

Projet Recherche en Ingénierie

Année 2023-2024 / Semestre 9

PRI-2023S9-SA302

Plateforme Véhicule Electrique

Fadi AAD (4025940)

Sebastián ZAGAL ROJAS (4027118)

6 février 2024

Site : Villeurbanne

Structure d'Accueil : SA ISP – Ingénierie des Systèmes Pilotés

Tuteurs scientifiques : Romain DELPOUX, Sébastien MORTEROLLE, Paolo MASSIONI

Organisme comandataire : Laboratoire Ampère – LaMCoS

Contact(s) de l'organisme : Romain DELPOUX, Sébastien MORTEROLLE

Remerciements

L'ensemble de l'équipe tenait à remercier l'ensemble des personnes ayant participé de près ou de loin à la réalisation de ce projet.

Nous tenons à exprimer d'abord notre plus grande gratitude à nos tuteurs DELPOUX Romain, MORTEROLLE Sébastien, MASSIONI Paolo pour nous avoir confié le projet de plateforme véhicule électrique et de leur assistance tout au long de notre travail.

Nous voulons également remercier chaleureusement monsieur POULIQUEN Ludovic pour ses précieux conseils et astuces qui nous ont dépanné plusieurs fois, et monsieur CANO Pierre pour nous avoir fourni l'aide matérielle requise pour l'avancement de notre projet.

Votre expertise et votre disponibilité ont été d'une grande valeur pour notre réussite. Nous espérons continuer à travailler ensemble à l'avenir.

Résumé

Ce projet consiste à programmer et téléguider un véhicule électrique à 4 roues motorisées. La conception du véhicule étant déjà étudiée, nous avons donc pour but d'écrire un programme pour contrôler le véhicule à partir d'une télécommande. Pour l'atteindre nous avons recours à une étude dimensionnelle, plusieurs programmations sur Matlab, et à une modification structurelle du véhicule.

Abstract

This project is about programming and wirelessly commanding a four-wheel drive electric vehicle. The design of the vehicle has already been studied, so our goal is to write a program to control the vehicle using a remote control. To achieve this, we are starting with a dimensional study, multiple Matlab programs, and a structural modification of the vehicle.

Table des Matières

Chapitre 1. Introduction.....	6
Chapitre 2. Dynamique du véhicule	7
2. 1. Caractéristiques du véhicule.....	7
2. 2. Puissance du véhicule.....	8
2. 3. Répartition de masse.....	8
2. 4. Forces agissant sur le véhicule	8
2. 5. Rayon de braquage minimal.....	9
2. 6. Angle de braquage.....	10
2. 7. Effet d'un différentiel	12
Chapitre 3. Modification de la plateforme.....	14
3. 1. Changement des cartes	14
3. 2. Changement de l'architecture du véhicule	15
3. 1. Replanification de l'alimentation électrique.....	16
Chapitre 4. Programmation de la commande	20
4. 1. Schéma entrée/sortie de la dSpace.....	20
4. 2. Mise en place d'une modélisation 3D	22
4. 3. Modèle final retenu.....	26
4. 3. a. Quelques modifications	26
4. 3. b. Composition de notre modèle de commande Simulink	26
Chapitre 5. Problèmes rencontrés	31
Chapitre 6. Conclusion et perspectives.....	32
Chapitre 7. Bibliographie.....	33

Table des Figures

Figure 1 - Schéma des différentes grandeurs de notre véhicule.....	7
Figure 2 - Paramétrage des roues avant.....	10
Figure 3 - Schéma modélisant notre cas avec une roue moyenne.....	11
Figure 4 - Schéma illustrant le parcours des roues arrière du véhicule lors d'un virage.	12
Figure 5 - Nouveau emplacement des composants sur le châssis.	15
Figure 6 - le plan du nouveau câblage de notre véhicule.	17
Figure 7 – Plaque pour fixer l'interrupteur et les accords électriques.....	18
Figure 8 - Schéma des variables mises en jeu par la dSpace.	20
Figure 9 – Modèle Simulink de Mathworks pour observer et piloter un véhicule en 3D.	22
Figure 10 – Schéma de modification de l'input avec les sous-systèmes SteeringProcessing et SpeedProcessing.....	23
Figure 11 – Assignation des valeurs de la mécanique interne des amortisseurs.....	24
Figure 12 – Modèle 3D final du véhicule.....	24
Figure 13 – Véhicule en 3D en train d'être contrôlé par l'utilisateur du Simulink.....	25
Figure 14 - Modèle Simulink final pour la commande du véhicule.....	26
Figure 15 – Sous-système « Vitesse_CAN ».	27
Figure 16 – Sous-système « Angle_CAN ».	27
Figure 17 - Sous-système "Vitesse_Arrières".	27
Figure 18 – Sous-système « Angle_Braquage ».	28
Figure 19 – Sous-système « Commande_PWM ».	29
Figure 20 – Sous-système « Retard_Signal”.	29

