Course Voitures Autonomes Paris Saclay (CoVAPSy) : Travaux pratiques autour des voitures autonomes

Culture Sciences de l'Ingénieur

La Revue 3E.I

école normale supérieure paris—saclay

Thomas BOULANGER[1] – Eve DÉLÈGUE[2] – Kévin HOARAU[3]
Anthony JUTON[4]

Édité le 06/11/2023

[1] *Élève en année de recherche pré-doctorale à l'étranger, ENS Paris-Saclay – DER Nikola Tesla*
[2] *Élève en année de recherche en intelligence artificielle, ENS Paris-Saclay – DER Nikola Tesla*
[3] *Élève en M2 Formation à l'Enseignement Supérieur, ENS Paris-Saclay – DER Nikola Tesla*
[4] *Professeur agrégé de physique appliquée au DER Nikola Tesla, ENS Paris-Saclay*

Cette ressource fait partie du N°111 de La Revue 3EI de janvier 2024.

Portés par l'essor de la recherche sur les véhicules autonomes, dans le but de travailler l'informatique embarquée et/ou l'intelligence artificielle, les établissements de l'Université Paris Saclay organisent depuis 2019 une course de voitures autonomes 1/10ème, ouverte à tous (voir la vidéo « *Course de voitures autonomes 2023* » [1]). Au fil de ces quatre années, les solutions ont mûries et sont devenues fiables, ce qui nous permet aujourd'hui de diffuser ces travaux.

Figure 1 : Voiture sur le simulateur webots

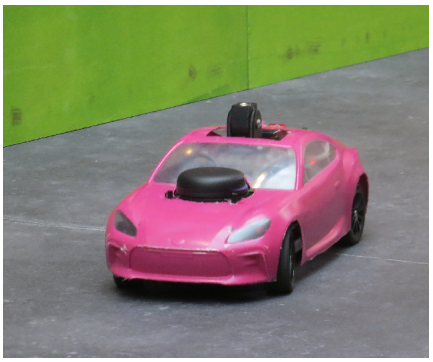

Figure 2 : Voiture réelle

Cette ressource présente et introduit deux supports de travaux pratiques et/ou projets, qui peuvent être travaillés indépendamment ou de façon complémentaire, en NSI, en SI, en BTS CIEL, en BUT GEII ou en école d'ingénieur :

- Une simulation sur le logiciel Webots où les étudiants travaillent sur le code de conduite autonome d'une voiture, en python ou en C, sur une piste virtuelle afin de parcourir la piste le plus rapidement possible [2]. Il est possible de faire concourir plusieurs voitures ensemble.
- Une partie expérimentale sur une voiture 1/10ème, mettant en œuvre l'acquisition des données du lidar, les commandes des moteurs et l'utilisation d'un nano-ordinateur raspberry pi [3] (python) ou d'un microcontrôler STM32 (langage C) [5]. L'algorithme de conduite autonome peut être travaillé directement sur la voiture réelle ou repris du code du simulateur compatible.
- Le travail à la fois sur simulateur et sur la voiture réelle permet d'étudier les limites de la simulation, l'amélioration du passage simulation → réalité...

Pour aller plus loin ou pour participer à la course de Paris Saclay, des ressources sont également disponibles sur le site de la course de voitures autonomes du plateau de Saclay [4] : https://ajuton-ens.github.io/CourseVoituresAutonomesSaclay/

La vidéo de l'édition 2023 est disponible sur Culture Sciences de l'Ingénieur [1].

Figure 3 : Grille de départ de la course de voitures autonomes de Paris Saclay 2023, [1]

Les pré-requis sont l'utilisation de python ou du C au choix (utilisation de fonctions, utilisation des tableaux) pour la simulation comme pour la partie expérimentale.

1 - Introduction

Depuis une vingtaine d'années, l'industrie automobile et les organismes de recherche ont pour objectif de faire rouler sur route ouverte des voitures autonomes, sans conducteur. Cette technologie offrirait la possibilité de voyager de manière plus sûre, en utilisant le temps du voyage pour d'autres activités de loisirs ou professionnelles.

Des voitures autonomes (Waymo, Renault notamment) sont déjà en phase de test dans le monde, mais leur adoption par le public soulève des questions importantes concernant notamment la sécurité des usagers (les Tesla ont notamment eu de graves problèmes de « freinage fantôme ») et la responsabilité en cas d'accident.

Par ailleurs, des voitures à l'autonomie partielle sont déjà commercialisées : parking automatique, maintien dans la voie sur autoroute sont des fonctionnalités courantes sur les voitures haut de gamme.

Figure 4 : Renault Zoe Cab, en phase de test (laboratoire Paris-Saclay Autonomous Lab), source Renault Group

Afin de pouvoir se déplacer sans conducteur, les voitures autonomes font appel à de nombreuses technologies intéressantes pour l'enseignement des sciences de l'ingénieur et de l'informatique. On peut les regrouper autour de trois fonctions principales :

- observer l'environnement autour de la voiture et s'y repérer à l'aide de capteurs (GPS, lidar, caméra, radar, télémètres ultrason, centrale inertielle, ...),
- décider de la direction et de la vitesse à l'aide d'un contrôleur embarqué,
- agir sur la direction et la propulsion (direction motorisée et moteur de la voiture).

