34]	\$20	0.1	3x3x3	0.03	-	-	2xVibration	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	Microcontroller	Custom
<b>OpenBot</b>	\$50*	0.5	24x15x12	0.7	1.5	45+	4xBDC	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	Smartphone	Android

Table 2: **Robots.** Comparison of wheeled robotic platforms. Top: Robots based on RC trucks. Bottom: Navigation robots for deployment at scale and in education. "–" indicates that no information is available. \* The cost of the smartphone is not included and varies.

Les objectifs de ce projet sont de démontrer la faisabilité d'un outil de collecte de données pour la locomotion à faible coût, réaliser un algorithme de conduite autonome limitant la saccade et d'obtenir un système suffisamment robuste pour permettre une collecte vidéo à l'extérieur des milieux cliniques usuels. Les travaux de réalisation de ces objectifs sont présentés dans ce mémoire. L'hypothèse première de ce projet est que les composantes électroniques sélectionnées seront suffisamment performantes pour maintenir une conduite autonome sur la piste d'athlétisme pour une gamme de vitesse permettant ainsi l'analyse d'une vaste population par le biais d'une multitude de tests.

# Chapitre 1 Matériel et logiciels

La réalisation du projet V.A.A.L.E.R.I.E. se base sur l'utilisation d'une multitude d'éléments principalement libre d'accès à faibles coûts et facilement retrouvables sur le marché. Le robot est basé sur certaines recommandations de la communauté F1TENTH (<https://f1tenth.org/>). La totalité des logiciels et bibliothèques utilisés dans le cadre du projet sont en libre accès. Ce chapitre détaille les composantes électroniques, mécaniques et logicielles du robot. Toutes les composantes utilisées permettent le contrôle autonome du véhicule en temps réel sans l'appui d'un effort de calcul externe (cloud computing). Tout le nécessaire pour la fabrication du véhicule est présenté en Annexe B Cahier de dessins

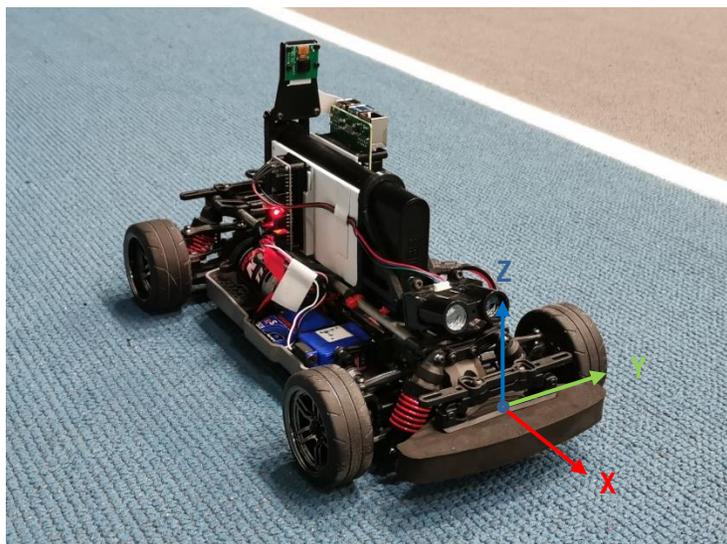


Figure 1 – Version finale assemblée de VAALERIE équipée pour la conduite autonome

## 1.1 Matériel

Plusieurs composantes sont nécessaires pour arriver à la conduite autonome d'un véhicule. La sélection de celles-ci revêt une importance toute particulière en raison des objectifs économiques et de performance du projet. La prochaine section décrira exhaustivement les composantes électroniques sélectionnées lors de ce projet.

### 1.1.1 Survol du matériel

Tableau 2 - Liste des composantes requises pour VAALERIE

Électronique	Qt.	Type
Raspberry Pi 4 avec 4 GB RAM	1	SBC
Kingston CANVAS Select Plus 100 Mo/s C10 – 32 GB	1	Carte SD
Garmin LIDAR-Lite V3	1	LiDAR
Adafruit PCA-9685	1	Pilote PWM/Servomoteur
Raspberry Pi Camera Board v1.3	1	Caméra
Garmin VIRB 360	1	Caméra
RAVPower 26800 mAh USB-C 3-ports	1	Bloc d'alimentation
Mécanique		
Traxxas 4-TEC 2.0 AWD	1	Véhicule télécommandé
Adaptateur en PLA	2	Support sur véhicule
Tige en acier 6 mm x 230 mm	2	Support sur véhicule
Collet en PLA	2	Support pour bloc
Bras d'attache en PLA	2	Pour capteur LiDAR
Chandelle de support en PLA	1	Support pour LiDAR
Adaptateur pour ventilateur en PLA	1	Pour ventilation RPi 4
Autres		
Ordinateur portable sur Windows 10 ou Linux	1	Contrôle et collecte
Traxxas 2S 5800 mAh LiPo	2	Piles pour véhicule
Câblage de type Dupont	~20	Câbles
AmazonBasics USB Type-C vers USB Type-C	1	Câble d'alimentation
Tube thermorétractable	~	Pour câblage

### 1.1.2 Ordinateur de bord

Dans le cadre de ce projet, le Raspberry Pi 4 (RP4), présenté à la Figure 2, édition 4 gigabits (GB) de RAM, est utilisé à titre de module de contrôle principal. Il assure toutes les fonctions de contrôle et d'analyse en temps réel. Le module initialement sélectionné était son homologue, l'Asus Tinkerboard S. Le Raspberry Pi 4 n'étant pas offert à l'époque, cet ordinateur se voulait l'alternative la plus performante sous le format SBC en raison de la présence de mémoire embarquée (eMMC) et de sa mémoire vive de 2 GB. Bien que beaucoup plus performant en termes de capacités d'opérations par seconde, le module Nvidia Jetson TX2, alors déjà disponible, n'a pas été sélectionné en raison de son coût élevé (250 \$ vs 75 \$ pour le RP4), de ses performances considérées excessives pour les besoins du projet et de sa gourmandise énergétique. Le RP4 est un ordinateur de type *Single Board Computer* (SBC) relativement performant. Il est basé sur une architecture de processeur ARM v8 64-bit à quatre cœurs. Le processeur permet une fréquence d'exécution à 1.5GHz et est équipé d'une

mémoire vive LPDDR4-3200. Le système d'exploitation (SE) est opéré à partir d'une carte mémoire de type SD ce qui rend l'appareil légèrement moins rapide que le Tinkerboard S. Bien que peut-être plus adapté en raison de ses performances, le Tinkerboard a finalement été abandonné en raison de bris répétés, de manque d'assistance technique et de la difficulté à fournir la puissance requise via le port micro-USB d'alimentation sur le SBC via un système portable.

Tableau 3 - Spécification du Raspberry Pi 4 4 GB RAM

Raspberry Pi 4	I/O	Alimentation
Broadcom BCM2177, Quad core Cortex-A72 64-bit SoC	Mini-HDMI x2	5 V CC via USB-C
4 GB LPDDR4-3200 SDRAM	USB 3.0 Type A	5 V CC via Ethernet
H.265 (encodage) et H.264 (encodage et décodage)	USB 2.0 Type A	
2.4 GHz et 5 GHz IEEE 802.11ac sans fils	UART, SPI, I2C, GPIO	
Bluetooth 5.0	Carte SD	
	Port MIPI DSI 2-voies	
	Gigabit Ethernet	



Figure 2 - Raspberry Pi 4, 4 GB RAM (image tirée de <https://www.buyapi.ca/product/raspberry-pi-4-model-b-4gb>)

### 1.1.3 Générateur de signal

Afin d'assurer la conduite du véhicule, le Traxxas 4-TEC 2.0 AWD est muni d'un pilote de moteur permettant l'alimentation et le contrôle du moteur CC à brosse ainsi que d'un servomoteur utile au contrôle du système de direction. Ces deux éléments sont contrôlés à partir de la modulation d'impulsions électriques appelées *Pulse Width Modulation* (PWM) générant un voltage apparent. L'interface du Raspberry Pi 4 permet la génération de PWM dit logiciel, mais ceux-ci sont trop imprécis pour fournir un contrôle adéquat sur les servomoteurs sans générer des perturbations. Il existe sur le marché bon nombre d'alternatives permettant la génération de PWM suffisamment précis pour le contrôle de servomoteurs à partir d'un SBC. Le pilote PCA-9685 d'Adafruit a été choisi dans le cadre du projet. Comparativement à d'autres options, le PCA-9685 est peu coûteux (16 \$) et est

compatible avec la tension disponible sur les broches du Raspberry Pi 4. Le PCA-9685 peut être contrôlé directement par le Raspberry Pi via un protocole I2C et peut être alimenté par l'ordinateur sans source électrique externe si la demande énergétique demeure faible. Le module demande une alimentation électronique de 3.3 V CC ainsi qu'une alimentation de contrôle de 5 V CC pouvant aller jusqu'à 6 V CC. Il offre une résolution de 12 bits permettant de moduler des pulses avec une résolution de 4  $\mu$ s à une fréquence de rafraîchissement de 60 Hz. Sa conception permet également l'ajout en parallèle d'autres éléments utilisant le protocole de communication I2C. La Figure 3 présente une photo du module utilisé.

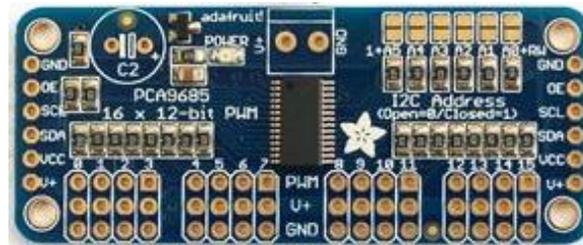


Figure 3 – Le module PCA-9685 d'Adafruit (<https://learn.adafruit.com/16-channel-pwm-servo-driver/faq>)

#### 1.1.4 Détection de la position relative

Afin d'assurer la détection continue de la position relative d'un participant par rapport au robot un *Light Detection And Ranging* (LiDAR) est utilisé. Un LiDAR est un dispositif de mesure permettant d'estimer la distance entre un émetteur et un élément placé devant la trajectoire d'analyse. Cette technologie mesure le temps entre l'envoi d'un faisceau lumineux et de retour après que celui-ci ait été réfléchi par un objet. La vitesse de la lumière étant fixe, il est ainsi possible d'estimer la position de l'objet ayant réfléchi la lumière. Le capteur de distance photoélectrique sélectionné dans le cadre de ce projet est le Garmin LIDAR-Lite V3, un module de détection directionnel (<https://buy.garmin.com/fr-CA/CA/p/557294>). Ce capteur a été choisi en raison de son prix abordable (175 \$), de la disponibilité d'une librairie *Python* en libre accès et de sa compatibilité avec le protocole de communication I2C déjà utilisé pour la génération des PWM utiles au contrôle du moteur et de la direction via le PCA-9685.



Figure 4 - Le capteur de distance par laser LIDAR-Lite V3 de Garmin (<https://buy.garmin.com/fr-CA/CA/p/557294>)

### 1.1.5 Caméra de lecture de ligne

Le projet comporte une caméra uniquement utile à la reconnaissance de ligne afin d'assurer la régulation autonome du robot mobile. La caméra choisie pour l'évaluation des lignes de piste est la Raspberry Pi Camera Board v1.3. Cette caméra permet une résolution maximale de 1920 x 1080 pixels (1080p) à 30 images par seconde. Il est toutefois possible d'augmenter la fréquence d'acquisition en diminuant la résolution de l'image. Elle est équipée d'un capteur Omnivision 5647 de 5 MP. La caméra communique avec le RP4 à l'aide des broches MIPI prévues à cet effet.

### 1.1.6 Caméra d'analyse

Afin de recueillir les images utiles à l'analyse du mouvement, le véhicule a été équipé d'une seconde caméra à capture de 360 degrés. Le modèle sélectionné est la Garmin VIRB 360. Cet appareil permet l'enregistrement vidéo à une résolution de 4096 x 2160 pixels (4K) en stabilisation sphérique avec une fréquence de capture à 30 images par seconde. L'appareil peut être utilisé conjointement à une application permettant l'acquisition simultanée de données physiologiques obtenues avec un appareil de mesure comme une montre d'entraînement instrumentée. Les séquences vidéo obtenues ont été traitées sommairement à l'aide de l'application VIRB Edit afin de modifier le format des images obtenues à une taille plus usuelle de 16 :9.

### 1.1.7 Traxxas 4-TEC 2.0 AWD

La plateforme de véhicule sélectionnée dans le cadre de ce projet est le châssis 4-TEC 2.0 à 4 roues motrices de Traxxas (McKinney, Texas, É-U). Le châssis est équipé d'une motorisation électrique à courant continu de 7.4 volts (Traxxas Titan 12T), d'un régulateur de moteur (Traxxas XL-5 ESC) et d'un servomoteur à haut couple (Traxxas 2075). La motorisation aux quatre roues est obtenue au moyen d'un arbre de transmission relié au système de transmission du moteur. Sous certaines conditions, le véhicule peut atteindre une vitesse maximale de 48 km/h (13.33 m/s).

### 1.1.8 Assemblage

Afin de sécuriser les équipements sélectionnés au véhicule, une multitude de pièces d'attache ont été dessinées sur Onshape (Version 1.133) et produites par impression 3D à l'aide d'une imprimante de marque Creality (Modèle Ender 3). Les différentes pièces réalisées sont présentées dans le cahier de dessin de l'Annexe B Cahier de dessins.

## 1.2 Survol de l'assemblage matériel

Cette section présente un survol de l'assemblage matériel du projet. L'appareil de mesure de distance, le LiDAR, est branché au Raspberry Pi 4 via les broches GPIO. La communication entre les deux appareils est établie grâce au protocole I<sup>2</sup>C. La caméra utilisée pour la conduite autonome est quant à elle connectée à l'ordinateur par le port CSI disponible sur le RP4. Afin de lancer l'application, mettre fin à un essai et recueillir des données comportement, le RP4 travaille conjointement avec un ordinateur portable (Windows ou Unix). La communication entre les deux ordinateurs est établie via le protocole Secure Shell (SSH) sur un réseau Wi-Fi privé émis par le RP4. La connexion permet un transfert bidirectionnel d'informations entre les deux appareils. L'Annexe A Connexion au robot via le protocole SSH présente le protocole de connexion des deux appareils. L'alimentation électrique du RP4 est assurée par un câble USB-C branché au port d'alimentation prévu à cet effet. L'énergie fournie provient d'un bloc de piles externe de 26 800 mAh (5 V). Le bus I<sup>2</sup>C utilisé pour le LiDAR permet également une communication à sens unique du RP4 au pilote PCA-9685. Ce pilote est utilisé afin de fournir les informations aux différents organes de contrôle par le biais d'un signal de modulation de largeur d'impulsions (PWM). La Figure 5 présente l'architecture simplifiée de l'assemblage du robot.

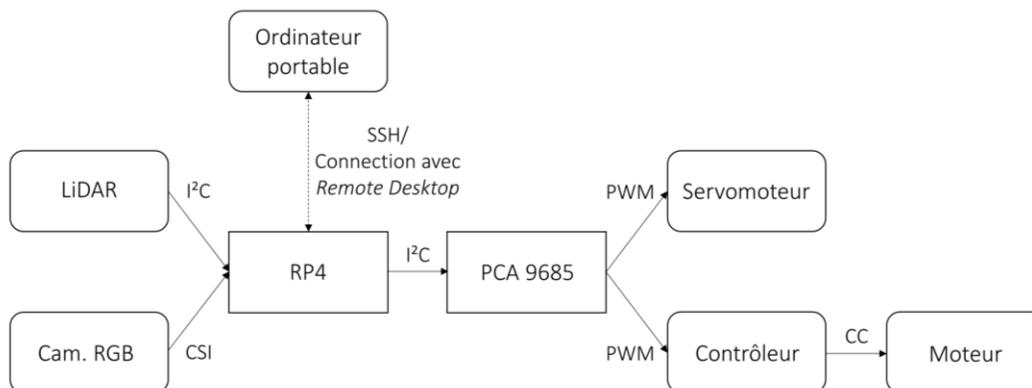


Figure 5 – Diagramme de bloc de l'architecture du matériel électronique du projet

## 1.3 Logiciels et bibliothèques

Afin d'atteindre l'objectif de conduite autonome du projet, une panoplie de modules, bibliothèques et logiciels ont dû être mis à profit. La prochaine section fait état et décrit les éléments cruciaux nécessaires à la réalisation de l'autonomie. Le Tableau 4 présente la liste alphabétiquement ordonnée des bibliothèques utilisées ainsi que leur version téléchargée lors de la réalisation de la programmation.

Tableau 4 - Liste des bibliothèques utilisées et leur version

Libraries	Version
Adafruit-Blinka	4.7.0
Adafruit-GPIO	1.0.3
Adafruit-PCA9685	1.0.1
Adafruit-PlatformDetect	2.8.0
Adafruit-PureIO	1.1.5
DateTime	4.3
adafruit-circuitpython-busdevice	4.3.1
adafruit-circuitpython-lidarlite	1.2.1
cycler	0.10.0
getch	1.0
key	0.4
matplotlib	3.2.1
numpy	1.17.4
opencv-python	4.1.0.25
pip	19.0.3
pyserial	3.4
python-dateutil	2.8.1
pyusb	1.0.2
scipy	1.4.1
setuptools	40.8.0
picamera	1.13

### 1.3.1 Raspberry Pi OS

Raspberry Pi OS (RPOS) est le système d'exploitation (SE) officiellement supporté par la Fondation Raspberry Pi. Le nom original, reconnu lors de l'installation dans le cadre de ce projet était Raspbian. À des fins de simplifications, le nouveau nom sera utilisé au cours de ce document. RPOS est une variante gratuite et libre accès de Debian, un SE actuellement basé sur un noyau Linux. Il a été développé et optimisé pour l'architecture du Raspberry Pi. À son installation, le SE comprend une suite de logiciels, langages et d'outils génériques à buts éducatifs comme Python, Java et Sonic Pi. Le SE peut être directement gravé sur une carte SD agissant à titre de disque principal pour l'ordinateur.

### 1.3.2 Python 3

Python 3 est le langage de programmation sélectionné dans le cadre de ce projet, plus spécifiquement Python 3.7.4. Python est un langage de programmation interprété et multiplateforme orienté objet non typé. Il existe de nombreux interpréteurs comme CPython (C++) et Jython (Java) permettant le lien entre le Python et d'autres langages.

### 1.3.3 NumPy

NumPy est une librairie permettant d'effectuer des calculs numériques divers en Python. Elle inclut le calcul matriciel et vectoriel, ainsi que les nombres complexes. NumPy est également l'outil priorisé par les programmeurs de Python pour le traitement des listes et tableaux pour sa flexibilité. La librairie est disponible sous une licence à libre accès. La version utilisée dans la version finale du programme de contrôle est la 1.17.4.

### 1.3.4 OpenCV

OpenCV est une librairie à libre accès utilisée pour la vision par ordinateur (de l'anglais Computer Vision [CV]) et l'apprentissage machine (de l'anglais Machine Learning [ML]). Des interfaces de la librairie sont disponibles en Python, Java, C++ et MATLAB. Elle est supportée sous Windows, Linux, Android et Mac OS. Dans le cadre du projet, la version 4.1.0.25 a été utilisée. Il ne s'agit pas de la dernière version disponible, mais celle-ci comportait la meilleure compatibilité avec les différents éléments du présent projet.

### 1.3.5 SciPy

La librairie SciPy est un outil de calcul permettant d'effectuer des routines numériques efficaces comme de l'interpolation, de l'intégration, de l'optimisation et du traitement de signal. Elle provient également d'un développement communautaire et est en libre d'accès. Comme son nom l'indique, elle supporte l'interface de Python. La version 1.4.1 a été utilisée pour le projet.

### 1.3.5 Matplotlib

Matplotlib est une librairie permettant la création de visualisations interactives, animées et/ou statiques en Python. Elle est le fruit d'un travail collaboratif à libre accès financé en grande partie par une charité à but non lucratif des États-Unis. La version 3.2.1 a été mise à profit lors de ces travaux de recherche.

## 1.4 Architecture du projet

Le projet a été développé selon une architecture séquentielle inspirée du Robot Operating System (ROS) (Koubâa, 2017), c'est-à-dire que des nœuds, ou objets peuvent être présent à titre de publieur ou d'abonné. Un objet publieur retourne une information utile pour un autre objet. Les objets abonnés sont quant à eux des

receveurs d'informations et peuvent par la suite traiter les données reçues pour effectuer une tâche. La tâche peut être numérique ou physique en fonction de la demande. L'architecture prévoit une structure séquentielle sans sous-programme (thread) afin de limiter la charge de calcul pour l'ordinateur. La séquence sélectionnée dépend du type d'analyse. La Figure 6 présente la structure hiérarchique du projet pour le mode « suivi » pour l'évaluation de la locomotion. Les flèches représentent le sens du transfert des informations générées d'un objet à l'autre. Les « packages » sont représentés par les encadrés pointillés.

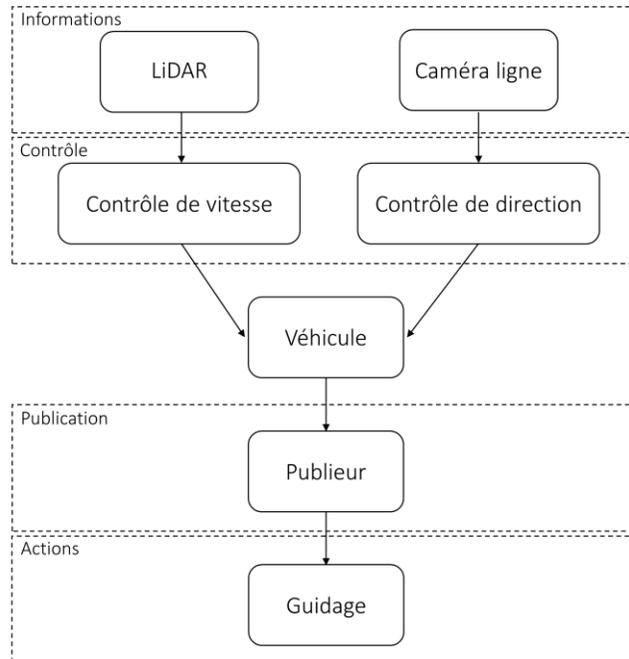


Figure 6 - Architecture du projet en mode « suivi » du participant. Chaque case représente un objet programmé en Python 3.7.