Table des photos

Photo 1 - Carte dsPICDEM MCLV-2.	14
Photo 2 - L'architecture finale de notre véhicule.....	19

Chapitre 1. Introduction

Dans le cadre de notre PRI, l'on nous propose de développer une plateforme de véhicule électrique pour développer des projets de recherche sur la dynamique du véhicule, la commande motrice, la conduite autonome et autres objectifs.

Ce projet fait suite à son lancement en 2018-2019 lors duquel la conception de cette plateforme a été établie. Celui-ci contient 4 moteurs électriques du type BLDC ainsi que 2 servo-moteurs qui assurent le suivi en direction du véhicule. Un ordinateur embarqué, appelé MicroAutoBox II, de la compagnie dSpace, figure sur la plateforme. Il est chargé de faire les calculs que le code embarqué requiert et de communiquer les moteurs entre eux pour contrôler la dynamique du véhicule. De plus nous trouvons des boîtes de commande de type Torque Controller et 4 batteries reliées à chaque moteur.

Au début du projet, nous trouvons un premier fichier de commande qui met en place des lois de commande pour contrôler un véhicule à 4 roues motrices à l'aide du logiciel MATLAB avec la toolbox Simscape. Ce modèle comporte un modèle des roues et du véhicule ainsi qu'une commande de vitesse et virage, accompagnée de divers contrôleurs de vitesse et moteurs équipé d'un différentiel de couple.

Dans ce projet nous avons pour objectif donc de commander notre véhicule en utilisant une télécommande. Nous allons changer les boîtes de commande de type Torque Controller pour des cartes dsPICDEM MCLV-2 de Microchip qui présentent plusieurs avantages. Nous allons aussi établir la dynamique complète du véhicule et l'intégrer dans notre programme de commande sur Matlab Simulink.

Chapitre 2. Dynamique du véhicule

Pour pouvoir commander notre véhicule par une télécommande, nous avons besoin d'établir sa dynamique afin de connaître son comportement sous diverses conditions. Nous nous intéressons principalement à la dynamique du virage et du différentiel.

2. 1. Caractéristiques du véhicule

Notre système se constitue des éléments suivants :

- 4 roues motrices
- 2 servomoteurs pour la rotation des roues avant
- 4 batteries pour alimenter le système

Notre véhicule possède les caractéristiques suivantes :

- Dimensions : 972x626x50 mm
- Masse : supposée de 100 kg
- Empattement : $a_1 + a_2 = 846$ mm
- Voie : $b = 250$ mm
- L'amortisseur est le Kind Shock KS 260 650 LBS. Il a une constante de ressort de 114N/mm et un entraxe de 165 mm