Figure 5 : Robot Taxi Waymo, actuellement en circulation en phase de test à Phoenix, Arizona et Los Angeles, Californie, Source Waymo

2 - Présentation de la voiture

La voiture utilisée pour la course de voitures autonomes de Paris Saclay (CoVAPSy) est au format 1/10ème, ce qui permet un coût raisonnable (1000 à 1500 euros) et un risque nul.

Le châssis TT-02 choisi est robuste et bon marché. Il est possible de trouver des pièces détachées chez tous les fournisseurs de modélisme. La voiture complète permet l'usage de microcontrôleur (Arduino ou STM32) et/ou d'un nano-ordinateur (raspberry Pi) et/ou d'un GPU (Jetson Nano), de lidar et/ou caméra, de télémètres et d'asservissement de vitesse.

Les schémas des cartes électroniques, les plans des pièces mécaniques et les références des composants sont disponibles sur le dépôt git de la course [4].

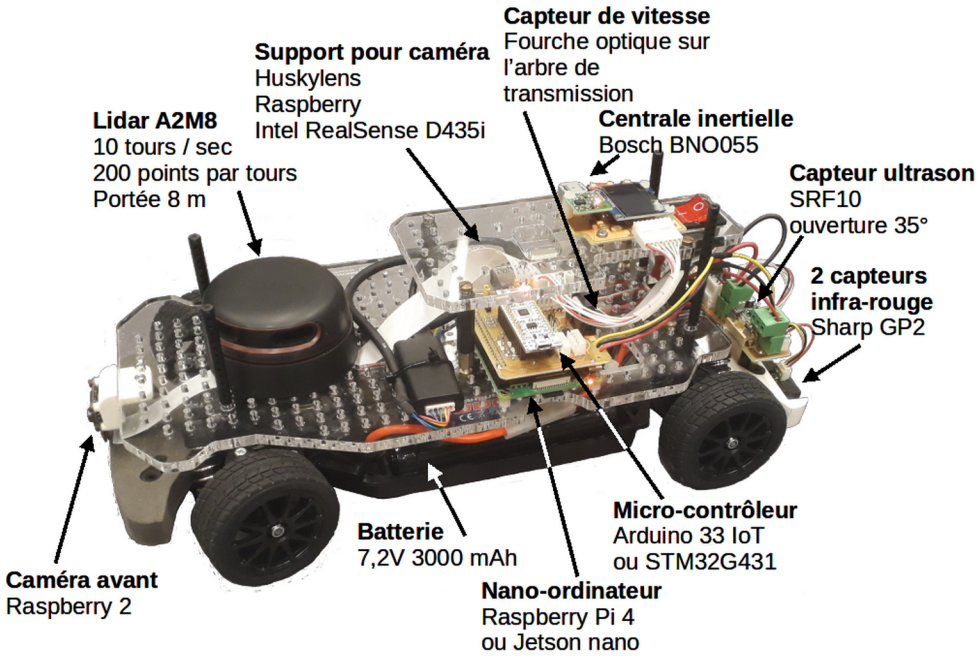

Figure 6 : Voiture autonome CoVAPSy 2023

Le diagramme SysML de blocs internes de la voiture CoVAPSy complète, sans caméra, est le suivant :