### 1.4.1 Programme LiDAR

De façon à permettre au robot d'implémenter un système de contrôle basé sur la position et la vitesse instantanée d'un sujet lors d'une évaluation, un objet de lecture de données adapté au module de détection sélectionné a été programmé. Cet objet est responsable de la connexion du capteur au protocole de communication I<sup>2</sup>C. Le programme interprète le signal du capteur et le converti par la suite en unité métrique, en l'occurrence des mètres. L'objet fait partie de la catégorie des informations dans le projet. L'appareil estime la distance entre son capteur et le participant en mesurant le temps de réponse de l'envoi d'un pulse lumineux. Le programme peut par la suite estimer la variation de vitesse entre le participant et le véhicule en évaluant la variation de la distance entre deux évaluations consécutives.

#### 1.4.2 Programme de la caméra de ligne

Similairement à l'objet LiDAR, un objet permettant l'acquisition d'un flux vidéo de la Raspberry Pi Camera Board v1.3 présentée en 1.1.5 a été développé. Le programme génère initialement un pont à l'aide d'une librairie afin de capter le signal CSI (Camera Serial Interface) provenant de la caméra utilisée. L'image RGB captée est par la suite soumise à différents traitements numériques afin de générer une projection de la trajectoire mathématique idéale devant être suivie par le véhicule (position médio-latérale par rapport aux lignes suivies).

#### 1.4.3 Contrôle de vitesse

Assurer le contrôle de vitesse est un élément primordial de la réussite du projet. Un sous-programme a été développé afin de permettre au système de convertir la distance perçue par le capteur LiDAR en commande machine de contrôle de vitesse. Le programme perçoit la distance comme l'information d'entrée en effectuant une requête au module LiDAR et calcule une valeur correspondante de pulse de type PWM pour le contrôle.

#### 1.4.4 Contrôle de direction

De façon similaire au contrôle de vitesse, le contrôle de la direction du véhicule, le cœur de la conduite autonome est réservé à un objet indépendant. Ce suiveur utilise la planification de trajectoire idéale du robot, laquelle provient d'une requête faite au programme de lecture de ligne, évalue la position de celui-ci à l'instant présent et calcule une réponse physique sous forme d'angulation des roues permettant la correction de la trajectoire du véhicule. Le contrôle est assuré par un algorithme de contrôle PID.

#### 1.4.5 Boucle de contrôle principale du véhicule

L'objet véhicule du programme agit à titre de chef d'orchestre. Il est responsable du passage des informations entre les intrants (les informations et régulations) et les extrants (les publications et les actions). C'est également cet objet qui est responsable de la séquence d'analyse, c'est-à-dire qu'il lance les appels procéduraux aux autres objets de façon à obtenir les informations nécessaires à la conduite du véhicule. L'interface de contrôle pour l'utilisateur provient également de cet objet.

#### 1.4.6 Programme de publication

Le publieur a pour but de récolter les informations traitées par le véhicule et les retransmettre aux différents modules de guidage. Il est un intermédiaire d'envoi permettant de faciliter la compréhension de la programmation.

#### 1.4.7 Programmation du guidage

L'objet de guidage traite quant à lui l'envoi des commandes aux différents appendices du robot. Il retire ses informations du publieur et envoie les signaux électriques demandés à l'aide de la carte PCA-9685 de laquelle il assure le contrôle. Ce module est le dernier passage informatique avant le monde physique. Il est le seul élément de la section actions qui est responsable des mouvements du robot (voir Figure 6).

### 1.5 Assemblage électrique

La conception électrique utilisée pour ce projet est présentée sous forme de schéma à l'Annexe D Montage électrique du circuit. Les détails entourant les décisions de son montage et des principes de connectivité sont présentés dans leur ensemble au cours des lignes de l'article publié dans le présent mémoire.

### 1.6 Environnement d'évaluation

Dans l'optique de réalisation d'évaluations de la locomotion, les travaux entourant le développement du robot ont été réalisés sur une piste d'athlétisme intérieure. L'Université du Québec à Chicoutimi (UQAC) possède une telle infrastructure dont les dimensions sont atypiques à l'exclusion des marqueurs de lignes qui respectent le standard de 0,10 m de largeur. En effet, une piste intérieure dite réglementaire dispose d'au moins quatre corridors dont la longueur totale du corridor intérieur est de 200m (Rigby Taylor). Les dimensions spécifiques de la piste sont indiquées dans le plan fourni par l'université et sont présentées à l'Annexe C Plan du Pavillon sportif de l'UQAC. Principalement, on dénote la présence d'un segment droit de 33 m et d'un rayon de courbure interne de 16 m. Chaque corridor dispose d'une largeur de 1 m en incluant la largeur des lignes de la piste.

## Chapitre 2 Vision par ordinateur

L'un des paramètres les plus importants permettant la réalisation du projet consiste en l'élaboration d'un algorithme de vision numérique. Dans le but d'assurer un cadrage adéquat de la caméra sur le participant, un algorithme de vision numérique permettant à VAALÉRIE de maintenir une conduite autonome dans un corridor de la piste a été développé. Ce chapitre présente les termes ayant mené au choix de l'élaboration d'un algorithme de contrôle plutôt qu'à l'utilisation de l'apprentissage machine comme un réseau neuronal convolutif. Tel que présenté par Steyvers et Waard (2000), l'œil humain s'adapte particulièrement bien à la conduite lorsque celle-ci est contrainte par des marqueurs comme des lignes continues. Néanmoins la nature continue de cette information s'arrime difficilement à l'interprétation discrète du monde d'un ordinateur. En effet, les systèmes modernes ont beaucoup plus de facilités à établir une évaluation ponctuelle des images en se référant aux pixels qu'à établir une reconnaissance de structures continues. De nombreux algorithmes de reconnaissance visuelle par apprentissage machine ont à ce jour permis la reconnaissance de forme, pensons par exemple aux réseaux neuronaux qui permettent la classification d'objets du quotidien, lorsque suffisamment entraîné (Kathuria, 2018). Néanmoins, ces systèmes nécessitent de longues périodes d'entraînement et utilisent une grande quantité de ressources machine à leur utilisation. Il faut également disposer d'une large base de données permettant d'entraîner ces réseaux. Par ailleurs, la nature cyclique de l'environnement de conduite risquerait d'induire du surapprentissage dans l'IA, diminuant ainsi la portabilité du système vers d'autres installations sportives dont les propriétés pourraient différer grandement. Ces éléments ont mené à la réalisation d'un programme ne faisant pas aller à l'intelligence artificielle.

### 2.1 Structure de l'algorithme de vision numérique

Le programme permettant la reconnaissance de ligne développé dans le cadre de ce projet peut être décomposé en 3 phases distinctes, soit l'acquisition, le traitement et la discrétisation. Elles ont pour but d'utiliser le caractère discret d'un ordinateur pour isoler et représenter mathématiquement de façon continue les images d'acquisition afin de déterminer la position du robot par rapport au centre du couloir d'analyse. Ces phases sont décrites dans les prochaines sous-sections.

#### 2.1.1 Acquisition

Comme mentionné dans la section 1.1.5, l'appendice responsable de la capture d'image est une caméra capable d'acquérir 30 images par secondes à une résolution 1920 x 1080 pixels (1080p). La résolution choisie dans le cadre du projet est de 960 x 640 pixels avec une fréquence d'analyse de 50 images par secondes. La communication entre l'appareil et l'ordinateur de bord est assurée par la librairie picamera qui permet l'insertion directe d'une capture vidéo image par image dans un programme Python. En associant les bons paramètres de collecte comme la résolution de l'image et la fréquence d'acquisition, une matrice bidimensionnelle contenant

la représentation entière BGR de chaque pixel est obtenue. Dès lors, l'image est prête à être transférée à l'algorithme de traitement.

### 2.1.2 Traitement

L'algorithme de traitement a pour but de modifier l'information visuelle obtenue dans la phase d'acquisition de manière à la rendre interprétable par le système de discrétisation. Les étapes de réalisation sont décrites dans l'article (Chapitre 4 Lane line detection) joint à ce mémoire. Plus généralement, la phase de traitement modifie l'image initiale de façon à obtenir une vision des marqueurs de la piste à vue d'oiseau, c'est-à-dire vu du dessus en plus de produire un masque dans lequel toutes couleurs autres que celles des lignes sont retirées, formant ainsi une image noire et blanche. La Figure 6.4 de l'article (Chapitre 4 Real Application for a Low-Cost Low-Power Self-Driving 1/10 Scale Car) présente le résultat final de la phase de traitement. On y aperçoit la vision projetée des lignes de pistes comparativement à l'image initiale perçue qui inclut la perspective induite par la caméra.

### 2.1.3 Discrétisation

Comme mentionné au début de ce chapitre, la nature discrète des ordinateurs force les méthodes d'analyse par vision numérique à interpréter les images par points, en l'occurrence, par pixel. Jusqu'à présent, le traitement a permis de modifier l'image d'acquisition en masque noir et blanc. En utilisant une fonction de redimensionnement, il est possible de représenter les lignes, ici les structures blanches, comme un alignement de pixels. Grâce à un filtre, il est possible d'éliminer les pixels dont la valeur se rapproche du noir, permettant ainsi d'éliminer les pixels adjacents ne permettant pas d'améliorer la qualité de la lecture. Le seuil d'intensité a été fixé à 170 sur 255, où 255 correspond à un blanc pur et 0 un noir exact. Par ailleurs, cette technique permet de limiter le nombre de points devant être traités afin de produire une lecture de ligne fiable. Une fois cette étape complétée, la position des pixels conservés est enregistrée dans une liste et est soumise à de nombreux traitements décrits dans la section « *Lane lines detection* » de l'article joint (Chapitre 4). Ces traitements permettent la conversion des points en pixels en valeurs mesurables, ici des mètres en plus d'effectuer une rotation de façon à les placer selon le système d'axe défini pour le véhicule. Finalement, le centre de la ligne est estimé à partir des données traitées et recueillies à partir de l'évaluation de deux polynômes du deuxième degré représentant chaque ligne.

## 2.2 Mécanique du traitement de l'image

### 2.2.1 Conversion du système d'encodage des couleurs

Il existe de nombreuses méthodes permettant d'isoler certains traits caractéristiques d'une image. Par exemple, le filtre de Canny (1986), encore énormément utilisé à ce jour, permet d'extraire les contrastes de couleurs d'une

image. Bien que cette méthode soit celle préconisée pour la reconnaissance de ligne, l'utilisation de ce type de filtre est extrêmement gourmande en termes de calcul. Pour contourner ce problème, il a été décidé de procéder à une analyse chromatique afin d'identifier les lignes de la piste. Le format usuel RGB, anciennement BGR est néanmoins un mauvais atout dans la détermination d'une teinte de couleur. L'encodage Hue, Saturation and Value, ou HSV permet de classifier les couleurs selon trois critères exactement comme le format RGB, mais utilise une astuce permettant de mieux discriminer les couleurs à choisir. La valeur de teinte, H, définit la couleur désirée tandis que la saturation et la valeur, respectivement S et V définissent la profondeur de la couleur ainsi que sa luminosité. Comme une saturation basse pour toutes teintes correspond à du blanc (Figure 7), il est possible de borner la couleur désirée en fonction d'un intervalle de teinte, de saturation et de valeur facilement. Comme le projet tend à extraire les structures blanches de l'image, en l'occurrence les lignes de la piste, des limites allant de  $[0, 0, 200]$  à  $[255, 255, 255]$  ont été utilisées, conservant ainsi toutes les teintes de blanc de la palette.

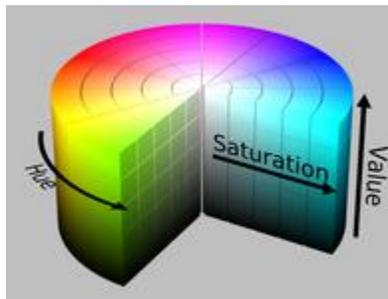


Figure 7 - Représentation visuelle de la distribution chromatique selon l'encodage HSV

### 2.2.2 Déformation de l'image

Le point de fuite principal de la perspective de l'image captée étant placé au centre, la disposition des lignes tend à joindre celles-ci vers le milieu de l'image. Bien que l'œil humain soit relativement efficace à interpréter cette perspective de façon à évaluer le parallélisme des lignes, un ordinateur ne peut y arriver sans qu'un algorithme ne soit implanté. De façon à retirer la perspective de l'image, une fonction de déformation a été mise à profit. Cette fonction utilise une matrice  $2 \times 2$  définissant la région d'intérêt (Region Of Interest – ROI) et l'image comme paramètre d'entrée. La ROI est limitée par les extrêmes des lignes observées sur une image de calibration sur laquelle les paramètres de position ont été mis en place avec précision. La transformation effectuée permet d'étirer l'image à partir des coins régissant la géométrie de la ROI, formant ainsi une impression de vue transversale (parallèle au sol) de l'image captée par la caméra. La Figure 8 présente l'image de calibration et les différents éléments permettant sa transformation. La zone d'évaluation est encadrée en orange et les coins formant la région d'intérêt sont représentés par des points rouges (à gauche) et verts (à droite). La

Figure 5 (Chapitre 4) de l'article présente également les différentes étapes de traitement de l'image mettant à l'évaluation du centre de ligne.



Figure 8 - Image originale utilisée pour la calibration.

### 2.2.3 Algorithme de tri de lignes

L'image étant maintenant transformée de façon à obtenir des lignes parallèles, un traitement supplémentaire est réalisé pour en retirer le caractère continu. La discrétisation de l'image décrite plus tôt dans la section 2.1.3 (Discrétisation) permet d'obtenir de façon relativement précise les différents points composant les lignes de la piste. Or, ces points doivent être triés de manière à classifier si le système perçoit une seule ou plusieurs lignes. De plus, il est possible que des points hors contexte comme ceux des lignes perpendiculaires ou des écritures sur la piste soient captés par le système bien qu'ils ne fassent pas partie de la sélection désirée. Pour se faire, il est nécessaire d'organiser un tri permettant d'ordonner les points d'importance et de rejeter le bruit. L'algorithme de tri de ligne est explicitement décrit dans la section « *Computer vision and control system– Lane lines detection* » (Chapitre 4) (Bégin, Duquette, & Lavallière, 2020). Tel que présenté dans cet article, le tri est effectué entre autres à partir d'un seuil angulaire dont l'origine mathématique est facilement explicable. En effet, le seuil utilisé correspond à la pente de la tangente à la courbe du point le plus loin du champ de vision de la ligne intérieur. Évidemment, cette valeur correspond à la dérivée de l'équation régissant le tracé de la ligne. Comme mentionné plus tôt, les courbes de la piste suivent une trajectoire circulaire dont le rayon interne est de 16 m. En considérant le centre de la courbe à 16 m au-dessus de l'axe horizontal (axe des x sur la Figure 9), on peut écrire l'équation régissant sa géométrie comme suit.

$$y = \pm\sqrt{16^2 - x^2} + 16 \quad (3.1)$$

$$\frac{dy}{dx} = \frac{-x}{\sqrt{16^2 - x^2}} \quad (3.2)$$

Pour  $x = 4 \text{ m}$ , la distance de projection maximale, la pente de la tangente est ainsi de 0.258, soit 14.47 degrés. Cette valeur peut ainsi être approximée à 15 degrés, représentant la dérive maximale perçue. La Figure 9 présente la géométrie du système selon la vue transversale.  $\beta$  représente l'angle maximal de fuite de la ligne par rapport à un déplacement parallèle à l'axe horizontal. Chaque corridor de course est borné par des traits gris à l'exclusion du corridor extérieur qui est borné par des traits noirs. Le trait pointillé représente la tangente de la courbe à 4 m.

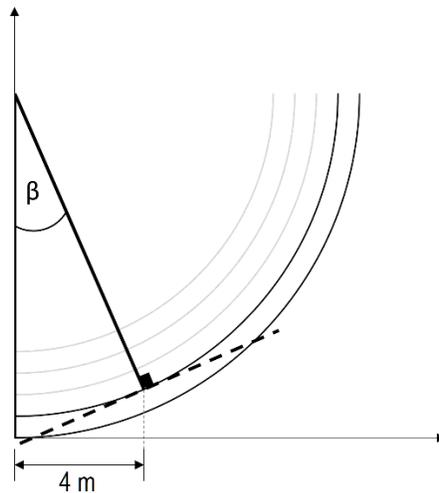


Figure 9 - Représentation de la dérive  $\beta$  de la ligne à 4 m, soit l'angle maximal perçu.

Le tri de lignes effectuées, l'algorithme génère par la suite une approximation mathématique de chaque ligne en utilisant un polynôme du deuxième degré. La trajectoire, en l'occurrence le centre du corridor, est par la suite estimée en effectuant la moyenne des deux polynômes estimés. Dans l'avenue où seulement l'une des deux lignes serait correctement perçue, l'algorithme estimera la trajectoire à partir de la translation de la ligne perçue sur une distance équivalente à l'erreur évaluée lors de la dernière analyse positionnelle. Dans le cas où aucune ligne n'est captée par la caméra, un premier *flag* est soulevé informant le programme de l'irrégularité observée. Un maximum de 5 *flags* consécutifs peut être soulevé avant que le véhicule ne considère qu'il soit hors de sa zone normale d'action et mette fin à la séquence de contrôle jusqu'à un arrêt complet. S'il s'agit néanmoins d'une erreur ponctuelle qui ne dépasse pas les 5 répétitions, le programme réinjectera la dernière projection de trajectoire comme étant celle actuelle. La nature relativement simpliste de la géométrie de la piste et la fréquence élevée d'analyse rendent cette approximation suffisante pour conserver un contrôle adéquat de la trajectoire. En effet, 5 erreurs consécutives à la vitesse maximale du véhicule ne correspondent qu'à une distance non contrôlée de 0,15 m.

## Chapitre 3 Asservissement

Deux paramètres essentiels ont dû être pris en considération lors de la réalisation du robot. Dans un premier temps, puisque l'environnement d'analyse a été limité aux corridors d'une piste de course, il est primordial de développer un modèle de contrôle permettant au robot de conserver une trajectoire rectiligne à l'intérieur des lignes de la piste. Rajamani (2006) catégorise ce type d'action comme le contrôle latéral des véhicules (contrôle médio-latéral). Une boucle de contrôle de type PID est responsable du contrôle médio-latéral du véhicule. Ce régulateur a pour but de conserver un cadrage adéquat de l'image en conservant une trajectoire parallèle et centrée aux lignes de corridor. D'autres types de régulations peuvent être utilisées afin d'assurer le contrôle en direction d'un véhicule à 4 roues motrices. Cependant, beaucoup de ces types dépendent d'un modèle physique du système. L'Annexe E Cinématique du véhicule propose un modèle en cinématique du véhicule développé dans le cadre de ce projet. Dans un deuxième temps, comme le référentiel de la majorité des analyses est fixé visuellement sur le participant, un suiveur permettant une supervision accrue de la distance entre celui-ci et le robot a dû être mis en place (contrôle longitudinal). Ce suiveur de type PD permet de conserver la bonne distance focale entre la caméra et le participant en plus de permettre l'évaluation de la vitesse de ce dernier. Ces deux systèmes sont décrits dans les prochaines sections de ce chapitre.

### 3.1 Contrôle de la trajectoire médio-latérale

De nombreuses méthodes permettant d'établir un contrôle de direction précis pour les véhicules sont à ce jour décrites dans la littérature. Raksincharoensak, Nagai, and Shino (2006) ont démontré que des systèmes orientés par le contrôle direct de la variable de contrôle proportionnelle  $P$  via le moment aux roues permet de générer un contrôle adéquat. Également, dans de nombreux, l'appel d'un modèle de contrôle prédictif (*Model Predictive Control – MPC*) permet d'établir des stratégies de conduite très performantes (Kun Law, Dalal, & Shearrow, 2018). D'autres groupes de recherche ont également démontré la faisabilité d'un contrôle via un réseau neuronal convolutif profond sur une plateforme similaire à celle utilisée dans le cadre de ce projet (Bechtel G., McElhiney, Kim, & Yun, 2018). Bien que ces alternatives de contrôle soient fortes intéressantes, celle-ci sont peu envisageables dans le cadre des travaux ici présentés en raison de leur effort de calcul important, de leur modèle dynamique trop complexe ou par le manque de données scientifiques permettant d'établir avec certitude que le modèle aura la capacité d'atteindre la conduite autonome dans la gamme de vitesse souhaitée. Ainsi, il a été déterminé qu'un système de contrôle de type PID (Proportionnel, Intégral, Dérivé) serait utilisé de manière à contourner les efforts de calcul dispendieux d'autres modèles. En plus d'être relativement simple à implémenter, l'élaboration d'un régulateur de ce type est grandement décrite dans la littérature (Abed, Ammar, & Shalaby, 2020), facilitant les travaux de cette section.

Comme spécifié lors de l'Introduction la trajectoire idéale que devrait emprunter le véhicule correspond au centre exact d'un corridor dont la largeur est de 1 m. L'objectif du contrôleur est de conserver le centre de masse du véhicule à 0.50 m de chaque ligne. Selon le système d'axe basé au centre de masse du véhicule, l'erreur de ligne observée correspond à la distance entre le centre du robot et le centre du couloir. Une déviation vers la gauche du centre du corridor correspond à une erreur positive tandis qu'une déviation vers la droite correspond à une valeur négative d'erreur. Parallèlement, une rotation des roues dans le sens antihoraire correspond à une angulation positive tandis qu'une angulation des roues dans le sens horaire correspond à une rotation négative. La Figure 10 présente la structure du régulateur de direction implémenté dans le projet. La consigne correspond à 0,0, soit au centre du corridor tandis que la position est le résultat de l'analyse de la position de du centre de masse par l'algorithme de reconnaissance de ligne par rapport au centre du corridor. Comme on cherche à maintenir une distance nulle entre le CM et le centre du corridor, la position du véhicule peut être interprétée directement comme l'erreur du système. Le régulateur calcule la valeur d'angle en degrés qui devra être appliquée à la direction pour corriger l'erreur perçue. Cette correction est par la suite appliquée au système de direction du véhicule en utilisant les unités cohérentes. Les variables de contrôle de la boucle de contrôle ont été réglées à l'aide de la méthode de Ziegler and Nichols (1993) telle que présentée dans l'article publié ci-joint (Chapitre 4).