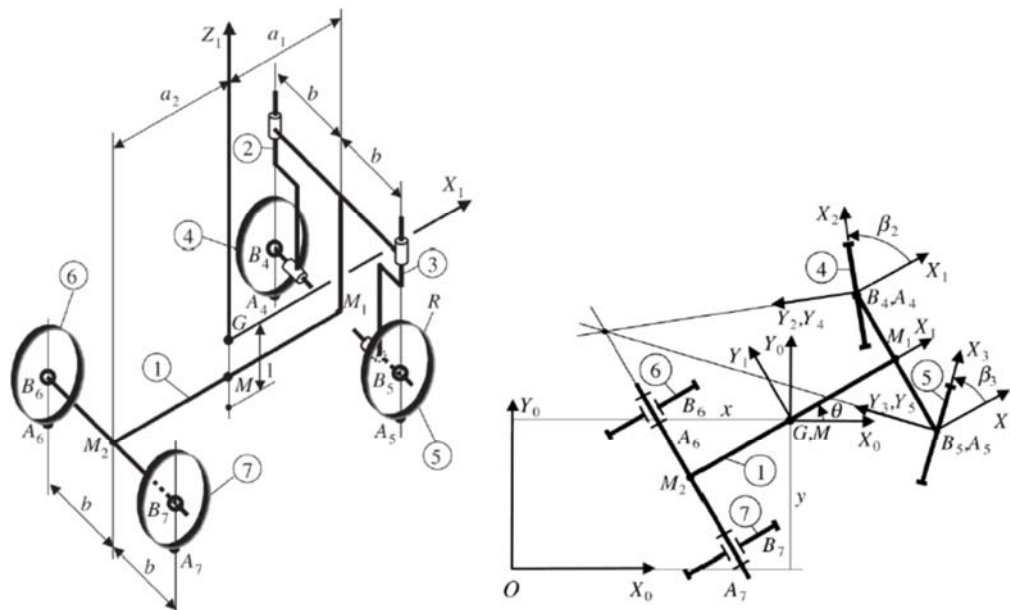


Figure 1 - Schéma des différentes grandeurs de notre véhicule.

2. 2. Puissance du véhicule

Notre véhicule est équipé de 4 moteurs actionnant sur chaque roue. Ils délivrent une puissance maximale de 350 W chacun, donc de 0.48 HP. On a une puissance totale de 1400 W et donc 1.9 HP.

2. 3. Répartition de masse

Dans notre cas, nous considérons que le centre de masse se trouve sur la ligne du milieu par rapport à la direction latérale et parallèle à la direction longitudinale.

Pour déterminer sa position approximative sur cette ligne, nous considérons que la batterie de 12V et la microautobox de dSpace constituent la plus grande masse de notre véhicule et que le centre de masse se trouve entre ces 2 composants. On le considérera au milieu de cette ligne, donc à une distance égale des 2 essieu $a_1 = a_2 = 423 \text{ mm}$.

2. 4. Forces agissant sur le véhicule

Force de résistance à l'avancement :

- Effort de roulement :

$$F_{roulement} = (a + b.V^2).M.g \quad (1)$$

Les coefficients a et b seront déterminés par les caractéristiques de la roue. Nous considérons dans notre cas que le coefficient de frottement est égal à 1. Nous aurons la formule suivante :

$$F_{roulement} = M.g.V^2 \quad (2)$$

- Effort aérodynamique :

$$F_{aero} = \frac{1}{2}\rho SC_x V^2 \quad (3)$$

Puisque la puissance maximale délivrée dans notre cas est de 1.9 HP et qu'il est difficile de calculer le coefficient C_x , on décide de négliger cette force dans notre étude.

- Effort de pente :

$$F_{pente} = M.g.\sin(\alpha) \quad (4)$$

Après avoir fait le calcul de ces forces, nous pouvons déterminer l'équation de la dynamique longitudinale du véhicule :

$$M \cdot \frac{dV}{dt} = F_{moteur} - F_{roult} - F_{aero} - F_{pente} - F_{frein} \quad (5)$$

De même, nous pouvons déterminer les formules suivantes qui agissent sur le véhicule :

$$V = \int \frac{1}{M} (F_{moteur} - F_{roult} - F_{aero} - F_{pente} - F_{frein}) dt \quad (6)$$

$$F_{moteur} = M \cdot \frac{dV}{dt} + F_{roult} + F_{aero} + F_{pente} + F_{frein} \quad (7)$$

2. 5. Rayon de braquage minimal

Un autre aspect de ce véhicule est la possibilité d'assurer un rayon de virage étroit. Les servo-moteurs assurent un angle de rotation de 50° dans chaque direction, donc un angle de 100° au total.

Nous calculons le rayon minimal de rotation par la formule suivante :

$$R_{g \min} = \sqrt{a_2^2 + \left(b + \frac{a_1 + a_2}{\tan(\beta)}\right)^2} \quad (8)$$

Avec :

- $a_1 + a_2 = 846 \text{ mm}$, $a_1 = a_2 = 423 \text{ mm}$
- $b = 250 \text{ mm}$

On trouve alors :

$$R_{g \min} = 1.05 \text{ m} \quad (9)$$

A noter que cette valeur est la valeur théorique. Le rayon de virage minimal effectif dépendra de la vitesse du véhicule ainsi que du coefficient de frottement de la roue μ . Ils sont liés par la relation suivante :

$$R_{g \min} = \frac{V}{\mu \cdot g} \quad (10)$$

2. 6. Angle de braquage

Nous voulons déterminer l'angle de braquage de chaque roue en fonction du rayon de braquage souhaité. Cette relation va dépendre de :