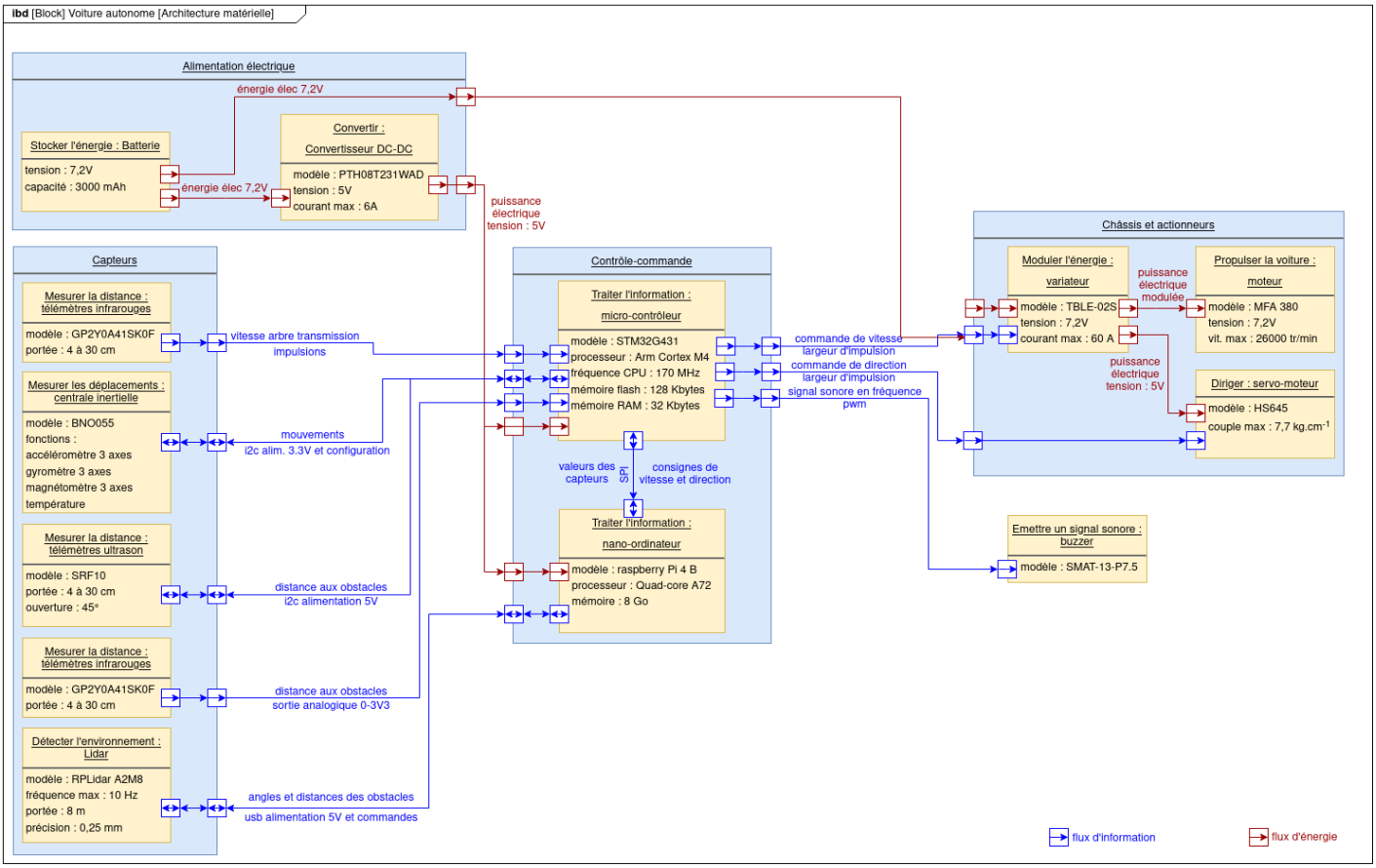

Figure 7 : Diagramme SysML de blocs internes de la voitures CoVAPSy 2023

Pour la première approche proposée par cette série de ressources (simulation et mise en œuvre expérimentale), une version simplifiée *CoVAPSy_RPIonly* est présentée, avec seulement un nano-ordinateur raspberry Pi et un lidar, pour un coût d'environ 650 euros.

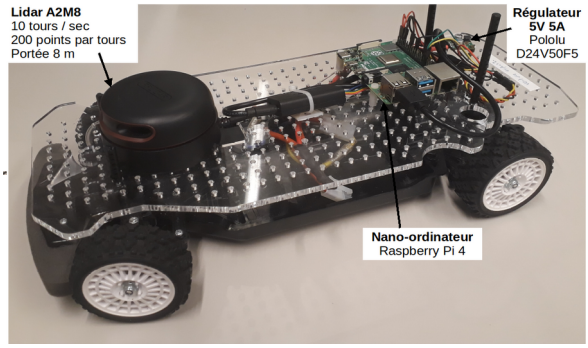

Figure 8 : Voiture autonome CoVAPSy RPIonly

Figure 9 : Voiture autonome CoVAPSy_RPIonly avec sa carrosserie

La liste des références des composants et de fournisseurs potentiels est fournie en annexe (§6).

Une version similaire de cette voiture simplifiée, contrôlée par un microcontrôleur STM32 programmé en langage C, est présentée dans la ressource « *CoVAPSy : Premiers programmes en langage C sur voiture réelle* » [5].

La même voiture a été conçue sur le simulateur Webots, avec des dimensions et des caractéristiques dynamiques les plus proches possible du châssis TT-02 utilisé. La carrosserie, issue d'un modèle de Chevrolet Camaro dessiné spécialement pour être simple à simuler, est un peu différente :

Figure 10 : La voiture CoVAPSy sur le simulateur

Les différents éléments de la voiture simplifiée sont présentés sur le diagramme de blocs internes suivant :