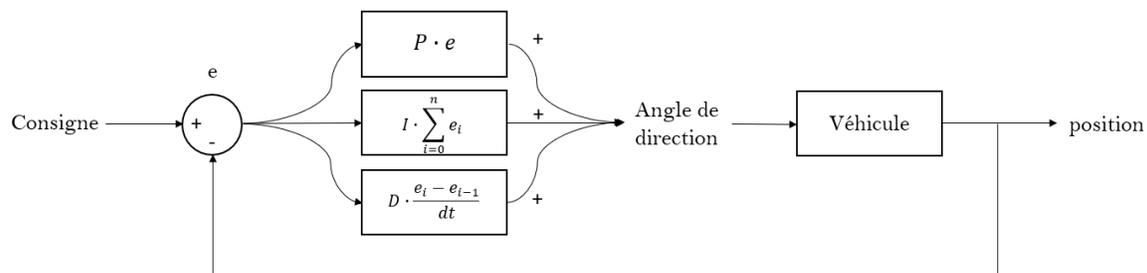


Figure 10 - Structure du régulateur de direction

### 3.2 Suivi longitudinal de la distance robot/patient

Tel que décrit au premier chapitre (1.4.1 Programme LiDAR), un appareil de mesure de distance de type lidar a été mis à profit afin de conserver une distance prédéterminée avec le patient. Comme cette position se modifie en temps réel lors d'une évaluation, il est primordial d'implémenter un second algorithme de contrôle permettant de moduler la puissance envoyée au moteur afin de conserver une distance moyenne stable. Le véhicule choisi pour la réalisation du projet contient un moteur à courant continu dont la charge nominale est de 7,4 Volts ainsi qu'un contrôleur électronique (1.1.7 Traxxas 4-TEC 2.0 AWD) permettant la conversion d'un signal PWM à un contrôle électrique. Pour des raisons similaires à celles spécifiées lors de la dernière section, un contrôleur PD a été utilisé afin d'assurer le contrôle de cette distance. Le régulateur de suivi prend comme valeur d'entrée la

distance perçue entre le patient et le robot et la convertie en longueur de pulse directement interprétable par le contrôleur électronique de vitesse. Les travaux de contrôle sont largement inspirés des modules d'enseignement offerts par MATLAB<sup>mc</sup> sur leur page YouTube<sup>TM</sup> (MathWorks, 2018).

# Chapitre 4 Real Application for a Low-Cost Low-Power Self-Driving 1/10 Scale Car

William Bégin<sup>1\*</sup>, Steven Duquette<sup>2</sup> and Martin Lavallière<sup>1</sup>

<sup>1</sup>Department of Health Sciences, Université du Québec à Chicoutimi, Canada

<sup>2</sup>Department of Mathematics and Computer Science, Université du Québec à Chicoutimi, Canada

\*Corresponding author: William Bégin, Department of Health

Sciences, Université du Québec à Chicoutimi, Canada

Accepted: November 30, 2020

Published online: December 02, 2020

Citation: Bégin W, Duquette S, Lavallière M (2020) Real Application for a Low-Cost Low-Power Self-Driving 1/10 Scale Car. *J Robotics Autom* 4(1):195-201

DOI: 10.36959/673/370

ISSN: 2642-4312

## 4.1 Résumé

Dans cet article, nous discutons d'une application réelle pour une voiture autonome à faible consommation d'énergie et à faible coût sur une plate-forme à l'échelle 1 / 10<sup>e</sup>. Nous présentons des algorithmes développés pour atteindre l'autonomie de conduite sur un ordinateur monocarte de faible puissance utilisant un processeur ARM dans un environnement contrôlé. Les auteurs donnent un aperçu de l'utilisabilité de cette technologie pour l'analyse de la marche et des performances. Nous effectuons des analyses de marche et de course sur une piste d'athlétisme en salle sur une configuration de came de suivi autonome. Nous démontrons en testant comment le véhicule peut produire des données fiables et cliniquement précieuses. Nous discutons des améliorations possibles et présentons des recommandations pour les travaux futurs.

## 4.2 Abstract

In this paper, we discuss a real-life application for a low power and low-cost self-driving car on a 1/10th scale platform. We present algorithms developed to achieve driving autonomy on a low-power Single Board Computer using an ARM-based processor in a controlled environment. The authors provide an insight on the usability of this technology for gait and performance running analysis. We perform walking and running analysis on an indoor athletic track over an autonomous follow cam configuration. We demonstrate through testing how the vehicle

can produce reliable and clinically valuable data. We discuss possible improvements and present recommendations for future works.

Keywords: Computer vision, Single Board Computer, Self-Driving, Locomotion Analysis, Low power computer, Autonomous driving, Raspberry Pi 4.

## **Introduction**

In early 2019, Tesla's CEO, Elon Musk, presented the company's first computer and chip to the world, named Full Self-Driving or FSD, capable of processing up to 2.5 billion pixels per second [1], an astonishing engineering feat. Although very impressive, many researchers and labs cannot afford or access this kind of computing power on portable devices for self-driving development. However, there are now many available options on the market capable of decent computing power at the lower end of the price range. For instance, let us think of the popular Nvidia Jetson Nano embedded computer or the even more popular, less brain-powered option, the Raspberry Pi 3B+. Though capable of self-driving on paper, very few studies present models with real-life conditions [2], reliability, scalability or specific applications other than following lane markings within closed lab doors [3,4].

This paper presents the full development of a 1/10 scale car equipped with a Raspberry Pi 4 capable of what would be considered as level 3 autonomous driving [5] used for the clinical analysis of locomotion.

## **Objective**

The basis of this research is to perform quantitative motion analysis such as running and walking [6] through a moving visual referential based on the subject's position. An indoor athletic track and field was selected as a control environment. The research takes place at a University featuring an atypical 168-meter 4-lane indoor track. Each lane has a common width of 1 meter with 5 cm white lane line markings. The single radius geometry conforms to the standards where the straight segment is twice the length of the inner radius, more precisely, a 16-meterspan in this case.

The secondary objective was to develop such an instrument using only low-cost devices capable of performing onboard computation, thus reducing the need for extra material. The limiting cost factor is of interest to compete with the currently used evaluation methods that are both affordable and well-established (ex. Paper-based checklist or timed evaluation using a stopwatch) [7,8]. Also, the system is developed to alleviate the use of wearable devices for conducting the gait analysis since such systems are often hard to use for a layperson in ecological settings [9].

## Project Configuration

### Mechanical components

The robot structure is based on recommendations stated by the F1TENTH community [3]. It is built around the TRAXXAS 1/10 4-Tec 2.0 AWD Chassis (McKinney, Texas, United States) (Figure 1). This R/C dedicated scale chassis can reach speeds of up to 48 km/h (30 mph) and figures, as stated by its name, a four-wheel drive configuration with differential on both the front and rear of the car. The vehicle is equipped with two ¼ inch (6 mm) steel rods functioning as mounts for the electronics. The assembly is mainly achieved with various 3D printed brackets and supports.

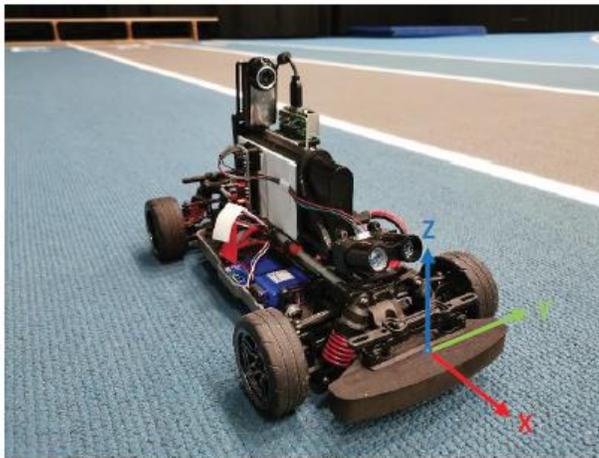


Figure 1: Complete assembly of the autonomous analysis robot.

### Electronic components

The computer module selected is the Raspberry Pi 4 (RP4) with 4GB of RAM (Cambridge, England). It has been equipped with a 32 GB Kingston Class 10 micro SD card (Fountain Valley, California, United States) for storage. The car comes stock with a TRAXXAS XL-5 Electronic Speed Controller (ESC), a Titan 12T brushed DC motor and a robust 2075 Waterproof servomotor. Both apparatus, the servomotor and the ESC, can be controlled via pulse modulation width (PWM). Though capable of producing software PWM throughout its general-purpose inputs and outputs (GPIO), the signal is too noisy to generate precise inputs for the ESC and the steering servomotor. To overcome this issue, an Adafruit PCA-9685 PWM (New-York, New-York, United States) driver board was added to the vehicle assembly. This device allows the RP4 to generate precise PWM via an I<sup>2</sup>C protocol and can be powered directly the computer. To conduct reliable subject following experiments, the robot was implemented with the Garmin LiDAR-LITE V3 (Olathe, Kansas, United States), a high precision laser distance sensor compatible with the RP4 I<sup>2</sup>C protocol. Lastly, the system uses a Logitech c905 (Lausanne, Switzerland) webcam as an RGB input directly connected to the RP4 USB port. Figure 2 shows the wiring diagram of the final assembly.

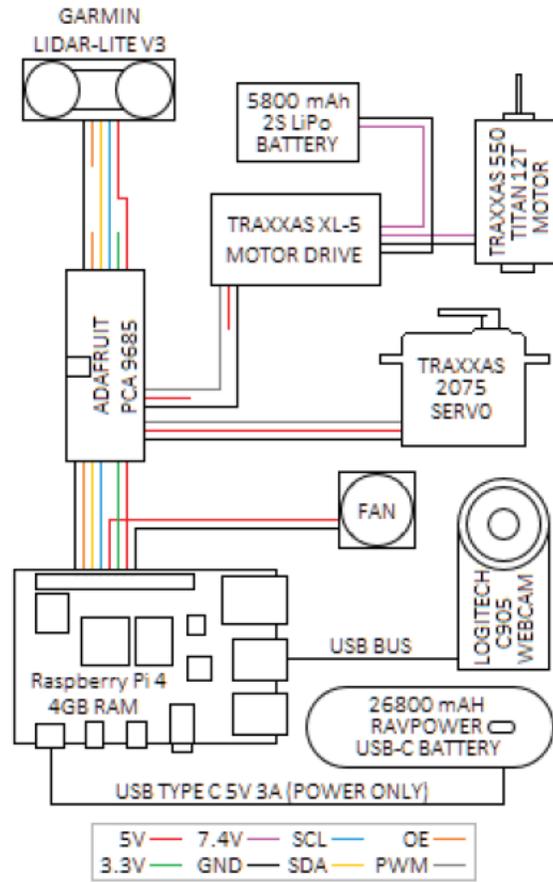


Figure 2: Final's assembly's line wiring diagram.

## Power

With simplicity in mind, robot assembly has been rigged with two independent power sources. A 7.4 V 5800 mAh 2s battery pack from the car manufacturer was used for propulsion, whereas a RAVPower 5 V 26 800 mAh USB type-C power bank was used to power the main computer and peripheral devices. This decision was also made to avoid the need of a voltage converter and to extend the robot's battery life.

## Software and project requirements

The RP4 runs on Raspbian, now called Raspberry Pi OS, a Debian-based operating system optimized for the Raspberry Pi product line.

Table 1 lists the libraries and the version used.

The entire code for this project was written in Python3, version 3.7.4. Many open-source libraries were needed to achieve this level of driving autonomy. Apart from the libraries needed for the integration of the sensor suite, Opencv, numpy and scipy were mainly used to complete this project.

**Table 1:** Project's required libraries.

<b>Library</b>	<b>Version</b>
Adafruit-Blinka	4.7.0
Adafruit-GPIO	1.0.3
Adafruit-PCA9685	1.0.1
Adafruit-PlatformDetect	2.8.0
Adafruit-PureIO	1.1.5
DateTime	4.3
adafruit-circuitpython-busdevice	4.3.1
adafruit-circuitpython-lidarlite	1.2.1
cycler	0.10.0
getch	1.0
key	0.4
matplotlib	3.2.1
numpy	1.17.4
opencv-python	4.1.0.25
pip	19.0.3
pyserial	3.4
python-dateutil	2.8.1
pyusb	1.0.2
scipy	1.4.1
setuptools	40.8.0
v4l2	0.2

## Project architecture

The project follows a rather simple architecture to reduce calculations and computing time. Information flows through the node sequentially without threads, as shown in Figure 3. Communication protocols are identified and represented by dashed lines. The robot performs its tasks by first reading the RGBcamera input and generating an output value through a controller. It then acquires data from the LiDAR sensor, providing the necessary values to perform a throttle output evaluation through another distinct controller. Values are then transferred to the PCA 9685 module by the RP4 via the I<sup>2</sup>C protocol which then outputs the signals to the ESC and to the steering servomotor, thus inducing trajectory modifications on the vehicle. The loop is repeated as long as needed, but can be stopped at any time by the evaluator or if the algorithm raises any safety concerns.

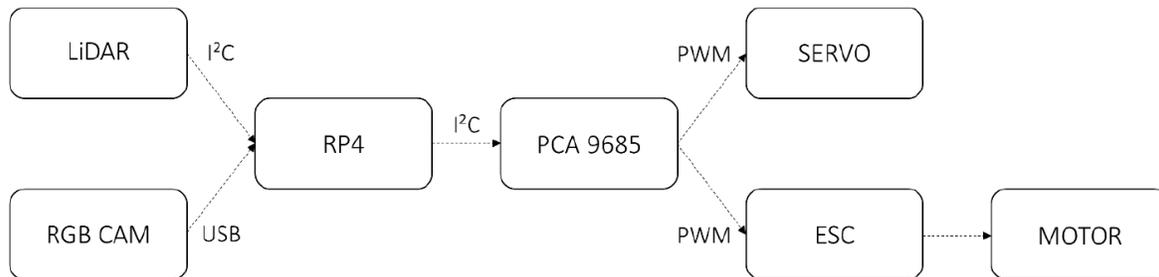


Figure 3: System diagram and communication protocol.

## Computer Vision and Control Systems

### Lane line detection

One of the key elements of the complete project is the use of a Low Power computer to perform a self-driving task. Though powerful enough to perform small artificial intelligence tasks, the RP4 is not optimized for neural network computations or machine learning training, which lead to the decision to produce a computer vision algorithm instead. As a matter of fact, the analysis environment of the project is sufficiently well-known and standardized to not require a more general approach line of a convolutional neural network. Many open-source lane detection algorithms are already available from a vast selection of suppliers, but none seemed to fit the needs of the project. Most of the codes available offer self-driving abilities with restricted capabilities. Projects often tend to use non-standard markings and scaled lane widths while not offering the speed capabilities anticipated by the team. To overcome these issues, it was decided that a computer vision algorithm would be developed, thus enabling self-driving in real-life conditions and applications, in our case within athletic track markings. Most of codes available over the Internet regarding lane detection tend to use the Canny edge detection over an area of interest (lane marking areas, generally the lower part of the input image) and then identify linear patterns with the Hough transform algorithm [10]. These patterns are returned as line extremums, then sorted as left and right line components and finally added to the evaluated frame for visualization. Though visually pleasing, this method does not provide a good approximation for curved lines, nor a mathematically valuable tool for trajectory evaluation.

The lane detection code written for this project is in part based on the Ross Kippenbrock algorithm [11]. An RGB  $960 \times 170$  resolution camera image is first fed into the program trimmed to remove over horizon area and resize to a  $300 \times 170$  size to lower computational costs. The image is then modified via a warp function to create a synthetic bird's eye view. Using a color mask function, the lane line pixels are then selected according to their color and shade. An eroding function is applied to filter noise, thus removing undesired pixel agglomerates. To extract a discrete output from a pixel array, the image is then resized to about fifty percent of its original size. Selecting only the brightest pixels (closest to  $[255, 255, 255]$ ), an array of around 200 points is created from the original image. Points are then rotated according to the vehicle coordinates (forward corresponding to +x axis

and left +y axis as seen in Figure 1). The origin is at the center of the vehicle. To achieve reliable line sorting, we developed a robust algorithm. A vertical pixel sum of the first half of the resulting image is performed, generating a pseudo signal according to the predominance of white pixels. Since the distance between the two lines is constant throughout the track length, we can perform a signal analysis on the sum array using the `scipy.signal` library. By considering the lane width as the predicted wavelength, a parameter of the library's `find_peaks` function, we can identify apex cause by the line markings. If more than two vertices were detected, we compare these values to previous results to avoid false line detection and select the two best options. By using the peaks as starting points for each line, we can define the slope angle of the segment formed by our initial point to the closest one. If this point is within a threshold, in the case of this study being 15, the next point is considered as being part of the line and added to a list. This threshold is derived from the maximum possible lane deviation (in degrees) of the innermost line of the track described as part of this research at a 2.5-meter distance (camera horizon) with an added error value of 5 degrees. The algorithm is repeated for each point, based on the last point added in the list and the previous slope angle as well as for each peak selected. This algorithm helps to remove perpendicular values and “off the chart” data. The code algorithm is presented below. Two 2nd degree polynomial equations representing lane lines are computed from the two separate data sets created. These two functions are then averaged to evaluate the estimated lane center.

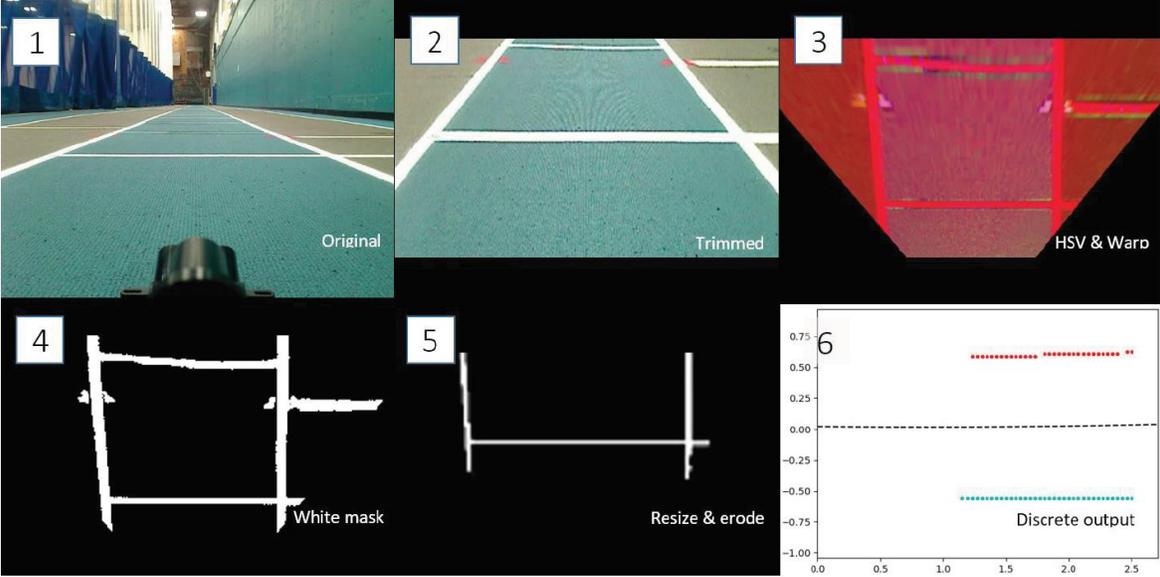


Figure 4. Lane lines filtering algorithm steps performed on actual data input.

## Algorithm 1: Line data sorting.

```
lines_point[ ]
for points in pointlist[ ] do
    |   sort  $(x_i, y_i)$  where  $x_i < x_{i+1} \forall i$ 
end for
histogram[ ]  $\leftarrow \sum_{i=0}^{n/2} image[i]$ ,  $n = image\ width$  for each rows in image height
peaks[ ]  $\leftarrow find\_peaks(histogram[ ], track\_width)$ 
peaks[ ]  $\leftarrow two\ best\ peaks\ compared\ to\ previous\ peaks[ ]$ 
for peak in peaks[ ] do
    |   last_theta  $\leftarrow 0$ 
    |   lst[ ]  $\leftarrow peak$ 
    |   for point in pointlist[ ] do
    |       |   theta  $\leftarrow angle\ between\ point\ and\ lst[-1] - last\_theta$ 
    |       |   if theta < 15 then
    |       |       |   lst.append(point)
    |       |       |   last_theta  $\leftarrow theta + last\_theta$ 
    |       |   end if
    |   end for
    |   lines_point.append(lst)
end for
```

## Steering control

As shown in Figure 4, the computer vision algorithm produces a smooth and reliable ideal trajectory for the car to use as a goal. This 2nd degree polynomial equation can be used to approximate the car's actual cross track error (CTE) by evaluating the function's value at the vertical intercept, the car being conveniently placed at the origin. The evaluated CTE is fed into a proportional-integral-derivative (PID) controller which then outputs a pulse width value converted into torque change by the motor. The robot was initially implemented with a simple model predictive control (MPC) using an in-house-developed mathematical steering model considering the four-wheel drive nature of the car. However, this method proved to be too time-consuming in terms of computing and was dropped early in the project.

## Throttle control

While steering control is always performed according to lane line markings, the robot can carry out different assessments requiring a wide range of speed control patterns. Training and evaluation with predefined velocities, only required a user speed profile input. However, many analyses are based on the subject's speed. To achieve a constant distance with the subject, the vehicle utilizes an optical distance measurement sensor. Control of speed and distance is achieved through a PID controller, using the current subject/robot distance as

an input and computing a throttling output for the ESC. As of right now, the PID parameters have been tuned using the Ziegler-Nichols method [12].

## Data collection

The main goal of this research is to produce similar results as what would be possible for an in-lab visual locomotion analysis. Such evaluations are generally performed on a treadmill or over a controlled area and use a so-called global referential where the room is the spatial reference. In this project, using such a system would induce the need for real-time position estimation as well as complex computations to correctly evaluate the subject's locomotion patterns. As a bypass, it was decided that the spatial reference would be fixed to the subject, thus nulling the need for accurate position estimations, as shown in Figure 5.

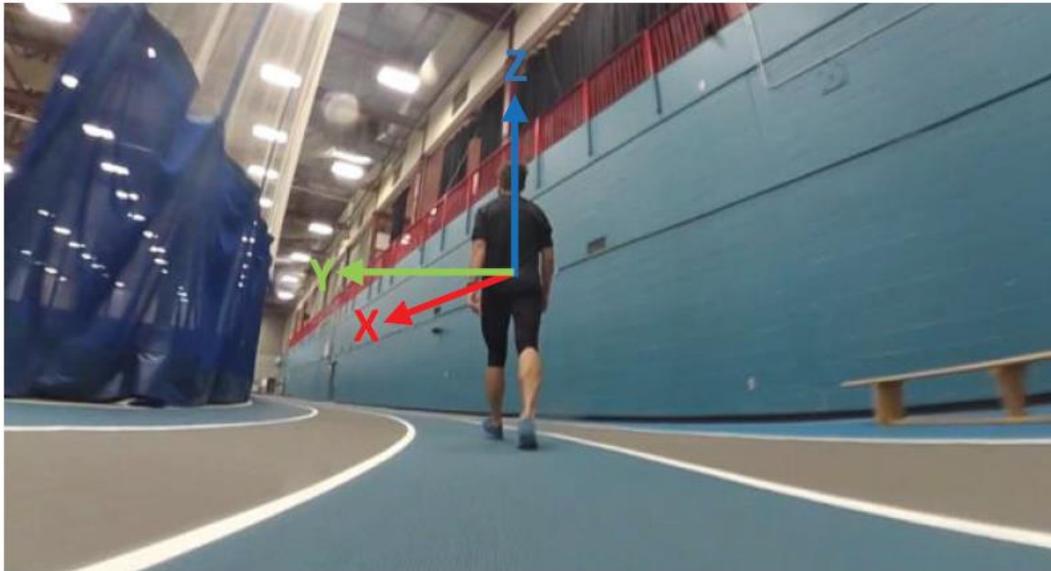


Figure 5: Walking analysis performed on an indoor athletic track with a goal distance of 2.5m.