- L'empattement du véhicule $a_1 + a_2 = 846 \text{ mm}$
- La voie entre les roues droite et gauches : $b = 250 \text{ mm}$
- Du rayon de braquage : R

Dans notre cas, le paramétrage des roues avant de notre véhicule est le suivant :

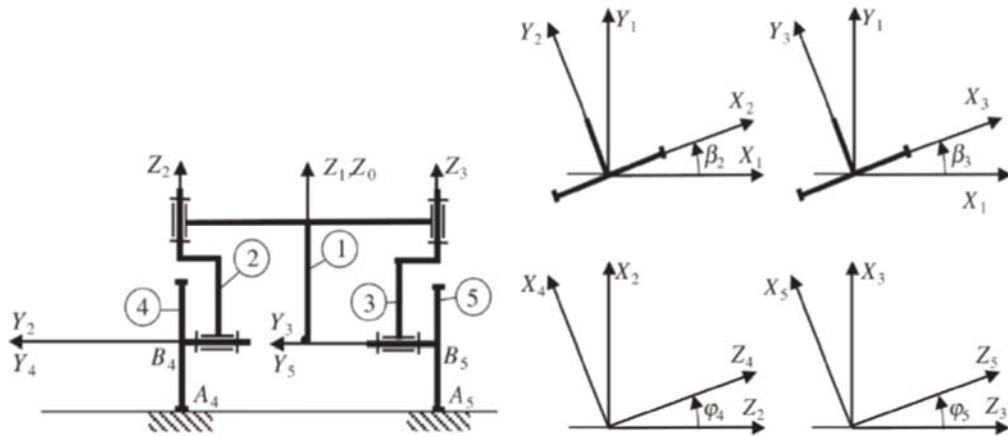


Figure 2 - Paramétrage des roues avant.

Nous écrivons les relations traduisant le roulement sans glissement des roues 4 et 5 :

$$\overrightarrow{V(A_4, 4/0)} = \vec{0} \quad (11)$$

$$\overrightarrow{V(A_5, 5/0)} = \vec{0} \quad (12)$$

Nous obtenons la relation suivante :

$$\cotg(\beta_3) - \cotg(\beta_2) = \frac{2b}{a_1 + a_2} \quad (13)$$

La roue intérieure doit braquer plus que la roue extérieure pour assurer un roulement sans glissement.

Nous considérons l'existence d'une roue moyenne avant pour simplifier nos calculs. Le schéma modélisant notre cas est le suivant :

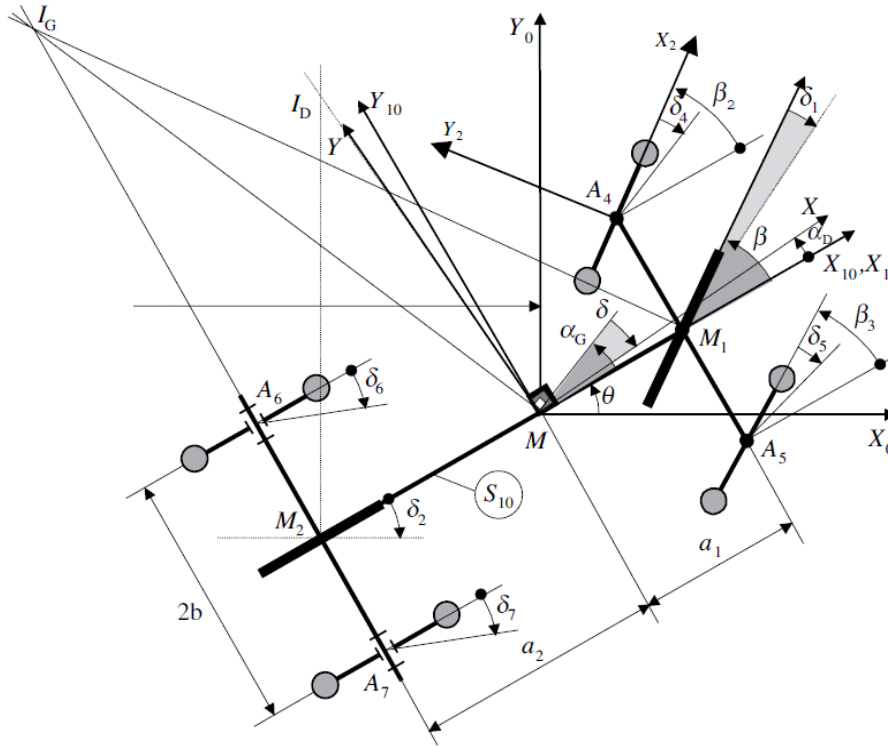


Figure 3 - Schéma modélisant notre cas avec une roue moyenne.