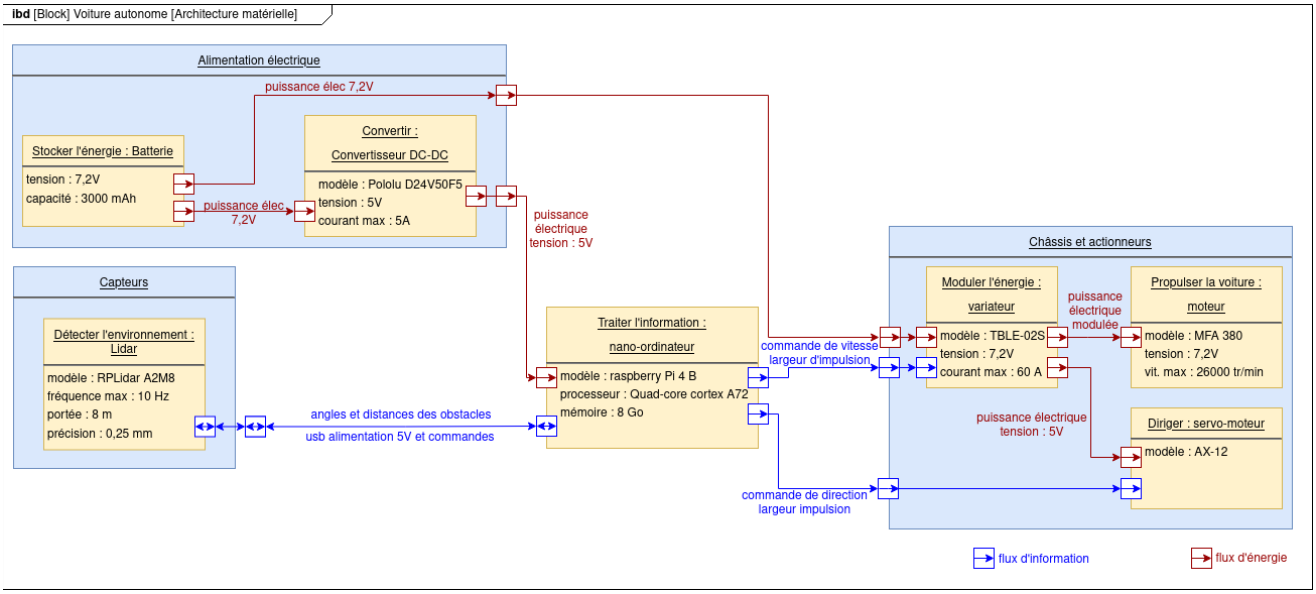

Figure 11 : Diagramme SysML de blocs internes de la voiture CoVAPSY_RPIonly

Un Lidar (*laser imaging detection and ranging* soit en français *détection et estimation de la distance par laser)* est un télémètre tournant. Une impulsion laser est émise par le Lidar, elle se réfléchit sur les objets qu'elle rencontre et revient sur un photorécepteur du Lidar. Le lidar A2M8 (ou son successeur A2M12) est un lidar bon marché utilisant la triangulation pour estimer la distance à l'objet ayant réfléchi l'impulsion laser.

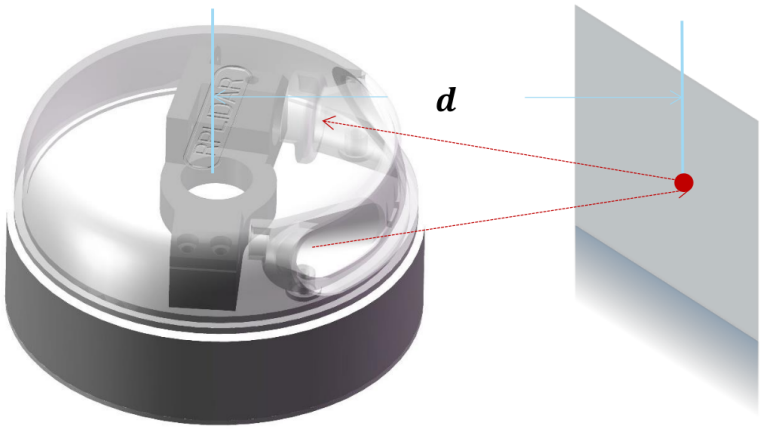

Figure 12 : Principe de la mesure de distance du Lidar Slamtec A2M8, source Slamtec

Les lidars plus précis (Slamtec S2 par exemple ou lidar professionnels Hokuyo) mesurent le temps de l'aller-retour (ToF Time of Flight) de l'impulsion laser pour estimer la distance à l'objet.

Le détecteur tourne sur lui-même afin de connaître la distance de l'ensemble des objets entourant la voiture. Les données du lidar sont fournies, sur le simulateur comme avec la voiture réelle, dans un tableau de 360 cases, chaque case correspondant à la mesure à 1 degré, l'angle 0 étant devant la voiture.