Although many evaluation types can be performed using the robot, this paper will focus on a following evaluation (the robot is placed behind the subject being evaluated). The basics of this evaluation are simple. A distance or time is first selected by the evaluator to conduct the desired evaluation. A track lane must then be selected independently of its color or length to perform the evaluation, according to the test requirements. To ensure a good visual frame for image capture, the evaluator is encouraged to modify the goal distance between the subject and the vehicle. The set distance for this paper was 2.5 meters ( $\approx 100$  inches). To engage an evaluation procedure, the vehicle is placed on the selected lane. Though the robot does not have to be perfectly placed in the center of the lane, positioning it on a fairly straight alignment and close to the middle will help the vehicle perform better from the beginning. The vehicle will recover the initial placement error within the first 5 meters. Critical positions include a plus/minus 45 cm away from the lane center and a plus/minus 30 degree offset from

the lane center tangent. Once the car is placed as desired, the subject can take place in front of the vehicle. We recommend placing the subject somewhere between 50 to 25 percent closer to the set distance to avoid premature correction from the robot. The evaluator can then proceed with the car's program start-up. This can be achieved via the remote desktop application available on the Windows platform (Windows 10, Microsoft, USA). The robot's main computer has been programmed to generate a secure Wi-Fi network from which a remote computer can be connected to perform the assessment. A starting countdown will lead to the beginning of the test. The subject can then proceed with to walk and/or run.

## Results

Trials were conducted through different testing which include a 180 m (600 ft.) walk (1.25m/s or 4.5km/h), a 180 m run (3.44 m/s or 12.4 km/h) and a trial at variable paces for 540 m (1800 ft.) (1.3, 1.5 and 2.5 m/s, or 4.7, 5.4 and 9 km/h, respectively). All tests were carried out on an indoor athletic track, as described in the objective. Experiments were conducted on two different subjects presenting no functional limitations or pathologies. Tests were conducted to better understand controller tuning and project limitations. Figure 6 presents the plotted results of the experiment, i.e. the CTE and steering output ( $\delta$ ) through time and the throttle output (APW) compared to the perceived distance (D) through time from the 555 m evaluation. All analyses were performed at an average rate of 10.1 Hz, a similar value to the 10.3 frames per second performed by the main computer when only the camera is in use with an external webcam app (Guvviewer, <http://gucvview.sourceforge.net/>) at a 960 × 720 resolution.



Figure 6: Results from the walking experiment a) Distance control primary axis and throttle output on the secondary axis; b) Perceived cross track error (primary axis) presented with the steering response (secondary axis).

## Discussion

Overall, current results show that this platform is effective at providing quantitative data of gait parameters [6] using a self-driven vehicle in a quasi-controlled environment. Data show that further improvements are needed on the tuning of the PID values. As an example, even if gait induces a sinusoidal kinematic response, output extirpated from this analysis are too rough to represent it graphically. A 0.15 Hz frequency on the distance control can easily be observed in Figure 6a, a number too small to be expressed as a gait pattern [13]. Moreover, some analysis segments, as seen at 100 and 240 seconds, present more dramatic PID corrections over the period of 20 seconds with an amplitude variation of 20 centimeters. These variations could be explained by the subject's torso movements, which are currently considered as steady by the software.

Regarding the autonomous lane driving control, Figure 6 depicts adjustments made by the vehicle based on what is perceived and computed by the camera. A variability in the steering output can be observed, and is proportional to the input from the lane line detection, giving it a saw tooth appearance. We can observe that even small lane detection errors are also considered in the vehicle's direction. Further data filtering could help prevent steering overcompensation, thus providing the system with a smoother drive. A more restrictive line

evaluation protocol could also be implemented to better discard “off the chart” line detection. Such results are promising since we were able to perform basic motion analysis from a steady frame on a moving subject. Although further testing and tuning will be needed to demonstrate the robot’s real capabilities, results obtained throughout this experiment demonstrate the feasibility of such tool.

## Cost

The estimated cost of the required equipment for the current system is approximately \$1,000CAD (\$750). As presented by Müller & Koltun [2], this project is comparable to other low-cost vehicles developed for automated driving and racing. Moreover, the current system is a stand-alone platform for navigation in quasi-unrestrained environments since other systems are typically designed to track lines on a laboratory floor where the aim of the project is to “train” the system to go faster using machine learning or neural networks. Therefore, the developed system responds to our second objective which was to provide a low-cost device for the analysis of human movement.

## Safety

Safety is a major concern when working with human and any living beings. To ensure the subject’s safety throughout our testing, the robot was developed with some preprogrammed safety measures. A programmed safety switch was integrated to the code, making it possible to terminate any evaluation at any time from the remote computer, thus completely stopping the car. Another safety measure comes from the LiDAR coding. The code was implemented with a proximity warning that will ultimately stop the car and terminate the program if the robot comes too close to, or goes too far from a test subject or any other obstacle. Moreover, lane lines must be identified at all times to ensure that the car remains running. To overcome glitches and inaccuracies, the code can raise a maximum of 3 consecutive undetected line warnings before terminating, thus completely stopping the vehicle. If the wireless connection fails between the remote computer and the robot, the vehicle will also come to a complete stop.

## Conclusion

The current work demonstrates that a low-cost low-power self-driving 1/10 scale car can be used to assess walking and running in a controlled environment. The data gathered is reliable and can be clinically valuable for health specialists when assessing one’s mobility. Future studies will focus on implementing more complex analyses of human movement for the automatization and production of quantitative data to limb movements [14]. To produce more reliable analyses in the future, the vehicle could be equipped with a less restrictive distance-measuring device, such as a rotating LiDAR or even through visual estimations. More improvements on lane line detection should also provide the robot with a steadier control. The actual processing speed of 10 frames per second seems to be a limiting factor for the team in view of the maximum speed that can be safely reached

by the robot. Further analysis and code modifications will be carried out to achieve a more efficient goal of 20 frames per second on the same hardware. Since the camera feed was identified as a limiting speed factor, a better, smarter, and faster U.S.B. 3.0 webcam could be implemented in parallel to a smaller input frame resolution.

Future work will include the onboard kinematic motion analysis to better understand human locomotion and further document motion patterns from this new perspective. Also, induced vibrations produced by the track deformities should be considered to better capture motion. Modern software can produce reliable video capture by numerical stabilization, which could be implemented in this project.

## Acknowledgements

This work was supported by a grant from Centre intersectoriel en santé durable (CISD) of Université du Québec à Chicoutimi (UQAC).

## References

1. Bos C (2019) Tesla's new HW3 self-driving computer-it's a beast (CleanTechnica Deep Dive). CleanTechnica.
2. Müller M, Koltun V (2020) OpenBot: Turning smartphones into robots. arXivLabs.
3. (2020) F1TENTH Foundation. F1TENTH.
4. Stranden JE (2019) Autonomous driving of a small-scale electric truck model with dynamic wireless charging. Norwegian University of Science and Technology.
5. Society of Automotive Engineers (2018) Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. SAE International, 35.
6. Cimolin V, Galli M (2014) Summary measures for clinical gait analysis: A literature review. *Gait and Posture* 39: 1005-1010.
7. Owsley C, Sloane M, McGwin G, et al. (2002) Timed instrumental activities of daily living tasks: Relationship to cognitive function and everyday performance assessments in older adults. *Gerontology* 48: 254-265.
8. Wall JC, Bell C, Campbell S, et al. (2000) The timed get-up-and-go test revisited: Measurement of the component tasks. *J Rehabil Res Dev* 37: 109-113.
9. Stone EE, Skubic M (2011) Passive in-home measurement of stride-to-stride gait variability comparing vision and kinect sensing. *Annu Int Conf IEEE Eng Med Biol Soc* 2011: 6491-6494.
10. Canny J (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 8: 679-698.
11. Keenan R, Kippenbrok R, Brucholtz B, et al. (2017) Advanced lane finding.
12. Ziegler JG, Nichols NB (1993) Optimum settings for automatic controllers. *J Dyn Sys Meas Control* 115: 220-222.

13. Winter DA (1993) Knowledge-base for diagnostic gait assessments. *Med Prog Technol* 19: 61-81.
14. Ramakrishnan T, Kim SH, Reed KB (2019) Human gait analysis metric for gait retraining. *Appl Bionics Biomech* 2019: 1286864.

# Chapitre 5 Résultats

Le grand nombre d'éléments formant le projet impose que les résultats soient divisés en plusieurs sections. La section suivante décomposera le projet selon les résultats obtenus par l'algorithme de reconnaissance de lignes, pour la conduite autonome, la collecte préliminaire de vidéo et d'analyse de la performance.

## 5.1 Résultats de vision numérique

L'algorithme de vision numérique est le cœur du contrôle latéral du véhicule, permettant la réalisation de collectes d'images stables. Il est primordial de générer une lecture stable afin de limiter les erreurs possiblement perçues par le contrôleur dans les étapes subséquentes. Il est d'abord pertinent de s'intéresser à la précision et la répétabilité pouvant être obtenue par le système.

### 5.1.1 Fréquence d'analyse

La fréquence d'analyse d'image pour atteindre la conduite autonome est un facteur de première importance. Humes, Bussey, Craig, and Kewley-Port (2009) ont démontré que l'œil humain a une fréquence d'analyse moyenne équivalente à 45 Hz. De nombreux travaux de recherche tendent à démontrer qu'il est possible d'atteindre la conduite autonome stable avec une fréquence de captation équivalente à 24 images par seconde. La caméra utilisée permet l'acquisition d'images à une fréquence allant jusqu'à 90 Hz bien que la majorité des analyses aient été effectuées à 40.5 Hz en raison du délai de traitement de l'image et de calculs de contrôle. À titre de comparaison, Bechtel G. et al. (2018) ont obtenu des fréquences d'analyse pouvant atteindre 40 Hz sur un Raspberry Pi 3 aux performances similaires à celle du RP4 utilisé dans le cadre de ce projet. Ces valeurs ont toutefois été obtenues à partir d'un contrôle effectué par un réseau convolutif de neurones (CNN) largement basé sur l'algorithme DAVE-2 de Nvidia (Bojarski et al., 2016). En contrepartie, des chercheurs ne sont pas parvenus à atteindre de telles fréquences avec des latences allant jusqu'à 1 Hz malgré l'utilisation d'ordinateur beaucoup plus puissant, en l'occurrence un NVIDIA Jetson TX2, avec un modèle de vision par ordinateur (Stranden, 2019).

### 5.1.2 Précision de lecture

Le traitement numérique effectué sur les images modifie de façon importante le caractère initial de chaque capture. Comme présenté au

Chapitre 2 Vision par ordinateur, de nombreuses manipulations informatiques dénaturent l'image initiale. Le rendu final du traitement produit une image dont la résolution correspond à 136 par 38 pixels avec une valeur respective de 54 pixels par mètre selon l'axe y (en largeur) et 22 pixels par mètre selon l'axe x (en profondeur). On peut estimer que chaque pixel offre une précision de 0,0185 m.

De manière à attester la répétabilité des données de l'algorithme de reconnaissance de lignes, une série de tests a été mise sur pieds. Dans un premier test, le véhicule a été placé au centre d'un corridor et son système d'identification de la position a été mis en marche tout en s'assurant que le contrôle de direction ait été au préalable éteint afin d'éviter une modification de la position du véhicule engendrée par des mouvements du système de direction. Une collecte de 60 secondes durant laquelle l'acquisition de la position estimée par la vision numérique a été effectuée. L'algorithme de contrôle complet a été réalisé durant l'essai afin de simuler adéquatement le délai entre les captures. La Figure 11 présente le signal de lecture de la position perçue pour un essai en fonction du temps par rapport à une position fixe (trait rouge). Le trait bleu représente la lecture observée, les traits pointillés, eux, les variations possibles induites par la précision de lecture (1 pixel = 0,0185 m). L'écart-type est présenté sous forme de zone ombragée derrière la lecture, ici en gris. Il est intéressant de remarquer que la lecture de la position se trouve à l'intérieur des limites de la précision d'un pixel, soit 0,0185 m. On peut remarquer que la variabilité de la lecture est également nettement inférieure à la précision offerte par l'algorithme. En effet, le bruit observé dépasse à peine par moment 0,005 m.

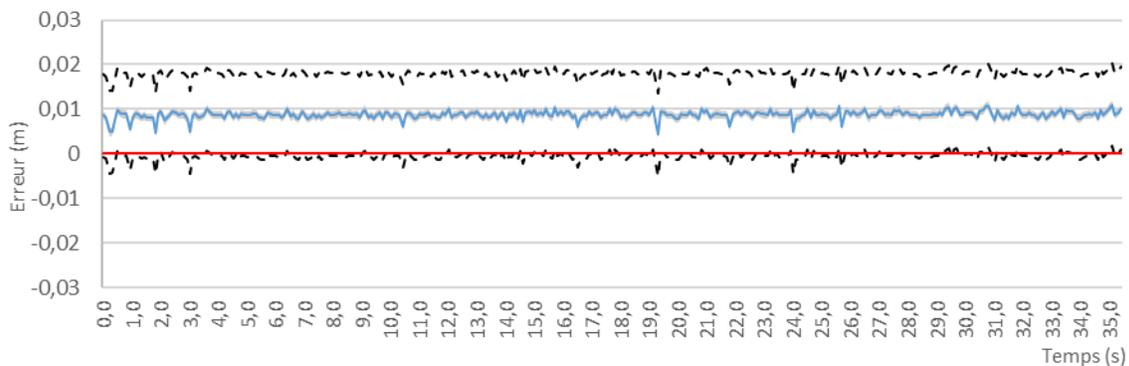


Figure 11 - Erreur perçue en fonction du temps sur un essai statique au centre du corridor

Dans un second test, la validité de l'évaluation positionnelle du système de vision sur toute la largeur d'un couloir a été évaluée. Pour ce faire, une nouvelle analyse avec le contrôle de direction éteint a été effectuée. Lors de ce test, le véhicule est initialement placé au centre du corridor pour une période de 45 secondes puis est déplacé latéralement de 0,1 m pour une nouvelle période de 45 secondes. Le déplacement est répété jusqu'à ce que le véhicule ait atteint l'extrémité latérale du corridor. La Figure 12 présente le résultat de l'observation de l'erreur selon quatre positions préétablies, soit 0, 0,1, 0,2, 0,3 mètre de la ligne centrale. La figure présente des valeurs négatives puisque le test a été effectué par incréments de position vers la droite en respect au référentiel choisi. Des essais avec des incréments positifs ont également été réalisés, mais leur symétrie avec celui-ci rend leur

présentation superflue. La position réelle, mesurée à l'aide d'un ruban, est représentée sur la figure par le trait continu rouge tandis que les valeurs estimées par l'algorithme sont présentées par les nuages de points bleus. Chaque position a été conservée sur une période de 45 secondes. On trouve respectivement une erreur moyenne sur la lecture de la position de  $-0.0159$  m pour une position centrale (0 m),  $-0,0354$  m pour une déviation latérale de 0,1 m,  $-0,0372$  m pour une position à 0,2 m de la ligne centrale et finalement  $-0,01347$  m pour une déviation latérale de 0,3 m. On remarque aisément que l'erreur estimée la plus faible se trouve dans la portion d'erreur équivalente à 0 m et que l'erreur d'estimation semble croître avec l'augmentation de l'erreur réelle. Il est possible que cette situation soit en partie causée par la modification importante de l'image lors de la phase de déformation de celle-ci pour obtenir une vue à vol d'oiseau de la piste. En effet, les manipulations d'étirement de l'image induisent une déformation plus importante aux pixels en périphérie que ceux au centre de l'image. Il est néanmoins important de dénoter que la précision maximale d'un pixel est respectée pour un cas d'erreur nulle et engendre une erreur minimale à 0,1 m et à 0,2 m d'erreur nominale.

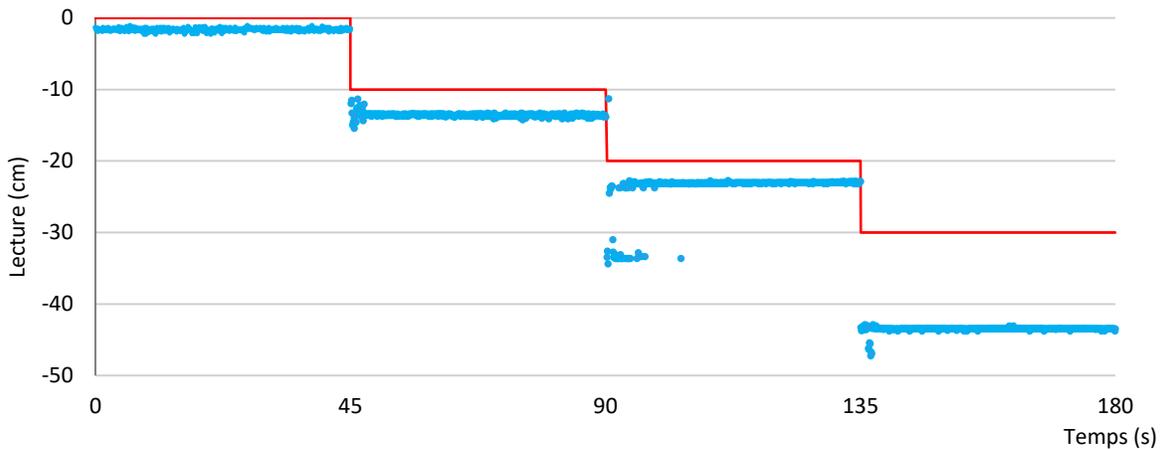


Figure 12 - Erreur mesurée selon la position en fonction du temps

Lors des essais sur le terrain, il a été remarqué que certaines valeurs extraordinaires d'erreurs étaient calculées par l'algorithme de vision numérique, induisant des corrections inutiles et importantes dans la conduite du véhicule, menant généralement à une déviation importante. Ces corrections poussent même parfois le système à tomber en arrêt. Dans l'optique d'éviter ce genre de problématique, le système d'analyse a été implémenté d'un filtre à moyenne mobile permettant de limiter les effets d'une valeur extraordinaire. La Figure 13 présente un essai où une lecture prononcée est observée à 27 secondes ( $-0,044$  m). Le trait bleu marque les lectures brutes observées tandis que le trait rouge délimite la lecture filtrée par la moyenne mobile effectuée sur les 5 dernières valeurs. Bien que l'erreur de lecture engendre une légère modification de l'erreur considérée, on remarque que celle-ci ne diffère pas en amplitude des autres erreurs filtrées lors de l'essai. En effet, l'erreur corrigée ne dépasse pas la précision de lecture de  $0,0185$  m. La Figure 14 présente la distribution des valeurs d'erreurs perçues par le système avec et sans algorithme de filtrage. On remarque aisément par la forme des violons que les données filtrées sont largement moins répandues que les données brutes.

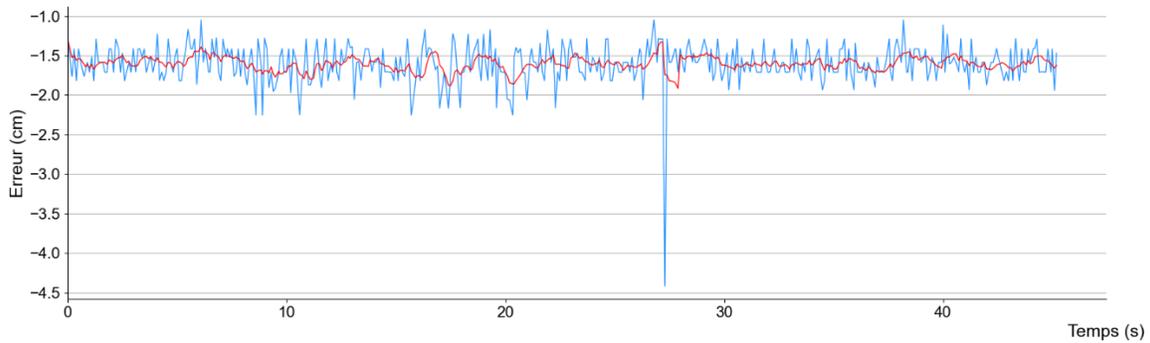


Figure 13 - Filtre d'erreur par moyenne mobile sur 5 points. Données brutes en bleu, signal transformé en rouge

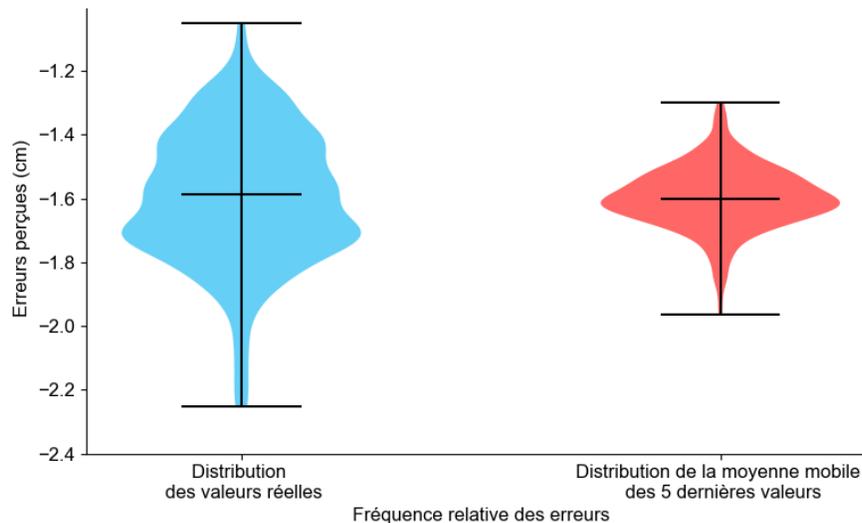


Figure 14 - Diagramme en violon représentant l'influence de l'utilisation d'une moyenne mobile sur 5 valeurs sur la variation de la lecture de la position du véhicule par rapport au centre de la voie comparée aux valeurs réelles sans traitements.