L'angle de braquage de la roue moyenne a donc la formule suivante :

$$\beta = \arctan\left(\frac{a_1 + a_2}{R}\right) \quad (14)$$

De plus :

$$\frac{1}{\tan(\beta)} = \frac{b}{a_1 + a_2} + \frac{1}{\tan(\beta_2)} \quad (15)$$

D'où :

$$\beta_2 = \arctan\left(\frac{\tan(\beta) \cdot (a_1 + a_2)}{a_1 + a_2 + b \cdot \tan(\beta)}\right) \quad (16)$$

Avec β_2 l'angle de braquage de la roue intérieure.

Après factorisation et simplification, nous obtenons la formule suivante :

$$\beta_2 = \arctan\left(\frac{a_1 + a_2}{R - b}\right) \quad (17)$$

Nous savons que β est la moyenne des 2 angles de braquage. Nous pouvons donc trouver l'angle de braquage de la roue extérieure β_3 par la relation suivante :

$$\beta_3 = 2.\beta - \beta_2 \quad (18)$$

$$\Rightarrow \beta_3 = 2.\beta - \arctan\left(\frac{\tan(\beta) \cdot (a_1 + a_2)}{a_1 + a_2 + b \cdot \tan(\beta)}\right) \quad (19)$$

$$\Rightarrow \beta_3 = \arctan\left(\frac{a_1 + a_2}{R + b}\right) \quad (20)$$

Ainsi, selon le sens du virage, la roue droite ou gauche sera affectée à une des valeurs précédentes.

Bien que ces équations soient utiles pour trouver les valeurs des angles lorsqu'on est en train de virer, quand l'on veut rester droit, soit un angle égal à zéro, elles peuvent provoquer une indétermination. Pour éviter ce problème et pour faciliter le calcul des angles dans notre modèle Simulink, considérant les valeurs limites pour notre projet, ce sera mieux de ne pas dépendre du rayon de braquage ni de l'angle de la roue moyenne. Cette hypothèse sera développée dans la partie Simulation 3D (cf. section [4.2.](#)).

2. 7. Effet d'un différentiel

De même, pour l'essieu arrière chaque roue aura une vitesse de rotation différente selon le rayon de braquage. Nous voulons trouver cette relation. Pour cela, nous utilisons le schéma suivant :

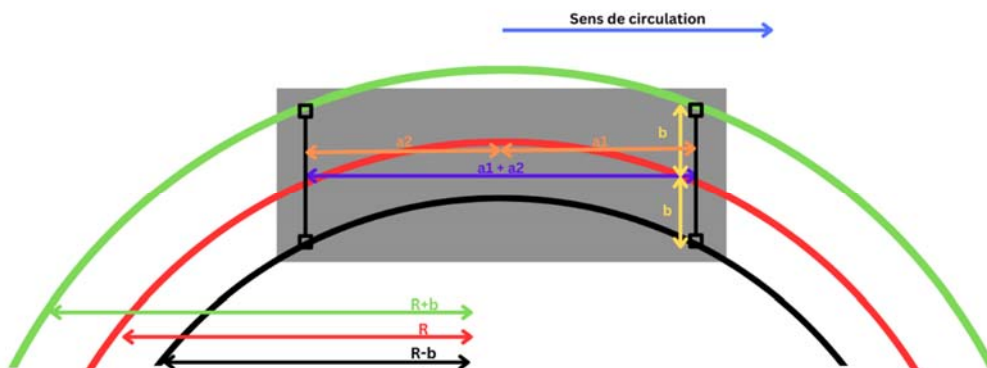


Figure 4 - Schéma illustrant le parcours des roues arrière du véhicule lors d'un virage.