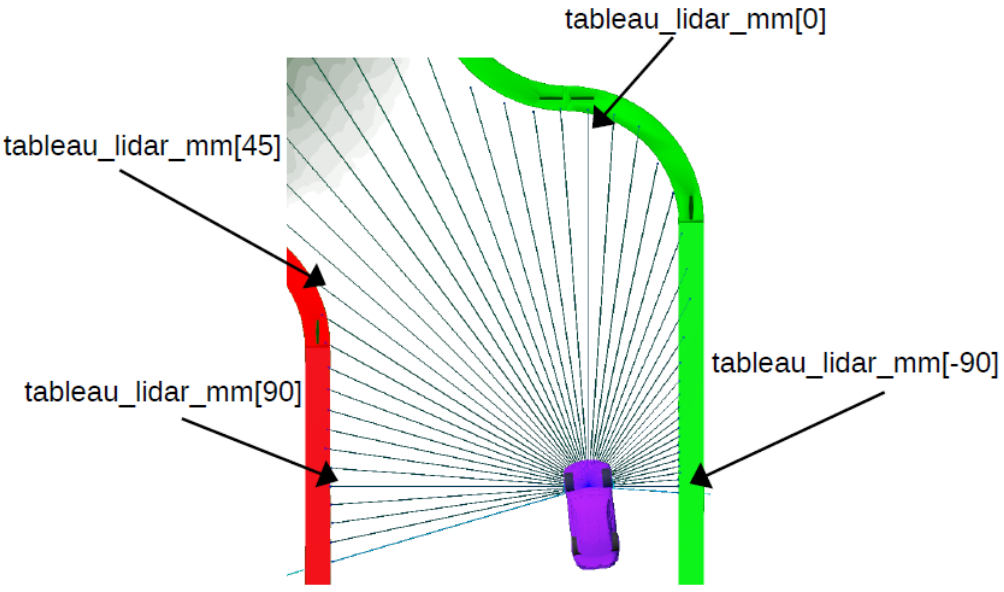

Figure 13 : Affichage sur le simulateur d'un rayon de mesure sur 4 du lidar

Deux moteurs sont utilisés sur la voiture, issus du modèle de voiture radiocommandée. Le premier sert à la propulsion de la voiture et est alimenté par un variateur de vitesse. Le second contrôle la direction de la voiture. C'est un servomoteur, un moteur asservi en position. Les deux reçoivent du nano-ordinateur ou du microcontrôleur une commande sous forme d'impulsion :

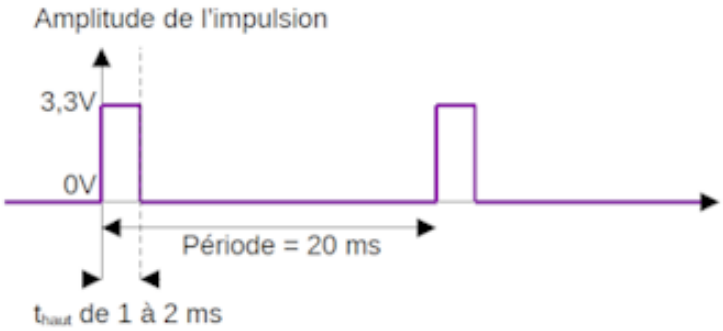

Figure 14 : Impulsion de commande du servomoteur et du variateur de propulsion

Suivant les marques de servomoteur et de variateur (Tamiya ne met pas toujours les mêmes variateurs dans ses kits de voitures 1/10ème et le servomoteur est acheté à part), les sens de rotation et les butées changent de valeurs (toujours entre 1 et 2 ms). Il est donc important de prendre le temps de calibrer ces butées, comme c'est expliqué dans la ressource « *CoVaPSy : Premiers programmes python sur la voiture réelle* » [2].

Une batterie NiMH 7,2V 3000 mAh fournit l'énergie à tous les éléments de la voiture (lidar, nano-ordinateurs, moteurs, …). Un régulateur Pololu D24V50F5 génère une tension 5V à partir de la tension 7,2V de la batterie pour l'alimentation du nano-ordinateur. Le variateur du moteur de propulsion fournit quant à lui la tension 5V d'alimentation du servomoteur de direction.

Un nano-ordinateur a la charge de piloter la voiture. La carte raspberry Pi, très populaire, regroupe une large communauté d'utilisateurs partageant leurs expériences, des guides de mise en œuvre, des forums de résolution de problème. Elle utilise une distribution linux pour laquelle existe une bibliothèque (python ou c++) pour l'utilisation du Lidar et une (également python ou c++) pour la génération d'impulsions en vue de commander les moteurs.

3 - Course de voitures simulées

La simulation utilise le logiciel Webots. La programmation peut être faite en python ou en C.