## 5.2 Vitesses et performances

La prochaine section s'intéresse aux résultats du groupe moteur du véhicule en conduite autonome. Les résultats présentés ont été obtenus à la suite de tests terrain réalisés sur la piste d'athlétisme de l'Université. Ces résultats sont également comparés à ceux obtenus dans la littérature.

### 5.2.1 Caractérisation du comportement du moteur

Comme le projet se veut une phase de prototypage et de démonstration de faisabilité de développement d'un outil d'analyse, certains éléments de mesure de précision ont été mis à l'écart de façon à simplifier l'ensemble du projet. Par exemple, le véhicule n'a pas été muni d'un dispositif de mesure de vitesse directe. Celle-ci est plutôt estimée à partir d'essais empiriques caractérisés par la commande électronique envoyée au moteur par le programme (pulse électrique). La Figure 15 présente les résultats des essais réalisés sur la piste d'athlétisme. La vitesse réelle a été mesurée en fonction du temps mesuré par tour de piste effectué, en l'occurrence 186.7 mètres. Les 13 essais effectués ont permis de présenter un modèle dont la courbe de tendance, décrite par un

polynôme du deuxième degré, décrit la vitesse mesurée en fonction du pulse imposé. La courbe de tendance obtenue a un coefficient de détermination équivalent à 96.57%. Bien qu'un polynôme de plus grand degré augmenterait ce coefficient, la caractérisation du comportement d'un moteur électrique est généralement plus similaire à une droite plus stable dans un intervalle court. L'utilisation d'un polynôme de plus haut degré correspondrait à du surapprentissage.

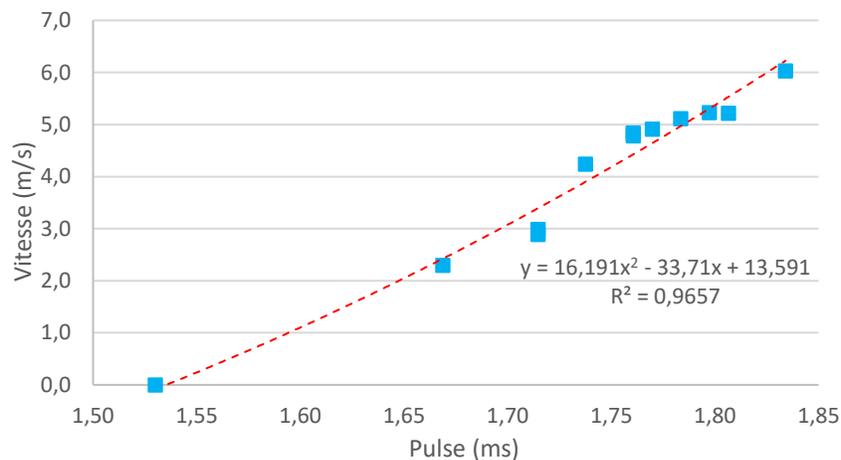


Figure 15 - Vitesse réelle mesurée en fonction de la commande (pulse) transmise par le système de contrôle.

## 5.2.2 Comparaison des résultats de vitesse

La vitesse maximale atteinte en conduite autonome par le véhicule est de 6,03 m/s (21,7 km/h). La limite a été fixée à cette valeur puisqu'il s'agit de la vitesse maximale à laquelle le véhicule est parvenu à réaliser un tour complet de piste à l'intérieur d'un corridor sans tomber en arrêt ou de celles-ci. Ces données sont en ligne avec les systèmes présentés dans le Tableau 1. Le projet VAALÉRIE se situe ainsi entre les plateformes DeepRacer et MuSHR, deux plateformes utilisant des algorithmes d'intelligence artificielle et des technologies de LiDAR à capture tridimensionnelles pour atteindre la conduite autonome. De plus, ces deux modèles utilisent des ordinateurs dont la performance est largement supérieure à celle du RP4.

## 5.3 Résultats de régulation latérale en déplacement

Comme présenté lors des dernières sections, deux régulateurs distincts ont été utilisés afin de produire la régulation nécessaire à la conduite autonome et au suivi adaptatif des participants. De nombreux essais effectués sur la piste ont permis l'acquisition de données menant au réglage adéquat des boucles de contrôle.

### 5.3.1 Résultat du contrôle médio-latéral

De manière à attester la fonctionnalité de la conduite autonome, le véhicule a été mis à l'essai sur la piste d'athlétisme. Lors de ces essais, le véhicule est initialement placé au centre du quatrième corridor et est activé

afin de compléter un déplacement total équivalent à un tour complet de piste, soit 186,7 m. La vitesse a été variée selon les essais afin d'en évaluer son influence sur la rectitude de la conduite. Au cours des essais, les valeurs d'estimation de l'erreur, les délais de calcul entre les boucles d'opération et les valeurs d'angles appliquées aux roues ont été accumulés par l'ordinateur de bord puis enregistrés en format texte pour être par la suite analysés via des logiciels tiers d'analyse comme Matlab et Excel. De manière à quantifier la qualité de la conduite du véhicule, l'écart-type des erreurs mesurées a été évalué pour une large gamme de vitesse. La Figure 16 présente l'écart-type de l'erreur en fonction de la vitesse de conduite. L'écart-type pour chaque cas est représenté par un carré bleu tandis que la courbe de tendance est dépeinte par un trait pointillé rouge. On remarque facilement que l'écart-type varie linéairement en fonction de la vitesse, conséquence directe de la précision longitudinale diminuée par l'augmentation de vitesse. Évidemment, un écart-type minimal est souhaité puisqu'une valeur faible indique peu de déviation latérale.

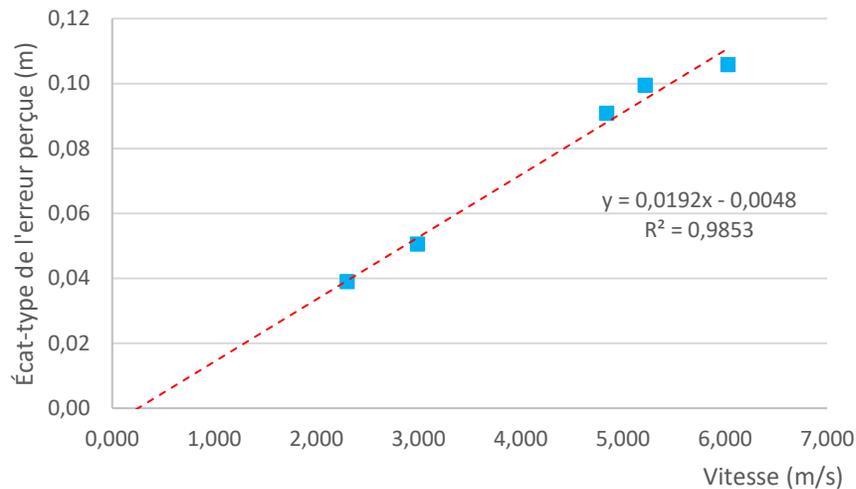


Figure 16 - Écart-type de l'erreur positionnelle mesurée en fonction de la vitesse du véhicule. Données recueillies sur 186.7m pour chaque groupe de vitesse avec une fréquence d'analyse à 40 Hz

Tel que présenté dans la Figure 16, le cas de conduite dont la vitesse est maximale, soit 6,03 m/s représente le cas critique répertorié en raison de de l'écart-type le plus important observé. D'autres valeurs intéressantes peuvent être extraites de la variation de l'erreur perçue comme l'à-coup (de l'anglais jerk), soit la première dérivée de l'accélération par rapport au temps. On définit dans la littérature plusieurs cas de qualification de la conduite automobile grâce à l'à-coup. On s'intéresse plus particulièrement à la valeur maximale observée telle que présentée par Zeng et al. (2018). Toutefois, l'absence de capteur inertiel sur le véhicule prohibe la quantification réelle des accélérations induites sur le robot. L'à-coup n'a donc pas été mesuré dans le cadre de ce projet.

### 5.3.2 Réglage du régulateur médio-latéral

Conformément à l'information mentionnée au cours de l'article publié (Chapitre 4), le réglage du régulateur latéral a été effectué en utilisant la méthode de Ziegler and Nichols (1993). Pour un essai d'analyse en mode « meneur de train » (lead ou pacesetter), les valeurs du régulateur ont été fixées à  $P = 7,3$ ,  $I = 0,12$ ,  $D = 1,4$ . Ce réglage spécifique est celui ayant permis la réalisation de l'essai à vitesse maximale de 6,03 m/s et pour toutes autres vitesses inférieures. Il est néanmoins intéressant d'étudier le comportement du contrôleur en fonction du temps afin d'évaluer sa réponse temporelle. À cet effet, la Figure 17 - Valeurs individuelles des éléments du contrôleur PID pour le contrôle médio-latéral sur un essai de 186.7 mètres à 6 m/s. est présentée ici-bas. La Figure 17 présente les valeurs calculées pour chaque paramètre du régulateur en fonction du temps. Il est à noter que la somme de chacune des composantes correspond à la correction appliquée en degrés aux roues du véhicule pour une erreur mesurée. On remarque que le facteur ayant semblablement le plus d'effet sur le résultat de la boucle de contrôle est celui d'intégration. En effet, malgré sa faible valeur, l'aire sous la courbe verte est beaucoup plus importante que celle des autres paramètres. Par ailleurs, il est intéressant de remarquer que la majorité des valeurs calculées se trouvent dans l'axe négatif, correspondant à une dérive vers la droite du véhicule. La phase initiale du pilotage, soit de 0 à 7 secondes, présente une variabilité majoritairement sans bruit avec des valeurs proportionnelles, dérivées et intégrées relativement faibles. Ce n'est en effet qu'à partir de 7 secondes que l'on remarque un accroissement important de la valeur d'intégration. Selon la vitesse de l'essai, soit 6 m/s, ce point correspond à 42 m à partir du point de départ de l'essai. Cette position équivaut à l'entrée en courbe du véhicule. Ainsi, il est juste de penser que la variation dans le facteur d'intégration est causée par la déviation des lignes (la courbe elle-même). À  $t = 16$  s, en sortie de courbe, le facteur tend à diminuer de nouveau vers le point nul. La valeur minimale observée à 23 secondes, à l'entrée de la deuxième courbe, correspond à une correction d'angle de -0,44 degré. Dès lors, la valeur recommence à diminuer linéairement en raison de la courbe. Pour un essai comprenant un grand nombre de tours, la variable d'intégration prendra de l'ampleur et induira une conduite insoutenable. Pour pallier ce problème, la variable d'intégration pourrait être remise à 0 en sortie de courbe à partir d'un algorithme de reconnaissance de courbure dans la trajectoire. Plus simplement, le terme d'intégration pourrait être remis à zéro lorsque l'erreur du véhicule atteint ou passe par le centre du corridor. Ces travaux ne seront toutefois pas implémentés dans le cadre de ce mémoire puisque la conduite obtenue permet de répondre aux hypothèses posées. Des essais ont été réalisés avec une valeur nulle pour le paramètre d'intégration, mais ceux-ci ont démontré des limitations au niveau du contrôle latéral en courbe à partir de 3,88 m/s, confirmant ainsi l'hypothèse selon laquelle il est le paramètre prédominant.

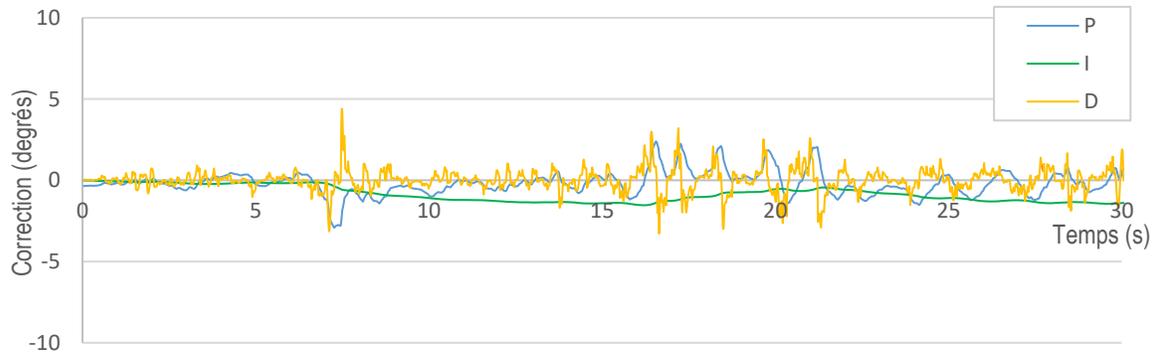


Figure 17 - Valeurs individuelles des éléments du contrôleur PID pour le contrôle médio-latéral sur un essai de 186.7 mètres à 6 m/s.

Afin d'aider la visualisation du contrôle médio-latéral, des résultats sont présentés à la Figure 18 - Position réelle du véhicule pour un essai. La position du véhicule sur un tour de piste tiré d'un essai réel effectué à 4,38 m/s (15,77 km/h). L'amplitude de la déviation médio-latérale du véhicule est représentée par un gradient de couleur. Une déviation positive (vers la gauche) est représentée en rouge tandis qu'une déviation négative (vers la droite) est représentée en bleu. Une erreur nulle est dépeinte par un trait gris.

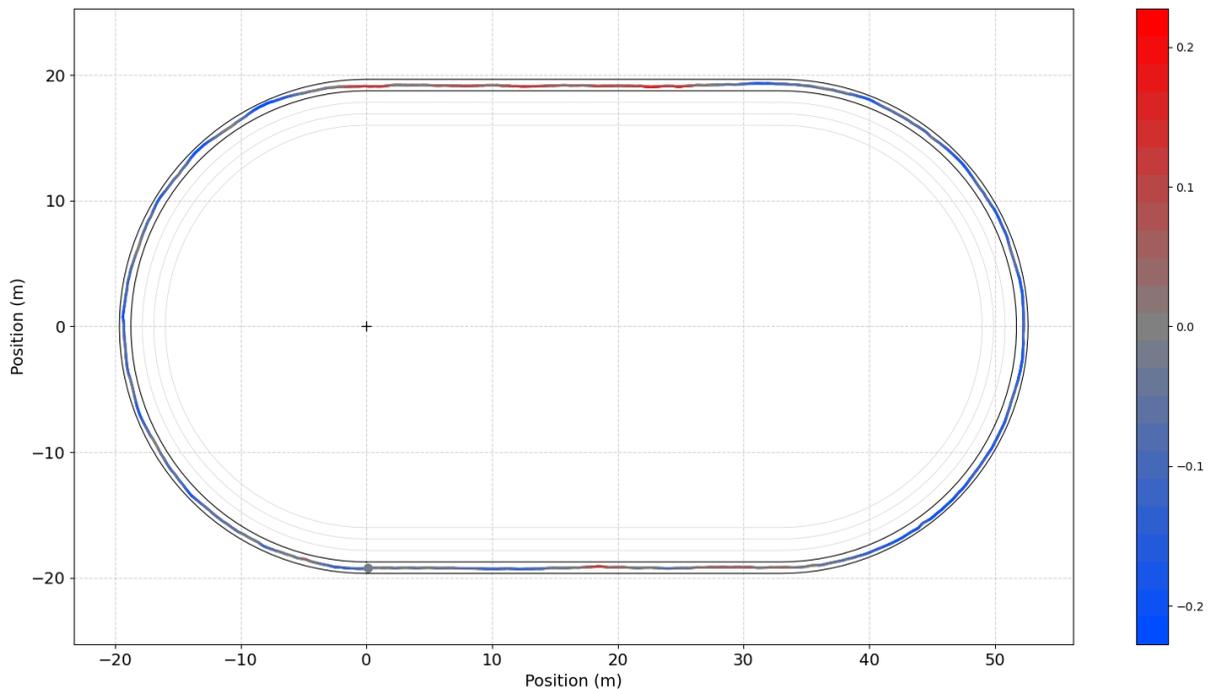


Figure 18 - Position réelle du véhicule pour un essai réel sur piste

### 5.3.3 Réglage du suiveur longitudinal

Puisque le but du projet était de démontrer la faisabilité de l'utilisation d'un système d'analyse ayant pour référentiel une personne en déplacement, il a été nécessaire de développer un système de régulation permettant au véhicule de suivre de façon précise la trajectoire empruntée par un participant mis en situation d'évaluation.

Tel que présenté à divers endroits lors de ce mémoire, le robot fait appel à l'utilisation d'un appareil de mesure de distance de type LiDAR linéaire afin d'estimer sa position relative en tout temps par rapport à la personne suivie. Similairement au régulateur développé pour le contrôle médio-latéral, le contrôle longitudinal, correspondant à l'axe x du véhicule (voir Figure 1) est effectué à l'aide d'un contrôleur de type *PD*. Les valeurs utilisées pour les tests présentés dans ce mémoire correspondent à  $P = 0.002$  et  $D = 0.0001$ . La Figure 19 dépeint la schématisation du contrôle effectué par le véhicule. Ce schéma est identique à celui proposé à la Figure 10 à l'exception de la consigne (la distance souhaitée) et de l'absence du terme d'intégration dans le contrôleur. Le terme d'intégration n'a pas été utilisé afin de simplifier le réglage du contrôle et éviter l'accumulation (windup) possible de l'intégrateur.

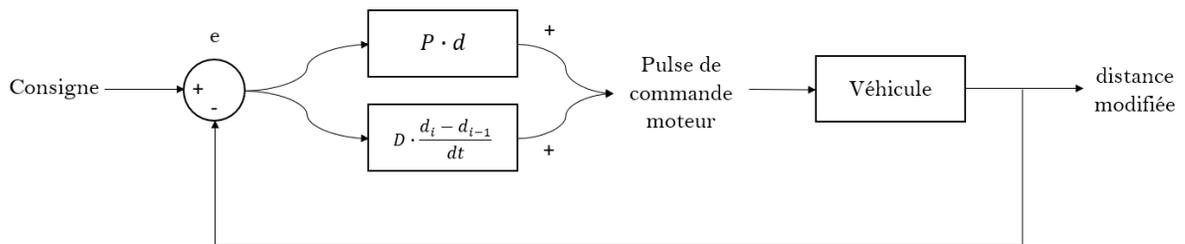


Figure 19 - Schématisation du suiveur longitudinal

La Figure 20 présente les résultats d'estimation de la vitesse d'un participant lors d'un test de marche avec une vitesse croissante en fonction du temps. Lors de ce test, le participant a été invité à se déplacer avec un pas naturel sur 160 m en augmentant légèrement son pas de marche. Le test a été réalisé à l'intérieur du premier corridor de course de la piste intérieure de l'Université depuis la ligne de départ. La vitesse évaluée (en m/s), présentée en bleu montre un grand nombre de corrections pouvant être associées à une valeur proportionnelle trop haute (Clippard, 2019). Une régression linéaire permet de bien visualiser l'augmentation graduelle de la vitesse du patient lors de l'essai. Bien que de nombreuses ondulations soient présentes dans le signal d'acquisition, la droite de régression permet de mieux cibler l'augmentation de la vitesse. La vitesse initiale équivaut à 1 m/s tandis que la vitesse finale est estimée à 1,35 m/s. L'écart-type de la variation de la vitesse sur l'ensemble de l'essai correspond à 0,237 m/s.

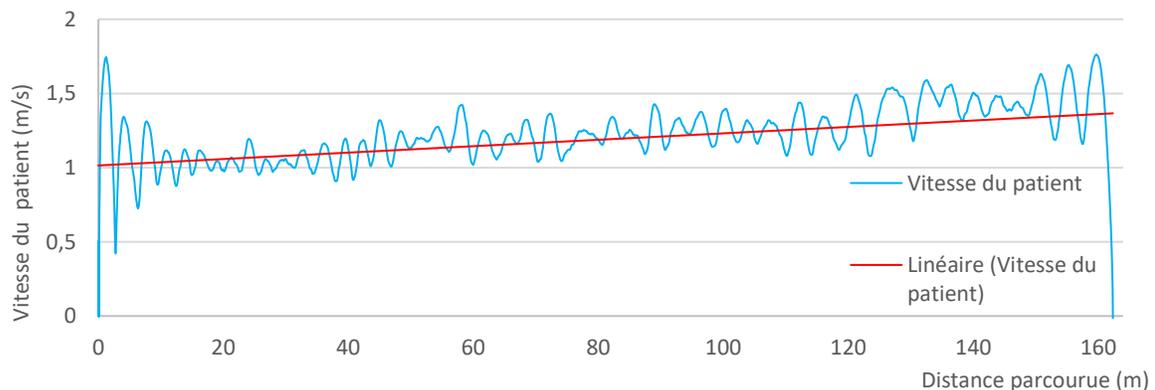


Figure 20 - Analyse de vitesse d'un patient avec augmentation continue de la vitesse sur 160 m

## 5.4 Évaluations comparatives et qualitatives

De façon à valider la fonctionnalité de l'outil développé dans le cadre de ces travaux, des évaluations filmées ont été réalisées. La prochaine section s'intéressera aux résultats visuels obtenus à la suite d'essais effectués sur la piste comparativement à une collecte réalisée en clinique.

### 5.4.1 Essais comparatifs par vidéo

La Figure 21 présente les images captées selon 3 plans distincts (frontal de dos, sagittal de droite et frontal de face) durant un essai de course à 4,45 m/s (16 km/h). Les 3 plans placés en haut de la Figure 21 ont été obtenus à l'aide d'un appareil de captation vidéo à haute définition (1920 x 1080 pixels @ 60 Hz). Le test a été réalisé en laboratoire sur un tapis h/p/cosmos pulsar® 3p (h/p/cosmos, Allemagne). Parallèlement, un essai réalisé sur piste avec le véhicule en mode « meneur de train » à une vitesse de 4,45 m/s a été réalisé afin de comparer qualitativement les images obtenues. Celles-ci ont été enregistrées à partir du même appareil que celui utilisé lors des tests statiques en laboratoire. Il est à noter que le grain de l'image a été modifié en raison de la médiocre luminosité du stade lors de l'essai et de la présence d'importantes vibrations lors du déplacement du véhicule causé par la porosité de la piste elle-même et possiblement une suspension trop rigide. À l'exclusion du grain de l'image légèrement réduite, les résultats obtenus dans les deux essais sont de qualité suffisante pour produire une évaluation biomécanique à l'aide d'une grille d'analyse ou d'un logiciel tiers.



Figure 21 - Comparaison des images obtenues lors de tests cliniques et en milieu écologique sur un essai de course à 16 km/h

# Chapitre 6 Discussion

Cette section abordera les différents choix effectués et résultats obtenus lors de la réalisation de ce projet en plus de présenter les possibles avenues d'amélioration pour des travaux futurs.