D'après notre schéma, lors d'un braquage de rayon moyen R , nous constatons que chaque roue circule sur un arc de rayon différent :

- La roue externe parcourt un arc de cercle de rayon : $R + b$
- La roue interne parcourt un arc de cercle de rayon : $R - b$

Considérons que le véhicule roule avec une vitesse linéaire constante v . Nous pouvons calculer la vitesse angulaire moyenne ω (pour le cercle d'arc R) par la formule suivante :

$$\omega = \frac{v}{R} \quad (21)$$

Par identification, nous pouvons trouver les vitesses recherchées :

$$v_{\text{intérieure}} = (R - b) \cdot \omega \quad (22)$$

$$v_{\text{extérieure}} = (R + b) \cdot \omega \quad (23)$$

Nous constatons que la vitesse de la roue extérieure au virage est plus grande que celle intérieure au virage. Ceci est cohérent avec nos attentes.

De même que pour l'angle de braquage, au moment de faire notre modèle, il sera mieux de ne pas dépendre du rayon du braquage pour déterminer les vitesses de chaque roue. Cette modification sera aussi abordée dans la section [4.2.](#) .

Chapitre 3. Modification de la plateforme

3. 1. Changement des cartes

Lors de la première phase de ce projet, il a été choisi de travailler avec des boites de commande de type Torque Controller. Celles-ci ont été remplacée au début de notre projet par des cartes dsPICDEM MCLV-2 de Microchip. Ces dernières ont été sélectionnées en raison de ses avantages divers qu'elles apportent. Celles qui nous intéressent le plus sont :

- Présence d'un port CAN permettant d'effectuer la communication entre la carte et la dSpace à l'intermédiaire de 2 fils uniquement.
- Présence d'un capteur à effet Hall et donc d'un contrôle de vitesse du moteur intégré dans la carte. La carte est donc autoasservie.
- Possibilité de contrôler la sortie manuellement par un potentiomètre pour tester divers systèmes sans devoir passer par l'intermédiaire du logiciel dSpace.
- Possibilité d'accorder un récepteur de signal de télécommande à la carte.

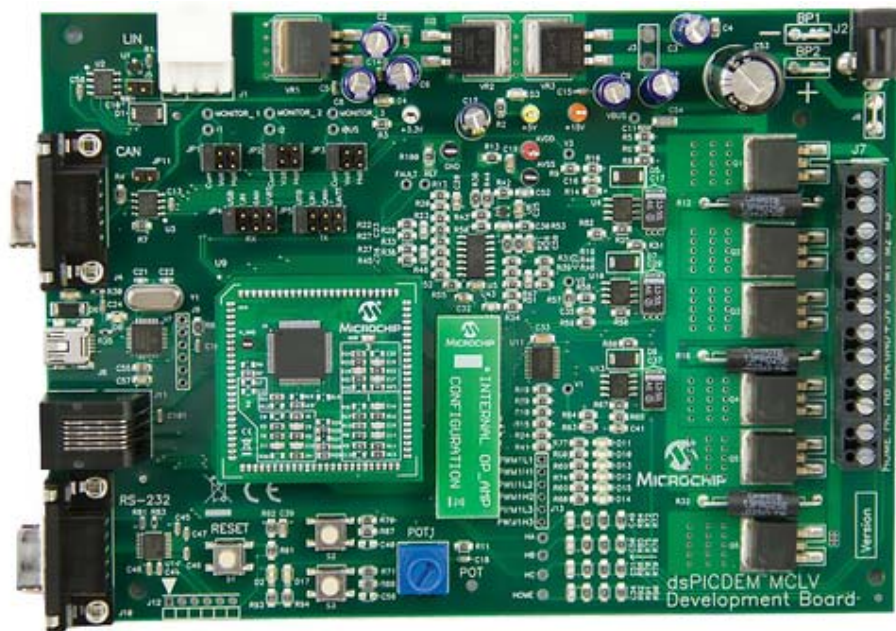


Photo 1 - Carte dsPICDEM MCLV-2.

Celle-ci doit être montée sur des entretoise placée entre elle et le châssis pour éviter toute interférence et court-circuit lors de la présence d'alimentation électrique.

Le lien pour la documentation de ces cartes se trouve en annexe.