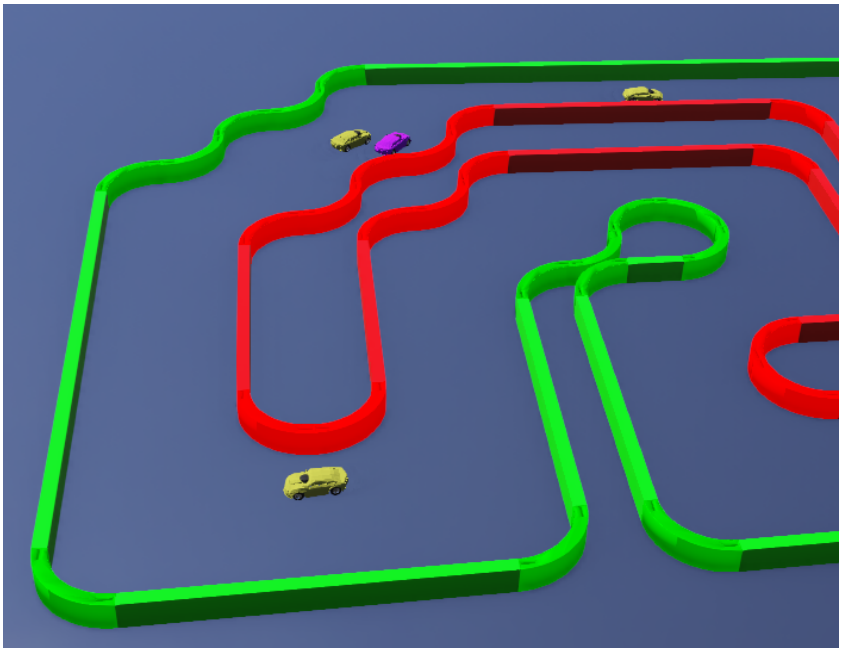

Figure 15 : Course de voitures autonomes sur le simulateur, entre une voiture étudiant (violette) et 3 voitures préprogrammées (« sparring partners ») en jaune

Le simulateur fourni propose une piste et un modèle de voiture proche de la voiture réelle (blocs jaunes de la figure 16). Les fonctions de traitement des données du lidar, de contrôle de la vitesse et de la direction sont données (en violet sur la figure 16), ainsi qu'un algorithme de conduite basique (en rouge sur la figure 16), en C sur l'une des voitures et en python sur une autre.

Le travail des élèves / étudiants est dans un premier temps de faire un programme de conduite plus performant que celui des voitures préprogrammées (sparring partners).

Il est ensuite possible de positionner plusieurs voitures sur la piste pour faire une course entre les voitures programmées par différents étudiants. Les pistes de travail sont nombreuses : Côté programmation, on peut travailler sur les machines à état pour le contrôle de la voiture ou ajouter un nœud superviseur pour mesurer le temps au tour des voitures. Côté asservissement, on peut travailler sur les algorithmes d'asservissement classiques (PID) ou sur des méthodes avancées.