## 6.1 Choix du matériel

Les différentes composantes sélectionnées dans le cadre de ce projet ont mené à bien la réalisation des objectifs initialement déterminés. Dans un premier temps, les deux méthodes d'alimentation électrique choisies permettent au système une flexibilité et des durées d'analyses dépassant largement les attentes du projet. En effet, une seule pile pour le véhicule produit une alimentation suffisante sur une période supérieure à 30 minutes à plein régime. La pile sélectionnée pour l'alimentation de l'ordinateur de bord a quant à elle rendu possible la réalisation de plusieurs dizaines d'analyses avant de nécessiter une recharge, supportant ainsi le caractère portatif du projet.

Bien que les performances offertes par la plateforme de véhicule sélectionnée aient également répondu aux différents critères du projet, l'utilisation et l'implémentation d'un programme pouvant communiquer adéquatement avec le matériel électronique d'origine du véhicule se sont avérées difficiles, voire prohibitives, à une recommandation. De nombreuses heures ont dû être déployées afin de contourner le protocole de sécurité liant la télécommande du véhicule et le receveur. D'autres pièces d'équipement disponibles sur le marché auraient permis la mise en fonction du système beaucoup plus rapidement. Néanmoins, la qualité de fabrication du véhicule lui-même aura facilité l'ajout mécanique de composantes en plus d'offrir des performances de conduites au-delà des attentes initiales. Il existe par contre de nombreuses options de véhicule sur le marché ayant une mécanique similaire à un prix beaucoup moins élevé que celui proposé par Traxxas.

Une tierce composante sélectionnée pour le projet se devant de faire partie de cette discussion est l'appareil de mesure de distance de type LiDAR. Les mesures réalisées à l'aide du capteur permettent d'établir adéquatement un suivi dépendant du patient. La précision du capteur couvre largement les besoins du projet en plus de répondre efficacement à la demande en termes de fréquence d'analyse. Par ailleurs, l'implémentation de la librairie python s'est réalisée sans anicroches. Les différents essais ont toutefois démontré que l'utilisation d'un capteur directionnel comme celui-ci pouvait apporter son lot de problèmes. En effet, comme la zone d'échantillonnage du capteur est relativement restreinte, il arrive que le participant ne soit pas repéré en raison d'une déviation médio-latérale trop importante, plus particulièrement en entrée et en sortie de courbe, moments critiques où le véhicule possède un lacet différent du patient, induisant ainsi une lecture ne pointant pas sur le patient. Des travaux supplémentaires permettraient l'ajout d'un système d'alignement pour le capteur de façon à ce que celui-ci pointe toujours en direction du patient. Ce système pourrait être réalisé à l'aide de la caméra

Pixy2, qui contient son propre processeur et programme permettant la reconnaissance d'images. De manière générale, l'utilisation d'un appareil de mesure de distance linéaire comme le Garmin LiDAR Lite V3 apporte son lot de problématiques et pourrait être à l'avenir remplacé par un système ayant une portée médio-latérale plus vaste. Par contre, ces systèmes sont beaucoup plus dispendieux que l'appareil choisi dans le cadre de ce projet.

La caméra utilisée dans la version finale du projet a présenté un atout important à la réalisation des différents objectifs. En effet, l'utilisation du port CSI de l'ordinateur de bord pour la caméra aura permis d'augmenter de façon importante la fréquence d'analyse comparativement à celles obtenues avec une webcam via un port USB. Cette modification aura permis la multiplication de la fréquence d'analyse par quatre, repoussant la vitesse maximale du véhicule un facteur de deux. De plus, l'utilisation d'une caméra web via le port USB en début de projet aura nécessité une grande attention, produit une faible résolution et des instabilités de pilote vidéo nuisant à la réalisation d'analyses. Le module a de plus diminué le montant total du projet par son faible coût par rapport à une caméra web conventionnelle. L'un des points d'amélioration au niveau de la caméra serait plutôt sur le plan de son système d'attache mécanique et sa position. En date d'écriture de ce mémoire, la caméra a été disposée au-dessus du véhicule afin d'augmenter la profondeur du champ de vision. Également, le montage mécanique est réalisé sur un système de pivot permettant l'ajustement angulaire de la caméra. Bien que ces éléments aient favorisé le développement rapide du concept, leurs fonctionnalités dans une approche clinique sont difficilement endossables. Par exemple, la position haute de la caméra augmente de façon importante les vibrations perçues par le capteur. L'utilisation d'un pivot engendre également parfois la modification de la position angulaire de la caméra, ayant ainsi un effet néfaste sur la précision de lecture du capteur lui-même. Pour surmonter ces problèmes, il serait préférable d'introduire un support fixe placé à l'avant du véhicule.

Au niveau de l'ordinateur de bord sélectionné, le RP4 se sera présenté comme un choix éclairé. La grande disponibilité d'information et le support de la communauté sur le produit facilitent son utilisation et sa modification. L'écosystème du RP4 est bien connu et il est facile de trouver des modules et appareils complémentaires à faible coût. Il existe dorénavant une version comportant 8 GB de mémoire vive du RP4, ce qui pourrait éventuellement être modifié dans l'assemblage. Bien que l'ordinateur ait répondu aux différents objectifs du projet, l'utilisation d'une carte micro SD comme mémoire embarquée a ralenti sensiblement le processus comparativement à d'autres appareils similaires comme le Asus Tinkerboard qui comporte de la mémoire embarquée (EEPROM). L'alimentation via le port USB-C aura de plus grandement facilité la sélection d'une unité d'alimentation suffisante.

En raison du manque de précision et de répétabilité des broches de prototypage en ce qui a trait à la génération d'ondes carrées de type PWM, la carte PCA-9685 a été mise à profit. En plus d'être peu coûteuse, cette solution s'est avérée être facile à implémenter et incroyablement stable au cours du projet.

## 6.2 Retour sur la programmation

L'entièreté du projet a été réalisée à l'aide de la version 3.7 de Python. La diversité des outils offerts dans le langage aura permis l'accomplissement des objectifs de ce projet. Plus particulièrement, la librairie OpenCV, utilisée pour effectuer le traitement de l'image, propose des outils ayant limité la lourdeur de la programmation et augmenté la vitesse de la boucle de contrôle par ses fonctions compilées en langage C++. Il est par contre important de dénoter que l'utilisation de OpenCV sur un RP4 comporte son plusieurs défis. En effet, avant d'être utilisée, la librairie doit être compilée, chose relativement fastidieuse sur le système d'exploitation du RP4. Les travaux de Rosebrock (2018) via la page web Pyimagesearch sont largement responsables de la réussite de cette section du projet.

Dans le domaine de la conduite autonome de véhicules, la rapidité d'exécution de la séquence de contrôle est de très grande importance. Le choix de Python 3.7 dans le cadre de ce projet peut sembler contre-intuitif. Comme démontré par Brihadiswaran (2020) dans un article du magazine Medium, certains scripts Python peuvent être de 50 à 60 fois plus lents que leur équivalent en C++ tout dépendant du compilateur utilisé. À la lueur de ces informations, il semble difficile de justifier le choix de Python comme langage de prédilection pour un système aussi sensible à la vitesse d'exécution. Néanmoins, le projet VAALÉRIE se veut le développement d'un outil à accès libre pouvant être modifié et adapté par plusieurs chercheurs selon leurs besoins. Bien que très rapide à l'exécution, le développement d'un programme en C++ est un exercice chronophage limitant les itérations rapides. De ce fait, le Java pourrait représenter un compromis intéressant. Au cours des dernières années, Python a connu un bond de popularité le rendant un des langages les plus utilisés au monde au profit du Java. Ainsi, afin de rejoindre le plus grand nombre de personnes possibles, il a été décidé de réaliser la programmation dans son entièreté en Python. Par ailleurs, comme démontré dans la section Résultats, le programme Python est tout de même parvenu à effectuer des analyses à une fréquence largement suffisante pour la conduite autonome.

## 6.3 Contrôle

L'utilisation adéquate des systèmes de contrôle dans un projet comme celui-ci est l'un des éléments décisifs quant à sa réussite. Le cadre de ces travaux a impliqué l'implémentation de deux systèmes de contrôle distincts. Le premier, un contrôle de type suiveur, permet l'asservissement de la distance maintenue entre le véhicule et le patient. Dans la phase actuelle du projet, un régulateur  $PD$  a été mis à profit. Toutefois, cette décision s'est avérée problématique en raison du bruit du capteur, la perte occasionnelle du signal réel en raison du déplacement de la personne en dehors du champ de lecture du système et du comportement de la motorisation du véhicule. C'est principalement le paramètre de dérivation qui s'est avéré problématique. En effet, la présence de faible bruit dans la lecture peut induire de grandes disparités dans le résultat du calcul de la vitesse. Ainsi, le

facteur induit par moment une modification importante du contrôle si la commande de sortie n'est pas bien filtrée. Une évaluation plus approfondie des comportements du capteur LiDAR et de la motorisation du véhicule pourrait permettre la mise en place de paramètres plus adaptés ou même modifier le type de régulateur utilisé. Le deuxième régulateur utilisé pour le maintien du contrôle médio-latéral a quant à lui été implémenté à l'aide d'un régulateur de type *PID*. Le régulateur aurait pu fournir directement la valeur de durée de pulse plutôt qu'un angle devant être converti, mais cette pratique aurait grandement diminué l'aisance de compréhension du code et limité le débogage du code lorsque nécessaire. De futurs travaux pourraient contourner cette étape afin de gagner du temps de calcul. De plus, en raison de la vaste plage de vitesse désirée pour le véhicule (0 à 22 km/h), de la nature non linéaire du système et du caractère cyclique de la piste d'athlétisme, l'utilisation d'un seul régulateur aura été un défi de taille. En effet, les essais ont démontré que différentes valeurs de paramètres en fonction de la vitesse permettaient d'améliorer le comportement général du véhicule selon la vitesse de déplacement. Qui plus est, l'implémentation de plusieurs régulateurs linéaires pour répondre à la nature non linéaire du système est une avenue plus ou moins élégante. Afin de contourner ce problème, un régulateur à logique floue pourrait être mis en place. De plus, un *Model Predictive Control* (MPC) pourrait représenter une avenue plus près des besoins du projet bien que ce modèle imposerait une modification importante de l'électronique du véhicule en raison de sa complexité de calcul (Borrelli, Falcone, Keviczky, Asgari, & Hrovat, 2005). Les résultats obtenus à l'aide d'un seul régulateur sont tout de même satisfaisants dans le cadre des objectifs du projet.

## **6.4 Portée dans l'analyse de la condition physique**

L'analyse de la marche ou de déplacement dans un terme plus large couvre une très vaste plage de vitesses. Bien que l'orientation première de ce projet soit la locomotion, soit le déplacement de base cherchant à limiter les dépenses énergétiques, la portée de l'utilisation de cet outil s'étend à bien plus. En effet, le robot pourrait être également utilisé à des fins d'analyse de la performance par la réalisation de différents tests: pensons par exemple à l'évaluation de la vitesse maximale anaérobie (VMA) par le test Léger-Boucher. Selon l'*American College of Sport Medicine* (ACSM), un  $VO_2\text{max}$  de 66,30, correspondant au 95<sup>e</sup> percentile, toutes populations confondues, équivaut à une vitesse maximale anaérobie de 18,82 km/h (5,23 m/s, 3 min 11,50 sec/km) (American College of Sports Medicine, 2017). En considérant cette valeur, il a été déterminé que la marque des 20 km/h (5,60 m/s) serait l'objectif de vitesse maximale du robot, permettant ainsi d'effectuer des analyses sur la grande majorité de la population. Conformément aux résultats obtenus, la vitesse moyenne maximale actuelle de 6,03 m/s permet amplement de couvrir cette gamme.

## 6.6 Applications futures

Les applications et modalités d'utilisation futures de ce projet sont nombreuses. Dans une vision d'entraînement sportif, le mode « *meneur de train* », socialement référé au nom de lapin de course, a été bonifié à l'aide d'un module de contrôle de la vitesse. Ce module permet d'établir des instructions de contrôle de puissance au moteur en fonction du temps. Ces instructions sont définies à l'aide d'une interface graphique orientée pour les utilisateurs permettant de contrôler par des entrées clavier ou par des déplacements de points avec le curseur directement dans le graphique les objectifs de vitesse du véhicule dans le temps. Selon les modalités d'entraînement désirées, il est possible de générer des fonctions de vitesse en fonction du temps à pallier ou selon une spline cubique passant par les points souhaités.

### 6.6.1 Test-Léger-Boucher

Le module de commande par affichage a également permis la mise en place de protocoles d'évaluations destinés à l'évaluation de la condition physique des athlètes. En effet, l'équipe s'est intéressée à l'utilisation du véhicule dans le cadre d'un test de course permettant de quantifier les performances sportives et la condition physique plus générale. En raison de sa compatibilité avec les critères de ce projet, les balbutiements de l'essai d'un test Léger-Boucher (Léger & Boucher, 1980) permettant l'estimation de la vitesse maximale anaérobie (VMA) et de la consommation maximale d'oxygène ( $VO_{2max}$ ) ont été entrepris. L'avantage premier de cette approche est la discrimination constante effectuée par le robot quant à la réussite d'un palier. En effet, en imposant le pas de course et en mesurant de façon accrue la position du coureur, il est théoriquement possible de déterminer de façon exacte le moment de décrochage de l'athlète lors du test. La Figure 22 présente l'interface graphique utilisée pour la réalisation d'un test Léger-Boucher à l'aide du véhicule. On dénote la présence des 24 paliers du test avec l'accroissement de la vitesse de 0,277 m/s (1 km/h) de celle-ci à chaque 120 secondes. Les valeurs de vitesses et de position peuvent être modifiées par un clic de la souris ou à l'aide du module d'entrées manuelles (gauche). Les résultats préliminaires de cette approche ont été présentés dans le cadre du 26e congrès de l'Association québécoise des Sciences de l'Activité Physique (AQSAP) (Bégin & Lavallière, 2021).

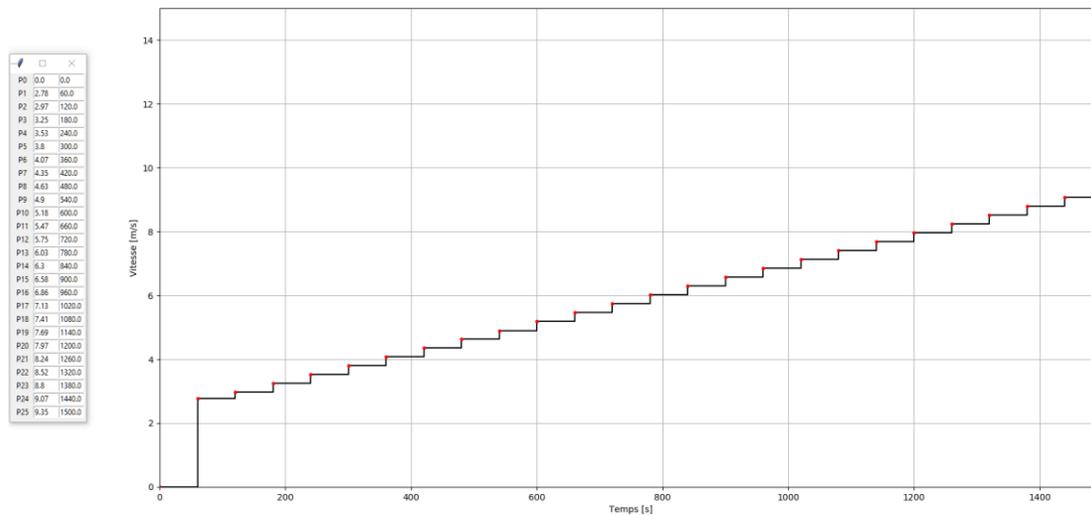


Figure 22 - Instructions de vitesse en fonction du temps pour un exemple de test Léger-Boucher réalisé par VAALÉRIE.

### 6.6.2 Test de marche de 6 minutes (6 Minute Walk Test)

Largement utilisé dans le domaine de la santé afin de documenter le niveau de capacités fonctionnelles de patients présentant diverses atteintes de santé, le test de marche de 6 minutes évalue le déplacement maximal pouvant être effectué par un patient à la marche sur une période de 360s (6 minutes). Le test est généralement réalisé par aller-retour entre deux plots distancés de 30 mètres. Néanmoins, peu d'espaces permettent l'obtention d'une telle distance et les cliniciens optent généralement pour la distance maximale pouvant être couverte à l'intérieur de leurs installations. Les critères d'arrêt du test sont le temps de marche (360 s) ou une distance (en mètres) lorsqu'une fatigue importante ne permet plus au patient de progresser dans l'évaluation. Dans une optique d'entraînement et de réadaptation physique, le système a été modifié afin de permettre la réalisation du test de marche de 6 minutes selon des conditions légèrement différentes. Conformément aux spécifications du projet, le test doit être effectué de façon continue sur une piste d'athlétisme à l'intérieur d'un corridor. De manière à documenter plus adéquatement l'évaluation, la programmation a été modifiée afin de recueillir des données concernant la vitesse du patient en temps réel, la distance parcourue à tout moment et le temps en déplacement dans un cas d'arrêt hâtif du test. Ces informations permettront certainement dans le futur de mieux documenter l'évolution de l'effet de l'entraînement ou la réadaptation physique des patients symptomatiques en plus de fournir des informations pertinentes quant à la réalisation du test. Par exemple, les données permettent d'identifier le point de rupture dans la vitesse de marche ou même d'évaluer la variation de vitesse au cours du test. Par ailleurs, des travaux supplémentaires permettront de détecter la cadence de pas du patient à l'aide des données du LiDAR. Des résultats probants sont en cours d'analyse, mais dépassent largement le cadre de ce mémoire. Les résultats préliminaires de cette approche ont été présentés sous forme d'affiche dans le cadre des conférences organisées par le regroupement du REPAR (Bégin, Da Silva, Beaulieu,

& Lavallière, 2021). La Figure 23 présente la vitesse en temps réel d'un extrait d'un tour de piste lors d'un test de marche de 6 minutes. Un trait violet correspond à une vitesse nulle tandis qu'une vitesse de 2,50 m/s est représentée par un trait jaune. L'erreur médio-latérale du véhicule a également été conservée sur la figure afin de présenter la précision du système.

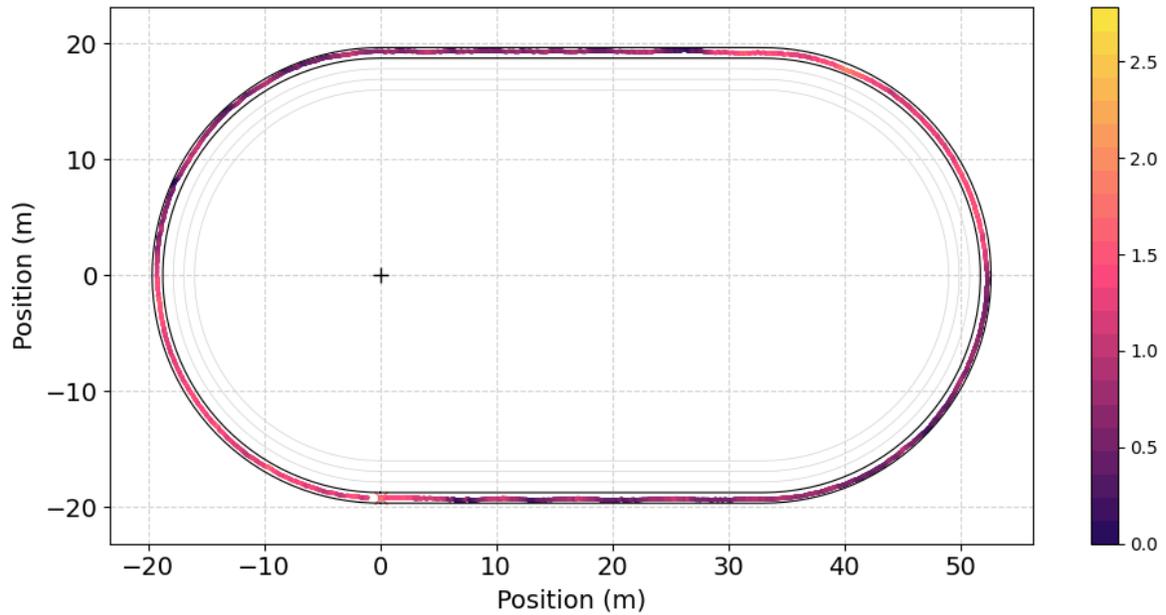


Figure 23 - Heatmap de la vitesse d'un patient en m/s lors d'un tour de piste de 180 m sur un test de marche de 6 minutes.

## Conclusion

Les travaux réalisés dans le cadre de ce mémoire ont permis le développement d'une preuve de concept permettant l'évaluation de certains paramètres de la locomotion humaine en milieu plus écologique. La démarche entreprise et les équipements sélectionnés ont permis la réalisation d'un robot capable de se déplacer de manière autonome à l'intérieur des lignes d'un corridor d'une piste d'athlétisme et ce pour une large gamme de vitesses. Les essais effectués sur piste ont permis d'établir la capacité du prototype à maintenir une trajectoire suffisamment rectiligne, et ce, même à haute vitesse, pour la collecte de vidéos utiles à l'analyse de la locomotion. Les composants électroniques utilisées ont également permis la réalisation de conduite autonome en vitesse adaptative basée sur le positionnement et la vitesse d'un participant sans intervention physique. Il reste à ce jour de nombreuses pistes pouvant être explorées à la suite des travaux réalisés. Une amélioration du système de contrôle et de la suite de capteurs sur le véhicule pourrait ouvrir une nouvelle voie vers l'analyse en milieu encore plus écologique. En effet, les dernières avancées en intelligence artificielle et la connaissance actuelle dans la théorie du contrôle laissent croire qu'il serait possible de pousser le système vers des environnements extérieurs et des résidences privées dans un futur rapproché pour la réalisation d'études dont l'impact pourrait être avoir une grande portée en documentant un milieu encore mal compris à ce jour.