3. 2. Changement de l'architecture du véhicule

Pour pouvoir accommoder ces nouvelles cartes sur notre châssis, nous devons changer l'emplacement de certains composants. Ce changement va nous pousser à superposer 2 batteries au milieu du châssis aux cotés droits et gauche. Nous aurons donc 4 batteries déplacées et 4 cartes nouvelles cartes qui auront besoin d'être fixées.

Nous proposons alors de commander les composants suivants :

- 16 Entretoise DR 110 VO fabriqué en Nylon, diamètre intérieur 5.5mm, extérieur 10mm et de longueur 10mm
- 16 vis de diamètre 3 mm et de longueur 15mm
- 16 écrous de diamètre 3 mm
- 8 vis diamètre 6 mm et longueur 100 mm
- 20 écrous diamètre 6 mm
- 10 plaques de fixation de batteries en aluminium 7075 (Dessin technique effectué précédemment)

Le site de commande de ces composants se trouve en annexe.

De plus, pour avoir une idée de la nouvelle position de ces composants, nous décidons de modifier la CAO précédente du véhicule pour visualiser ces changements. Voici une capture d'écran de notre nouveau modèle.

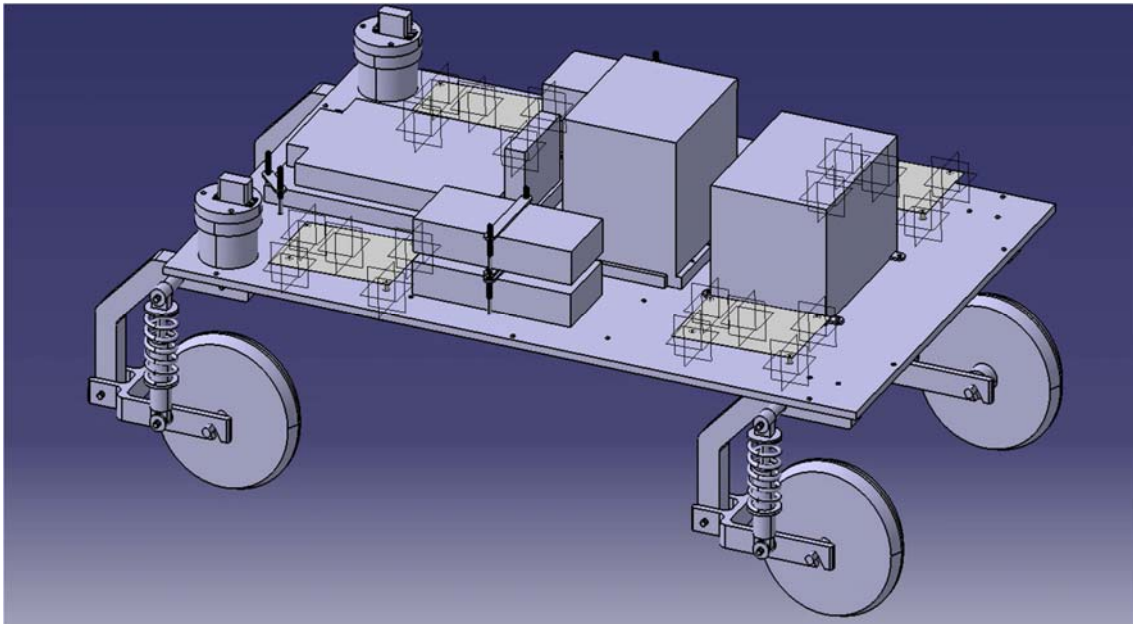


Figure 5 - Nouveau emplacement des composants sur le châssis.

3. 1. Replanification de l'alimentation électrique

Nous décidons de modifier le circuit électrique du véhicule pour ajouter des batteries plus facilement ultérieurement tout en garantissant la sécurité des composants du véhicule pour éviter les chocs électriques.

En effet, lors du branchement direct des batteries aux cartes, des chocs électriques se produisent et se manifestent par l'apparition d'étincelles et d'un bruit. Afin de les éviter, nous décidons de mettre en place un tableau de commande électrique auquel seront connectés les batteries et toutes les cartes Microchip de commande ainsi que la microautobox de la dSpace. Ce tableau aura un interrupteur principal qui coupera l'alimentation lors de l'arrêt du véhicule sans devoir déconnecter les cartes et puis les reconnecter lors du besoin. Nous évitons alors les chocs électriques.