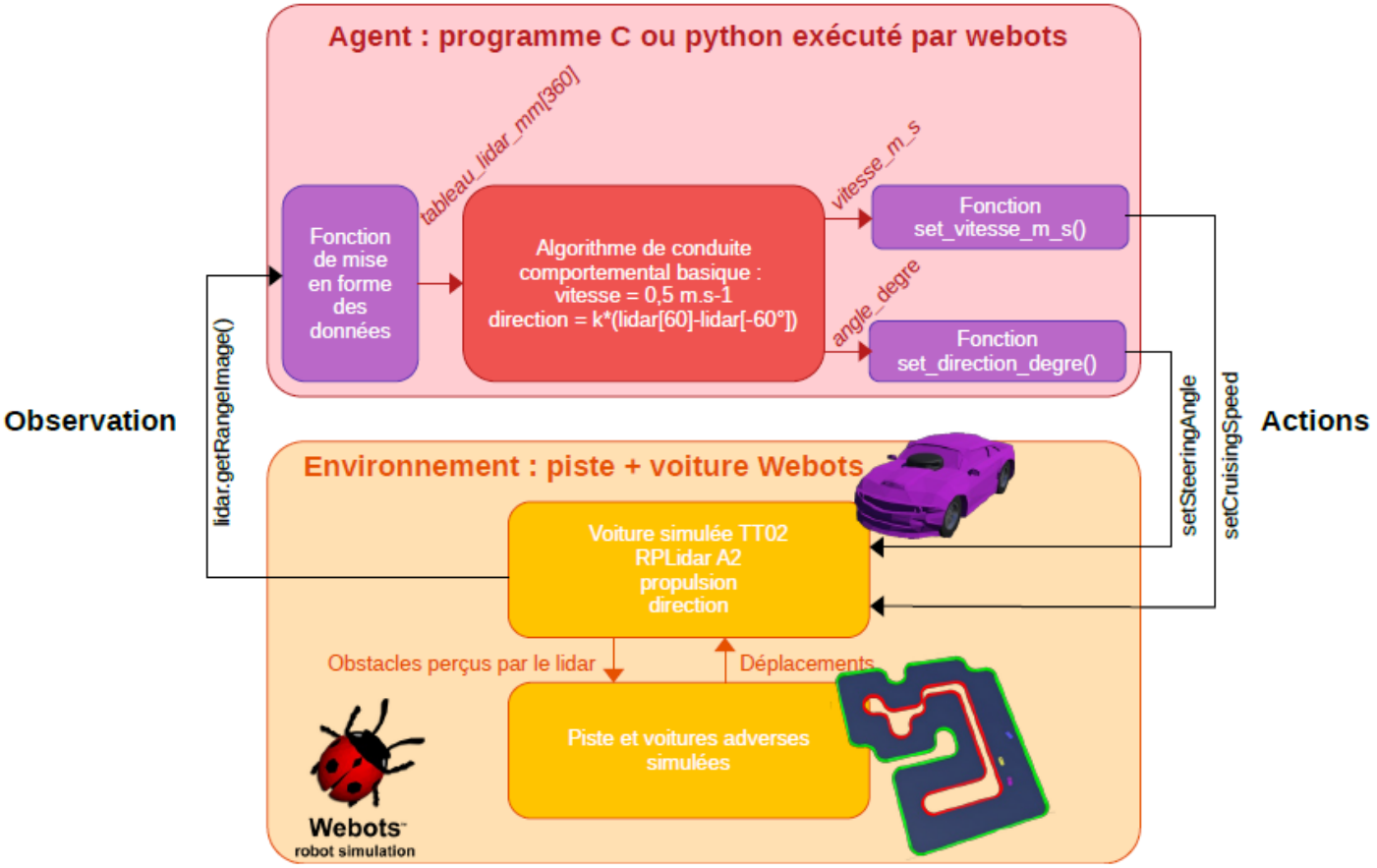

Figure 16 : Schéma des blocs logiciels fournis pour le travail sur le simulateur

Le ressource « *CoVapSy : Mise en œuvre du simulateur Webots* » [2] décrit en détail la prise en main du simulateur et les fonctions fournies, jusqu'au premier programme de conduite basique.

4 - Course de voitures réelles

La ressource « *CoVAPSy : Premiers programmes python sur la voiture réelle* » [3] traite de la mise en œuvre de la voiture CoVAPSy_RPIonly, programmable en python. La ressource « *CoVAPSy : Premiers programmes en langage C sur voiture réelle* » [5] propose la mise en œuvre avec la voiture CoVAPSy_STM32, programmée en C.

La voiture simulée ayant été conçue pour être la plus proche possible de la voiture réelle, des fonctions de traitement des données du lidar, de contrôle de la vitesse et de la direction (en rouge sur la figure 17) ont été développées avec un comportement proche de celles du simulateur. Le même algorithme de conduite basique (en rouge) peut donc être utilisé.