# Bibliographie

- Abed, M. E., Ammar, H. H., & Shalaby, R. (2020). *Steering Control for Autonomous Vehicles Using PID Control with Gradient Descent Tuning and Behavioral Cloning*. Paper presented at the 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES).
- American College of Sports Medicine, R., D., Ehrman, J.K., Liguori, G. and Magal, M. (2017). *ACSM's Guidelines for Exercise Testing and Prescription*: Wolters Kluwer.
- Balaji, B., Mallya, S., Genc, S., Gupta, S., Dirac, L., Khare, V., . . . Karuppasamy, D. (2019). DeepRacer: Educational Autonomous Racing Platform for Experimentation with Sim2Real Reinforced Learning. *Arxiv*.
- Bechtel G., M., McElhiney, E., Kim, M., & Yun, H. (2018). *DeepPicar: A Low-cost Deep Neural Network-based Autonomous Car*. University of Kansas, Indiana University, USA.
- Bégin, W., Da Silva, R., Beaulieu, L.-D., & Lavallière, M. (2021). *Analyse de la locomotion lors d'un test de marche de 6 minutes par un véhicule à guidage automatique: Étude de faisabilité technologique*. Paper presented at the Journée scientifique REPAR-INTER Sherbrooke, QC. <https://event.fourwaves.com/repair-inter2021/abstracts/99c26013-3a66-416a-85fd-5467e70e4748>
- Bégin, W., Duquette, S., & Lavallière, M. (2020). Real Application for a Low-Cost Low-Power Self-Driving 1/10 Scale Car. *Journal of Robotics and Automation*, 4(1), 195-201. doi:10.36959/673/370
- Bégin, W., & Lavallière, M. (2021). *VAALÉRIE, une nouvelle approche pour l'évaluation des déterminants de la performance de la course à pied*. Paper presented at the 26e congrès de l'Association québécoise des Sciences de l'Activité Physique (AQSAP), Québec, QC, (Virtuel). <https://aqsapcongres.wordpress.com/programme-2021-journee-du-28-avril/>
- Bojarski, M., Firner, B., Flepp, B., Jackel, L., Muller, U., Zieba, K., & Del Testa, D. (2016). *End -to-end Deep Learning for Self-Driving Cars*. Blog post. NVIDIA Deloper Blog. Retrieved from <https://developer.nvidia.com/blog/deep-learning-self-driving-cars/>
- Borrelli, F., Falcone, P., Keviczky, T., Asgari, J., & Hrovat, D. (2005). MPC-based approach to active steering for autonomous vehicle systems. *International journal of vehicle autonomous systems*, 3(2-4), 265-291.
- Brihadiswaran, G. (2020, 23 July). A performance Comparison Between C, Java, and Python. *Medium*. Retrieved from <https://medium.com/swlh/a-performance-comparison-between-c-java-and-python-df3890545f6d>
- Canny, J. (1986). A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell*, 8(6), 679-698. Retrieved from <https://www.ncbi.nlm.nih.gov/pubmed/21869365>
- Carreira, J., Agrawal, P., Fragkiadaki, K., & Malik, J. (2016). *Human pose estimation with iterative error feedback*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Clippard. (2019). Practical PID Process Dynamics with Proportional Pressure Controllers. Retrieved from <https://clippard.com/cms/wiki/practical-pid-process-dynamics-proportional-pressure-controllers>
- Cuesta-Vargas, A. I., Galán-Mercant, A., & Williams, J. M. (2010). The use of inertial sensors system for human motion analysis. *Physical Therapy Reviews*, 15(6), 462-473.
- Humes, L. E., Bussey, T. A., Craig, T. A., & Kewley-Port, D. (2009). The effects of age on sensory thresholds and temporal gap detection in hearing, vision and touch. *Attention, Perception & Psychophysics*, 71, 860-871. doi:10.3758/APP.71.4.860
- Kathuria, A. (2018, 2018-04-23). What's new in YOLO v3. Retrieved from <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>
- Koubâa, A. (2017). *Robot Operating System (ROS) (Vol. 1)*: Springer.
- Kun Law, C., Dalal, D., & Shearrow, S. (2018). *Robust Model Predictive Control for Autonomous Vehicule/Self-Driving Cars*. arXiv.org.
- Lamine, H., Bennour, S., Laribi, M., Romdhane, L., & Zaghoul, S. (2017). Evaluation of Calibrated Kinect Gait Kinematics Using a Vicon Motion Capture System. *Computer Methods in Biomechanics and Biomedical Engineering*, 20, 111-112. doi:10.1080/10255842.2017.1382886

- Lebel, K. (2017). *Validation, optimisation et valorisation de la mesure d'orientation issue de centrales inertielles pour la biomécanique clinique*. Université du Québec à Montréal,
- Léger, L., & Boucher, R. (1980). An indirect continuous running multistage field test: the Université de Montréal Track Test. *Canadian Journal of applied sport sciences*, 77-84.
- MathWorks. (2018). Understanding PID Control. Retrieved from [https://www.youtube.com/watch?v=wkfEZmsQqiA&list=PLn8PRpmsu08pQBqixYFXSsODEF3Jqmm-y&ab\\_channel=MATLAB](https://www.youtube.com/watch?v=wkfEZmsQqiA&list=PLn8PRpmsu08pQBqixYFXSsODEF3Jqmm-y&ab_channel=MATLAB)
- Merriau, P., Dupuis, Y., Boutteau, R., Vasseur, P., & Savatier, X. (2017). A Study of Vicon System Positioning Performance. *Sensors (Basel)*, 17(7). doi:10.3390/s17071591
- Müller, M., & Koltun, V. (2020). OpenBot: Turning Smartphones into Robots. *arXivLabs*, 17. Retrieved from <https://arxiv.org/pdf/2008.10631.pdf>
- Polack, P., Althé, F., d'Andréa-Novel, B., & de La Fortelle, A. (2017). *The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?* Paper presented at the 2017 IEEE intelligent vehicles symposium (IV).
- Rajamani, R. (2006). *Vehicule Dynamics and Control*. New-York, USA: Springer.
- Raksincharoensak, P., Nagai, M., & Shino, M. (2006). Lane keeping control strategy with direct yaw moment control input by considering dynamics of electric vehicle. *International Journal of Vehicle Mechanics and Mobility*, 44, 192-201. doi:10.1080/00423110600870006
- Ramakrishnan, T., Kim, S. H., & Reed, K. B. (2019). Human Gait Analysis Metric for Gait Retraining. *Appl Bionics Biomech*, 2019, 1286864. doi:10.1155/2019/1286864
- Rigby Taylor. (2019, 18 September 2019). Line Marking Measurements - Athletics 200m Track. Retrieved from [https://www.rigbytaylor.com/wp-content/uploads/LineMarkings\\_Athletics200m.pdf](https://www.rigbytaylor.com/wp-content/uploads/LineMarkings_Athletics200m.pdf)
- Rosebrock, A. (2018, April 17, 2021). Pyimagesearch. Retrieved from <https://www.pyimagesearch.com/>
- S. Srinivasa, S., Lancaster, P., Michalove, J., Schmittle, M., Summers, C., Rockett, M., . . . Sadeghi, F. (2019). MuSHR: A Low-Cost, Open-Source Robotic Racecar for Education and Research. *arXivLabs*, 4.
- Steyvers, F. J. J. M., & Waard, D. D. (2000). Road-edge delineation in rural areas: effects on driving behaviour. *Ergonomics*, 43(2), 223-238. doi:10.1080/001401300184576
- Stranden, J. E. (2019). *Autonomous driving of a small-scale electric truck model with dynamic wireless charging*. Norwegian University of Science and Technology, Retrieved from [https://drive.google.com/file/d/1z\\_e1Ti49YLCHxgyXCcEzL6ghCVWzpjO/view](https://drive.google.com/file/d/1z_e1Ti49YLCHxgyXCcEzL6ghCVWzpjO/view)
- Toshev, A., & Szegedy, C. (2014). *DeepPose: Human pose estimation via deep neural networks*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Veilleux, L. N., Raison, M., Rauch, F., Robert, M., & Ballaz, L. (2016). Agreement of spatio-temporal gait parameters between a vertical ground reaction force decomposition algorithm and a motion capture system. *Gait Posture*, 43, 257-264. doi:10.1016/j.gaitpost.2015.10.007
- Webster, K. E., Wittwer, J. E., & Feller, J. A. (2005). Validity of the GAITRite walkway system for the measurement of averaged and individual step parameters of gait. *Gait Posture*, 22(4), 317-321. doi:10.1016/j.gaitpost.2004.10.005
- Zeng, X., Cui, H., Song, D., Yang, N., Liu, T., Chen, H., . . . Lei, Y. (2018). Jerk Analysis of a Power-Split Hybrid Electric Vehicule Based on a Data-Driven Vehicule Dynamics Model. *Energies*, 20. doi:10.3390/en11061537
- Zhou, H., & Hu, H. (2008). Human motion tracking for rehabilitation—A survey. *Biomedical signal processing and control*, 3(1), 1-18.
- Ziegler, J. G., & Nichols, N. B. (1993). Optimum Settings for Automatic Controllers. *J. Dyn. Sys., Meas., Control.*, 115(2B). doi:10.1115/1.2899060

## Annexe A Connexion au robot via le protocole SSH

De manière à établir une communication entre le robot et le chercheur, le protocole *Secure Shell (SSH)* a été utilisé. Ce protocole permet d'établir un lien de contrôle entre deux ordinateurs par le biais d'un réseau Wi-Fi. Il est possible d'utiliser le protocole *SSH* à l'aide de l'invite de commandes Windows, Linux ou Mac OS. Sous Windows, l'application installée par défaut *Remote Desktop* permet également l'utilisation du protocole en plus de partager l'écran de l'ordinateur dont on demande le contrôle. Afin de rendre l'utilisation du robot le plus facile possible, l'utilisation de l'application *Remote Desktop* a été priorisée dans le cadre de ce projet. Comme la connexion au protocole SSH requiert la connaissance de l'adresse IP de l'ordinateur dont on souhaite prendre contrôle et qu'il est parfois difficile d'accéder à cette information sur des réseaux sécurisés, la carte réseau de l'ordinateur de bord du véhicule a été programmée de façon à générer un réseau Wi-Fi via lequel la communication est établie. L'annexe suivante détaille les étapes permettant la mise en communication entre VAALERIE et un poste de travail Windows.

### A.1 Réseau sans-fils VAALERIE

Lors du démarrage du véhicule, c'est-à-dire lors du branchement de l'ordinateur de bord, un réseau sans-fils du nom de *Vaaleries\_Network* se met automatiquement en marche. Comme le réseau est non sécurisé, il est facile de s'y connecter à l'aide de l'interface de paramètres de connexion au réseau de Windows. Une fois la connexion établie au réseau sans-fils dédiés, l'application *Remote Desktop* peut être lancée par l'utilisateur.

### A.2 Remote Desktop Connection

Dès son lancement, l'application affiche une page d'accueil sur laquelle l'ordinateur sur lequel on désire effectuer une connexion doit être identifié (voir label *Computer* sur la **Erreur ! Source du renvoi introuvable.**). De manière générale, il est possible d'utiliser l'adresse IP pour y arriver, mais il est également possible d'utiliser le nom de l'appareil. Pour des raisons évidentes, l'ordinateur de bord a été nommé « *vaalerie* ». En entrant ce nom, il est possible de lancer la connexion en cliquant sur *Connect*. Le client est par la suite accueilli par une page réservée à une connexion sécurisée. Le type de session utilisé est Xorg tandis que le nom d'utilisateur correspond à « *pi* ». La **Erreur ! Source du renvoi introuvable.** présente le panneau d'affichage de la connexion sécurisée avec les différents labels devant être remplis. Le bouton « OK » permet de finaliser la connexion.

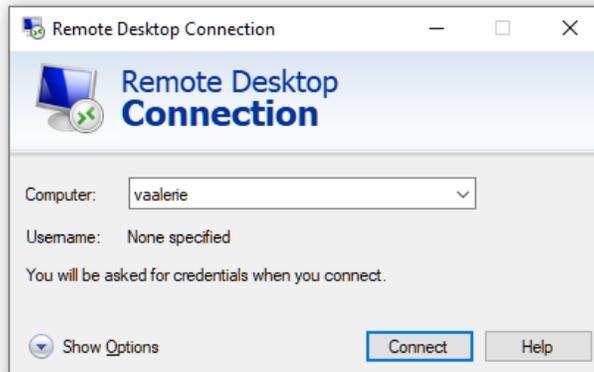


Figure A - 1 Panneau d'accueil de Remote Desktop Connection

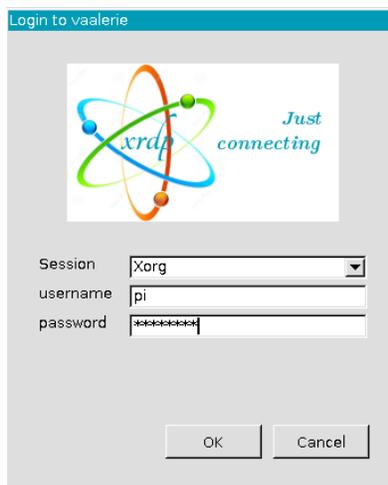


Figure A - 2 Panneau d'accueil de connexion sécurisée

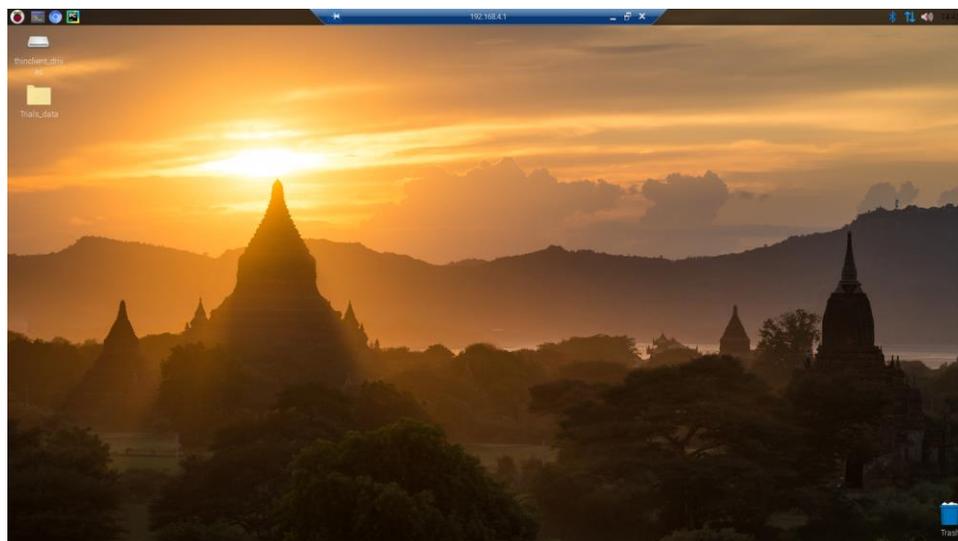
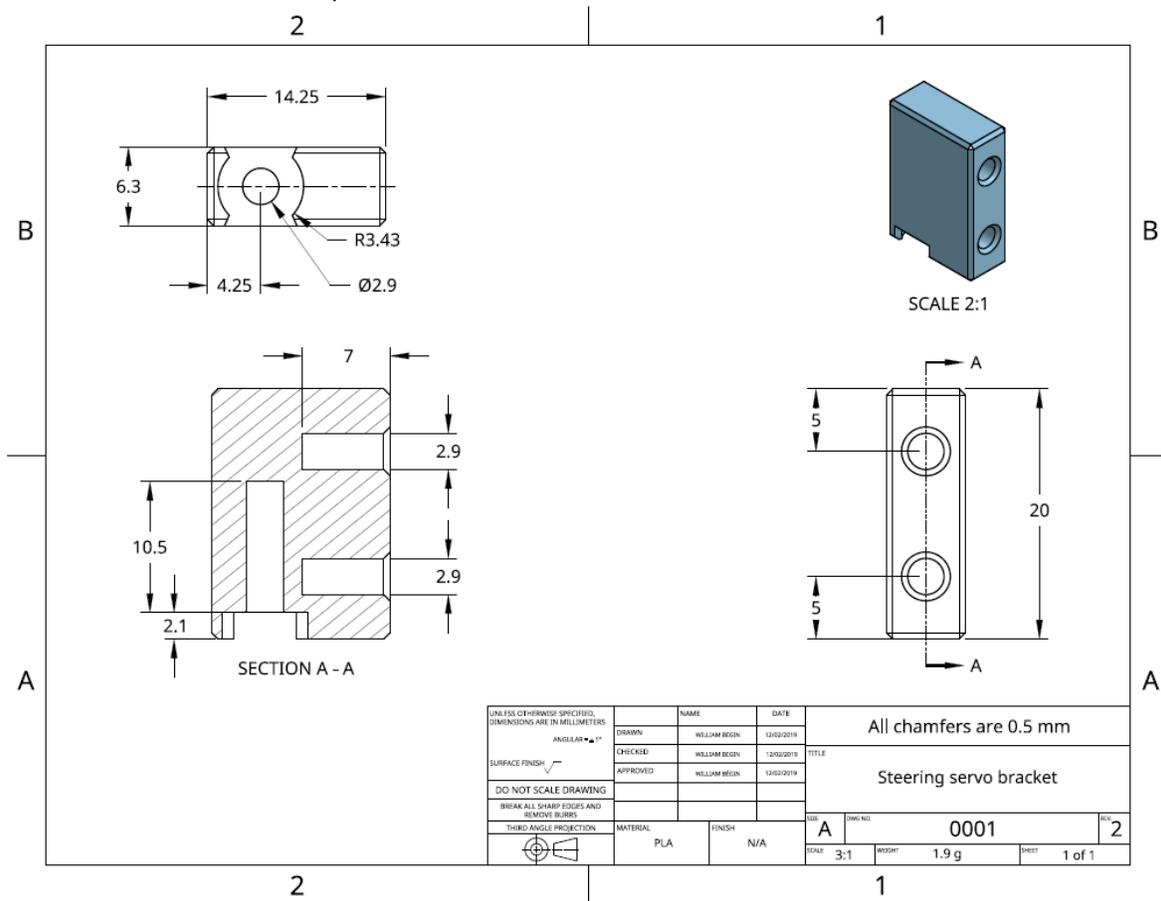
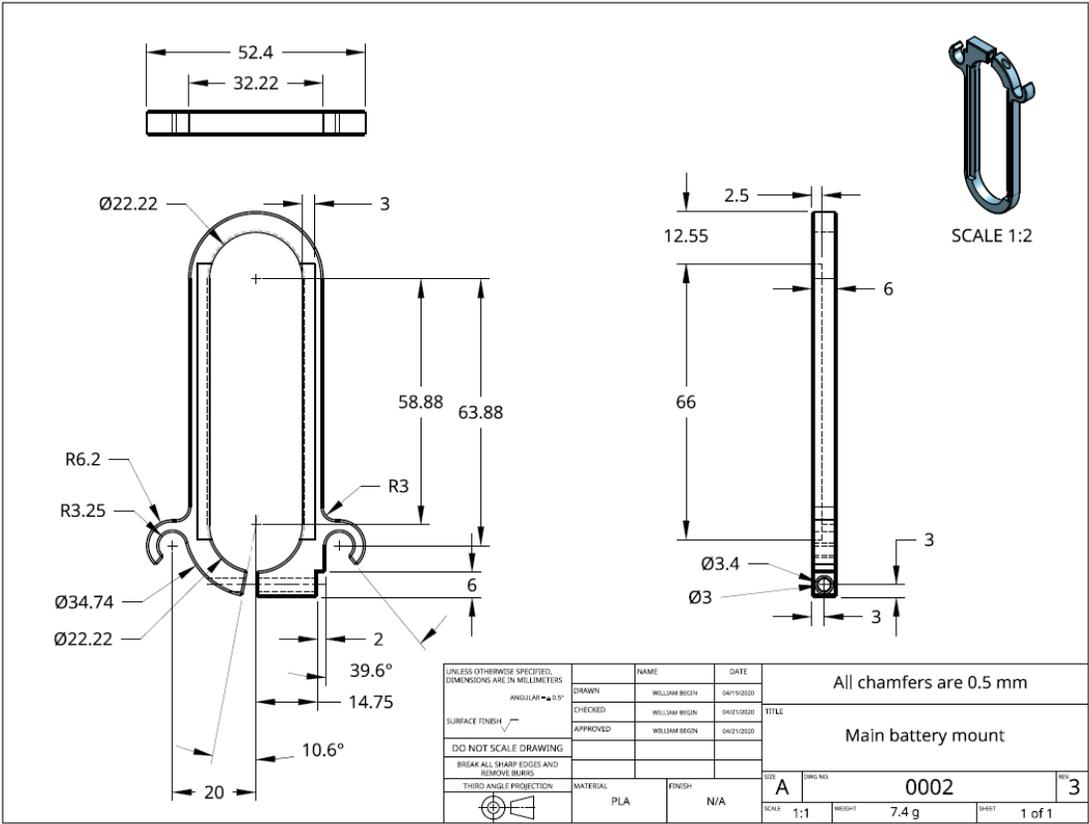


Figure A - 3 Bureau de VAALERIE (RPOS)

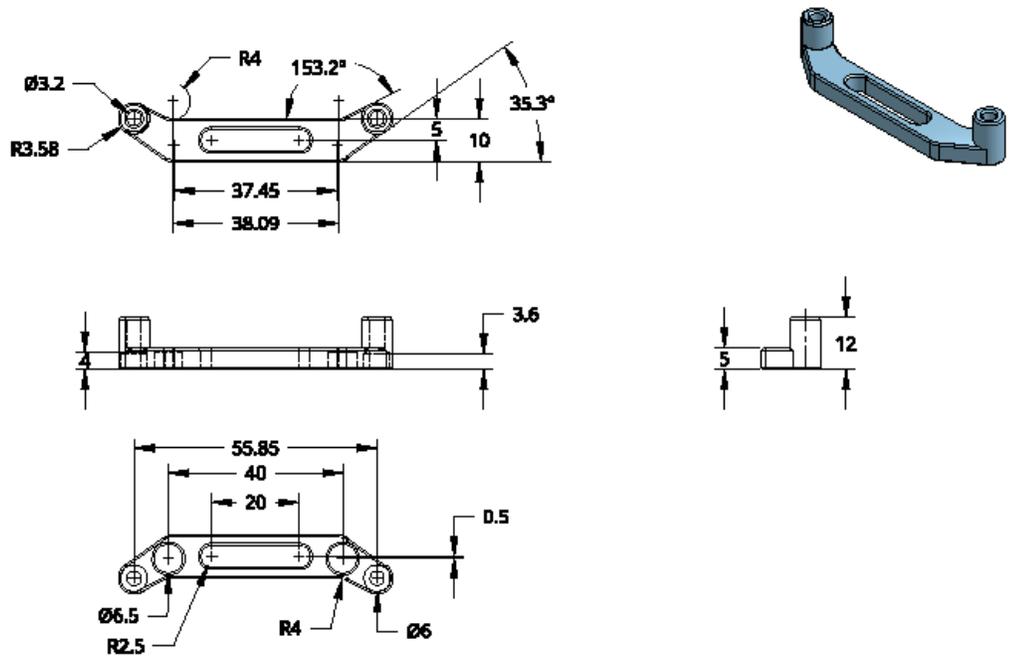
# Annexe B Cahier de dessins

N.B. Le ratio des dessins n'est pas conservé









DESIGNED BY	DATE	DATE	All chamfers are 0.5 mm										
CHECKED BY	DATE	DATE											
APPROVED BY	DATE	DATE	Rebar front adapter										
REVISION	DESCRIPTION	DATE											
1													
<table border="1"> <tr> <td>SCALE</td> <td>1:1</td> </tr> <tr> <td>DATE</td> <td>2.8.0</td> </tr> <tr> <td>PAGE</td> <td>1 of 1</td> </tr> </table>			SCALE	1:1	DATE	2.8.0	PAGE	1 of 1	<table border="1"> <tr> <td>PROJECT</td> <td>0004</td> </tr> <tr> <td>REV</td> <td>2</td> </tr> </table>	PROJECT	0004	REV	2
SCALE	1:1												
DATE	2.8.0												
PAGE	1 of 1												
PROJECT	0004												
REV	2												