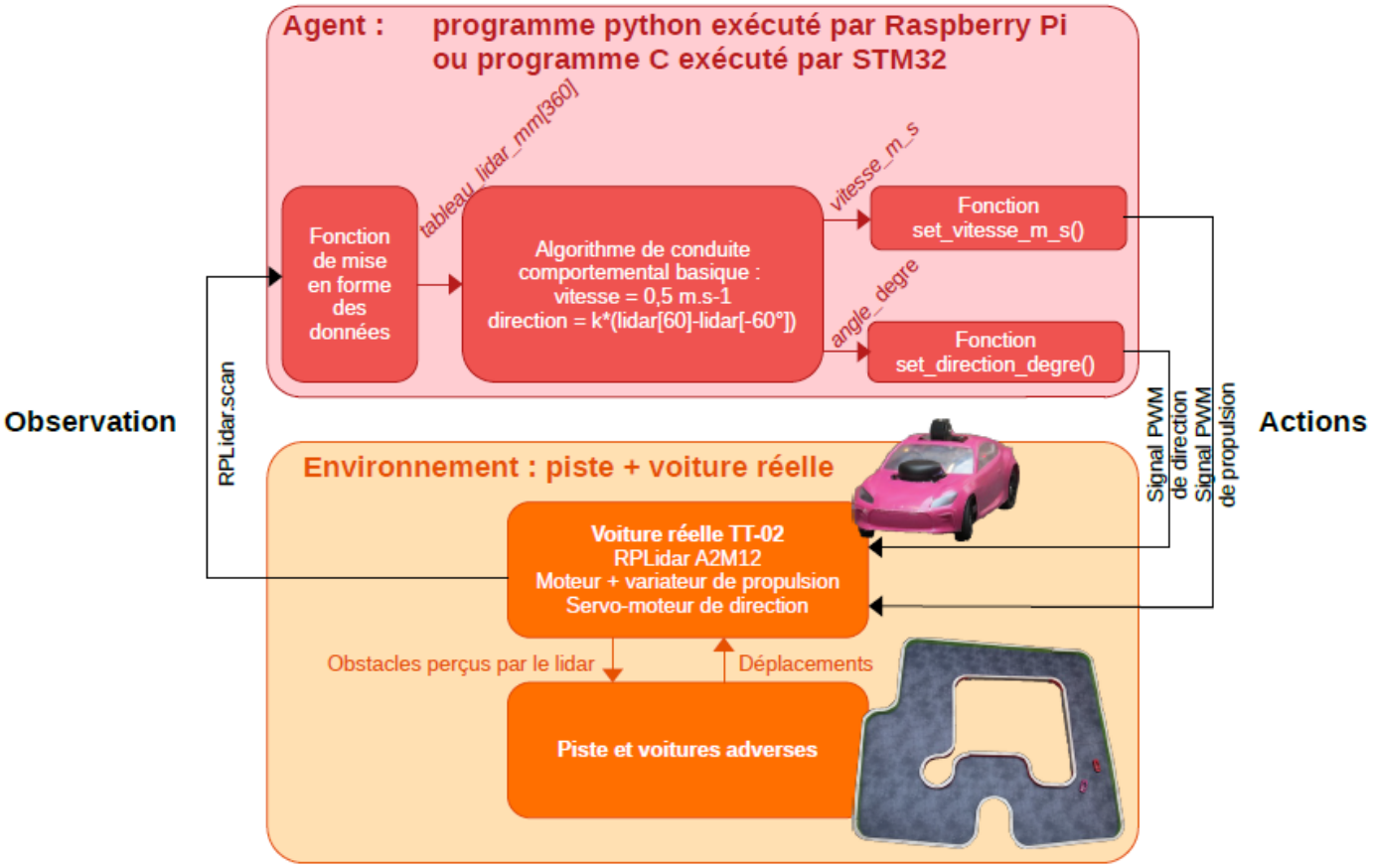

Figure 17 : Schéma des blocs logiciels fournis pour le travail sur la voiture réelle

5 - Conclusion

Ces deux activités permettent la simulation du fonctionnement d'une voiture autonome sur un circuit virtuel et la mise en pratique sur une piste réelle. Ces activités menées en parallèle permettent de mettre en évidence l'écart entre le domaine simulé et le domaine réel et de travailler à l'amélioration du modèle (prise en compte de la dynamique de la direction), à l'amélioration de la voiture réelle (asservissement de vitesse), pour rendre plus robuste le passage simulation → réalité.

Cette ressource s'appuie sur l'utilisation d'un seul Lidar pour la perception de l'environnement, bien adapté à une piste avec des bordures. Les voitures autonomes réelles circulent sur des routes sans bordure et utilisent notamment des caméras pour repérer le marquage au sol. L'utilisation de caméras implique des unités de calcul plus complexes et des algorithmes de traitement d'image avancés. L'ajout d'une caméra sur la voiture autonome 1/10ème permet d'approcher ces problématiques, mais au prix d'une complexité qui dépasse le public bac/bac+2 ciblé par ces ressources.

6 - Annexe

Désignation	Fabricant	Référence fabricant	Fournisseur	Réf fournisseur	Prix TTC	Remarque
Tamiya TT-02 Toyota GR 86 KIT 58694	Tamiya	58694	RCTeam	58694	134,90 €	disponible aussi chez Conrad
Orion Chargeur IQ801 1A ORI30197	Orion	IQ801	RCTeam	ORI30197	15,90 €	disponible aussi chez Conrad
Accu 7.2v Nimh 3000mah Tamiya T1006300	Tamiya	T1006300	RCTeam	T1006300	27,30 €	disponible aussi chez Conrad
Servomoteur Konect 9kg 0.13s KN-0913LVMG	Konect	KN-0913LVMG	RCTeam	KN-0913LVMG	19,90 €	disponible aussi chez Conrad
Scanner Laser 360° RPLIDAR A2M12	Slamtec	RPLIDAR A2M12	Robotshop	RB-Rpk-22	263,90 €	disponible aussi chez Gotronic
Convertisseur DC-DC Pololu D24V50F5	Pololu	D24V50F5	Robotshop	RB-Pol-295	56,41 €	disponible aussi chez Gotronic
Raspberry Pi 4 Modèle B, 8Go	Raspberry	PI4-8GB	Kubii	PI48GB	95,40 €	appeler en tant qu'établissement
Carte Micro-SD SanDisk Classe 10 32GB	SanDisk	SDXC CI10 UHS-1	Kubii	SD_SANDISK	11,95 €	d'enseignement si rupture de stock.
				TOTAL TTC	625,66 €	

Tableau 1 : Références et fournisseurs pour la voiture CoVAPSy_RPIonly

Références :

[1]: Course de voitures autonomes 2023, mai 2023, https://eduscol.education.fr/sti/si-ens-paris-saclay/actualites/course-voitures-autonomes-2023-resultats

[2]: CoVAPSy : Mise en œuvre du Simulateur Webots, T. Boulanger, E. Délègue, K. Hoarau, A. Juton, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/covapsy-mise-en-oeuvre-du-simulateur-webots

[3]: CoVAPSy : Premiers programmes python sur la voiture réelle, T. Boulanger, E. Délègue, K. Hoarau, A. Juton, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/covapsy-premiers-programmes-python-sur-voiture-reelle

[4]: Dépôt git de la course de voitures autonomes de Paris Saclay : https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay

[5]: CoVAPSy : Premiers programmes en langage C sur voiture réelle , A. Azan, A. Juton, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/covapsy-premiers-programmes-langage-c-sur-la-voiture-reelle