# Annexe C Plan du Pavillon sportif de l'UQAC

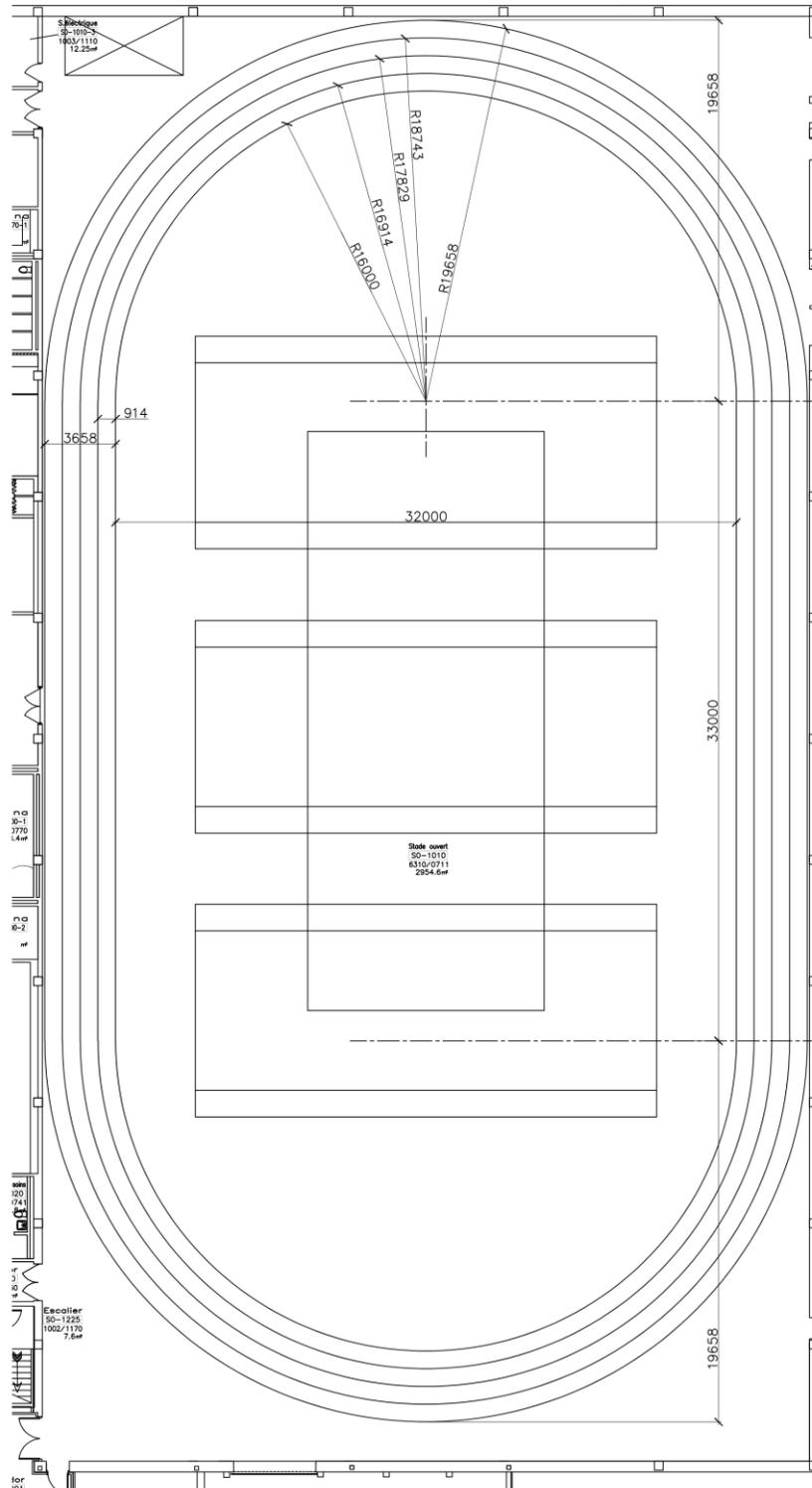


Figure C - 1 Plan de la piste d'athlétisme du pavillon sportif de l'Université



# Annexe E Cinématique du véhicule

Afin de contrôler adéquatement le véhicule et interpréter de façon réaliste son comportement, un modèle mathématique basé sur des données empiriques a été développé. De nombreux ouvrages traitant de la conduite autonome font appel au modèle de la bicyclette (Rajamani, 2006) afin de représenter la cinématique d'un véhicule. Cette section présente ce modèle en plus de décrire les limitations de ce dernier dans son application pour le présent projet. Une version améliorée du modèle est par la suite proposée et évaluée à la suite d'une évaluation empirique. Les variables utilisées au cours de cette annexe sont présentées ci-bas.

$l_r$  = Distance entre l'essieu arrière et le CM  
 $l_f$  = Distance entre l'essieu avant et le CM  
 $C$  = Centre de rotation instantané (CRI)  
 $\delta$  = Angle de direction des roues  
 $\theta$  = Angle complémentaire de direction  
 $\alpha$  = Angle de changement de direction  
 $\beta$  = Angle du CRI  
 $R_0$  = Rayon du centre de rotation instantané au CM  
 $R_1$  = Distance du centre de rotation instantané à l'axe central  
 $i$  = Ajustement de compatibilité géométrique  
 $V$  = Vitesse du véhicule  
 $\dot{\psi}$  = Vitesse angulaire du véhicule  
 $\psi$  = Lacet du véhicule  
 $x$  = Position du centre de masse selon l'axe longitudinal  
 $y$  = Position du centre de masse selon l'axe transversal  
 $L$  = Déplacement du véhicule  
 $e$  = Variation de la position sur l'erreur totale

## 2.1 Modèle de la bicyclette

La représentation physique du comportement d'un véhicule à quatre roues présente un défi de taille et apporte une lourdeur mathématique ne permettant généralement pas d'améliorer les prévisions d'un modèle plus simple. Afin de simplifier les calculs de prévision de position d'un véhicule, la littérature fait généralement appel au modèle simplifié de la bicyclette. Ce modèle permet de représenter les deux roues avant d'un véhicule, responsable du contrôle de la direction, comme une seule roue agissant sur le contrôle en plus de joindre les deux roues arrière en une seule, formant ainsi un assemblage à deux roues similaires à une bicyclette. La **Erreur ! Source du renvoi introuvable.** présentée ici-bas montre la disposition géométrique du modèle ainsi que ses variables associées.

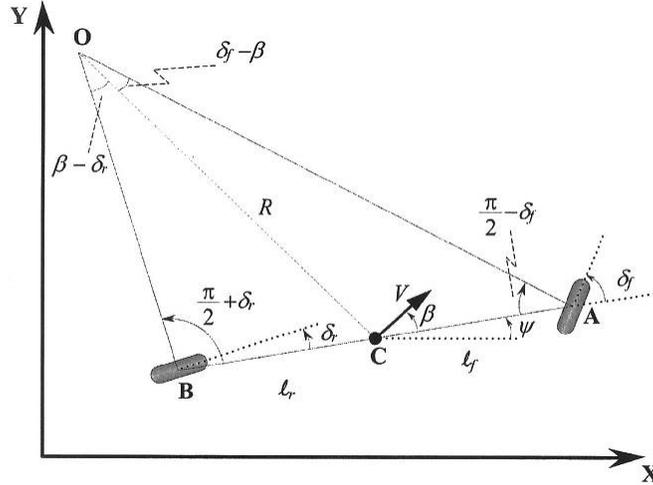


Figure E - 1 Cinématique latérale d'un véhicule, modèle de la bicyclette, tiré de Wang et Qi (2001)

On dénote  $\delta_f$  et  $\delta_r$  l'angle de direction des roues avant et arrière respectivement,  $V$  la vitesse du véhicule,  $\beta$  l'angle du déplacement du véhicule, aussi appelé l'angle de dérive,  $l_f$  et  $l_r$  les distances entre l'avant et l'arrière du véhicule jusqu'au centre de masse,  $R$  le rayon formé par la distance entre le centre de masse du véhicule et son centre de rotation instantané  $O$  et  $\psi$  le lacet du véhicule par rapport à un référentiel fixe  $XY$ . Le modèle a pour hypothèse que la dérive des roues avant et arrière est négligeable pour des vitesses inférieures à 5 m/s (18 km/h). En effet, la force latérale exercée sur chaque roue lors d'un virage équivaut à

$$F_w = \frac{mV^2}{R} \quad (2.1)$$

où  $m$  correspond à la masse du véhicule en kg. Ainsi, pour de petites vitesses et une grande masse, le terme de vitesse n'influencera que très peu le comportement des roues. En se basant sur des règles de trigonométries générales comme la loi du sinus, il est possible d'obtenir la série d'équations suivantes représentant le modèle :

$$\dot{X} = V \cos(\psi + \beta) \quad (2.2)$$

$$\dot{Y} = V \sin(\psi + \beta) \quad (2.3)$$

$$\dot{\psi} = \frac{V \cos(\beta)}{l_r + l_f} (\tan(\delta_f) - \tan(\delta_r)) \quad (2.4)$$

De plus, l'angle de dérive peut être représenté comme une fonction de la géométrie du véhicule et des valeurs d'entrée d'angle des roues. On trouve que :

$$\beta = \tan^{-1} \left( \frac{l_f \cdot \tan \delta_r + l_r \cdot \tan \delta_f}{l_f + l_r} \right) \quad (2.5)$$

Cependant, comme la masse du robot développé (2.5 kg) est largement inférieure à celle d'une automobile et que les vitesses désirées sont bien supérieures à 5 m/s, le modèle de la bicyclette est difficilement implantable dans le cadre de ce projet. Par ailleurs, des essais empiriques ont permis de démontrer que les équations du modèle ne représentaient pas de façon fidèle le comportement du véhicule. Il existe également une version dynamique du modèle incluant la réaction aux appuis sur chaque roue se basant sur le comportement dynamique du véhicule. Ce modèle a pour nom le modèle dynamique de la bicyclette et est utilisé dans le développement de véhicules autonomes dont la complexité dépasse largement celle désirée dans le cadre de ce projet (Polack, Altché, d'Andréa-Novel, & de La Fortelle, 2017). Bien que très précis, la lourdeur numérique qu'apporte le modèle dynamique est non négligeable et impose un poids de calcul qui surpasse les capacités de l'ordinateur actuellement sélectionné, affectant de ce fait la fréquence d'exécution des analyses en plus de représenter inadéquatement le comportement cinétique du véhicule à faible masse. Ainsi, un modèle plus simple répondant aux besoins du projet a été développé.

## 2.2 Modèle adapté

Pour modéliser adéquatement la physique régissant le robot, un nouveau modèle a été mis au point. La **Erreur ! Source du renvoi introuvable.** représente la géométrie du véhicule utilisé dans le cadre du projet (toutes les mesures sont en centimètres). On dénote par ailleurs que la géométrie du système de direction ne respecte pas la direction d'Ackermann (Rajamani, 2006), donc que deux centres de rotation instantanés existent, forçant ainsi un glissement de la roue extérieure.

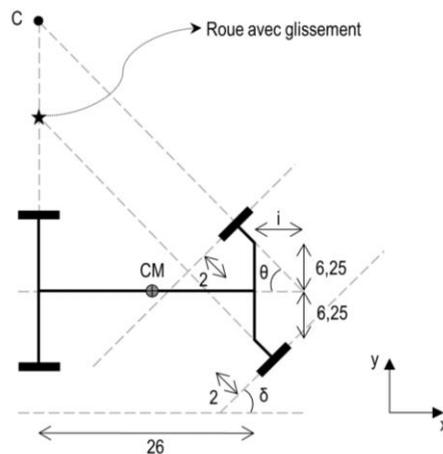


Figure E - 2 Représentation géométrique du modèle du véhicule utilisé pour le projet.

Un ensemble de variables ont été développées afin de décrire le véhicule géométriquement. La **Erreur ! Source du renvoi introuvable.** présente une schématisation du système de direction de la roue intérieure au virage du véhicule:

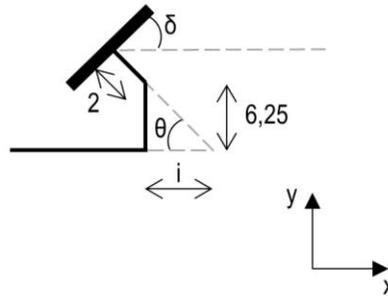


Figure E - 3 Isolation géométrique de la roue responsable de la trajectoire du véhicule.

Comme présenté à la **Erreur ! Source du renvoi introuvable.**, la position du centre de rotation instantané p peut être déterminée comme une fonction de  $l_f$ ,  $l_r$ ,  $i$  et  $\delta$ .

$$i = \frac{\frac{T}{2}}{\tan\left(\frac{\pi}{2} - \delta\right)} \quad (2.6)$$

$$R_1 = (l_r + l_f + i) \tan\left(\frac{\pi}{2} - \delta\right) \quad (2.7)$$

$$R_0 = \sqrt{R_1^2 + l_r^2} \quad (2.8)$$

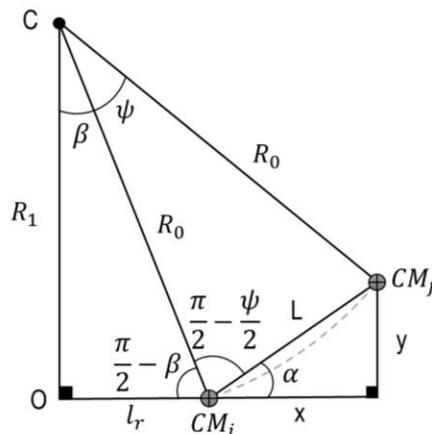


Figure E - 4 Modèle de translation du véhicule par rapport au centre de masse.

On dénote à la **Erreur ! Source du renvoi introuvable.**  $CM_i$  la position initiale et  $CM_f$  la position finale. Le trait pointillé détaille la trajectoire curviligne empruntée par le centre de masse tandis que le trait L représente le déplacement linéaire du centre de masse. Il est à noter que pour un petit rayon  $R_0$  et un angle  $\psi$  faible, la

longueur de l'arc est environ égale à la corde  $L$ . De cette façon, la courbure de la trajectoire peut être négligée. Par ailleurs, la vitesse angulaire du véhicule peut s'exprimer comme le rapport de la vitesse sur le rayon au centre de rotation instantané :

$$\dot{\psi} = \frac{V}{R_0} \quad (2.9)$$

En considérant un intervalle de temps  $dt$  très petit et une vitesse angulaire constante pour un même  $dt$  connue, il est possible d'exprimer le déplacement rectiligne du  $CM$ ,  $L$ , comme suit:

$$\psi = \frac{V}{R_0} dt \quad (2.10)$$

$$L = 2 R_0 \sin \frac{\psi}{2} \quad (2.11)$$

$$\beta = \arctan \frac{l_r}{R_1} \quad (2.12)$$

$$\alpha = \beta + \frac{\psi}{2} \quad (2.13)$$

$$x = L \cos \alpha \quad (2.14)$$

$$y = L \sin \alpha \quad (2.15)$$

De manière à attester la validité du modèle, des essais ont été réalisés sur la piste d'athlétisme de l'Université. Lors de ces tests, le robot a été placé au centre d'un corridor de course selon une position standardisée délimitée à l'aide d'un appui fixe permettant d'appuyer l'arrière du véhicule et du ruban adhésif aidant à la délimitation latérale du véhicule. À partir de la position de départ, un angle de direction des roues est fixé via le programme de contrôle. Un déplacement de 0.4 m (dans le sens des lignes du corridor) par rapport à un référentiel global est par la suite effectué par une seconde commande logicielle. La valeur de 0.4 m correspond approximativement au déplacement maximal pouvant être fait par le robot sur une période correspondant à 3 exécutions de la boucle de contrôle à la vitesse maximale de 6.5 m/s désirée pour ce projet. Comme le robot est mis en arrêt lorsqu'une suite de 3 erreurs survient, la distance utilisée présente le pire cas possible de déviation latérale. Une fois la distance couverte, le véhicule est arrêté physiquement par un obstacle et la position finale et son lacet sont déterminés à l'aide d'un ruban à mesurer. Les lignes de la piste sont utilisées à titre de référence pour la prise de mesures. La Figure E - 5 propose une représentation de l'expérience réalisée selon une vue de haut. Le **Erreur ! Source du renvoi introuvable.** présente la moyenne des résultats obtenus pour trois angles respectifs. Un total de 3 essais a été réalisé pour chaque angle initial.

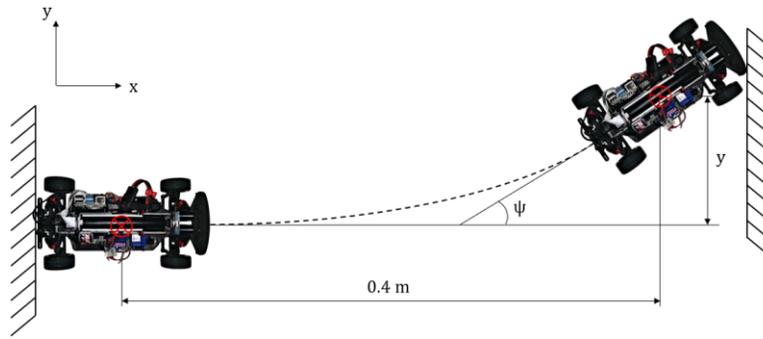


Figure E - 5 Représentation géométrique des essais de validation réalisés

Tableau E - 1 Essai de validation du modèle cinématique du véhicule.

$\delta$ réel	x réel	y estimé	$\psi$ estimé	y réel	$\Psi$ réel	%E(y)	%E( $\psi$ )
10	0.4	9.5	11	$8.83 \pm 0.11$	$12.25 \pm 0.18$	7.58	10.2
20	0.4	18.6	30	$19.38 \pm 0.10$	$32.70 \pm 0.73$	4.02	8.25
25	0.4	29	47	$27.20 \pm 0.17$	$44.80 \pm 0.30$	6.62	4.91

### 2.2.1 Utilisation du modèle

La démarche initiale de ce projet prévoyait l'utilisation d'un algorithme de contrôle prévisionnel appelé Model Predictive Control (MPC) dont la dépendance à un modèle physique est d'une haute importance pour réaliser une conduite adéquate. Bien qu'un contrôle PID ait finalement été sélectionné, tel que présenté à la section Chapitre 3, et que ce système soit indépendant du modèle physique, un simulateur permettant l'ajustement des paramètres de contrôles a su mettre à bon usage le modèle. La Figure E - 6 présente l'affichage de sortie du simulateur avec les paramètres de contrôles finaux du véhicule. La simulation est effectuée sur une piste numérique dont les dimensions correspondent à celle de la piste d'athlétisme de l'Université du Québec à Chicoutimi. La trajectoire est évaluée à l'aide d'un polynôme du deuxième degré passant par les points de la ligne centrale du couloir d'analyse, en l'occurrence pour l'exemple présenté ici, celui le plus à l'extérieur (corridor 4). On remarque une déviation un peu plus importante de la trajectoire en sortie de courbe similairement à ce qui est observé lors des essais réels. La trajectoire simulée, en rouge, est estimée à partir de la projection de la ligne médiane de la voie de déplacement présentée avec un trait pointillé. Le déplacement est effectué dans le sens antihoraire conformément aux pratiques usuelles en athlétisme. Le système de coordonnées globales de l'affichage a pour origine le centre de l'arc de cercle formant la courbe de la piste (voir trait noir en croix en position  $\{0, 0\}$ ).

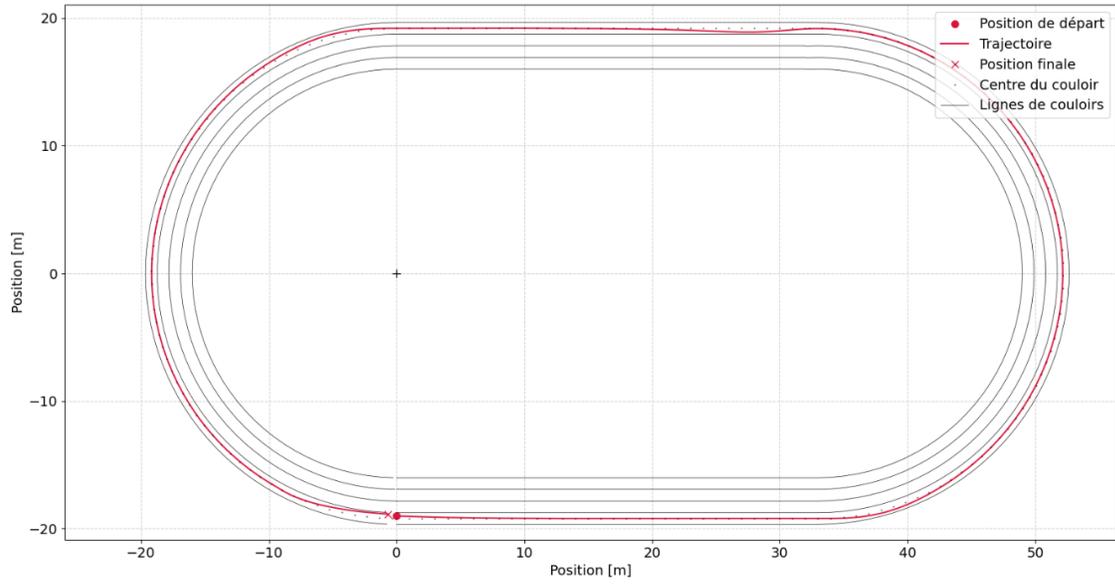


Figure E - 6 Résultats graphiques de la simulation de conduite autonome.