En conséquence, nous allons brancher les 2 batteries fournies au tableau. Chaque batterie va alimenter 2 cartes de contrôle. Mais sachant qu'une des cartes à l'avant ne sera pas utilisée, nous allons coupler une batterie avec la dSpace et la carte.

Voici notre schéma représentant la division du câblage qui sera fourni à monsieur Pierre CANO pour nous fournir les câbles requis ainsi que l'interrupteur :

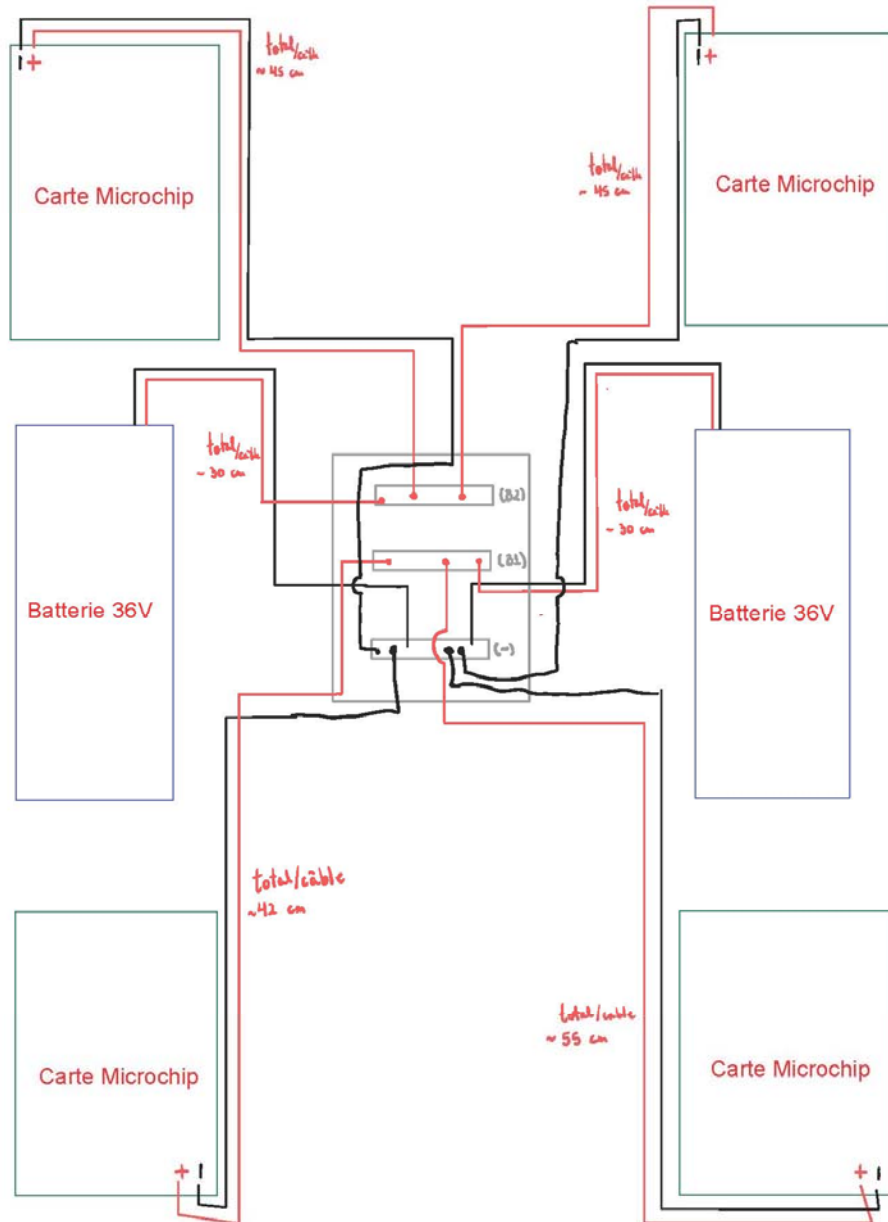


Figure 6 - le plan du nouveau câblage de notre véhicule.

Afin d'accrocher tous ces composants au châssis du véhicule, nous effectuons la conception d'une plaque de fixation destinée à cet effet. Elle permettra d'ajouter un interrupteur entre les câbles des borne positives d'entrées des batteries et les câbles reliés aux cartes Microchip. Elle permet également de relier la borne négative des batteries directement à celles des cartes.

Voici une capture d'écran de cette plaque effectuée sur SolidWorks :

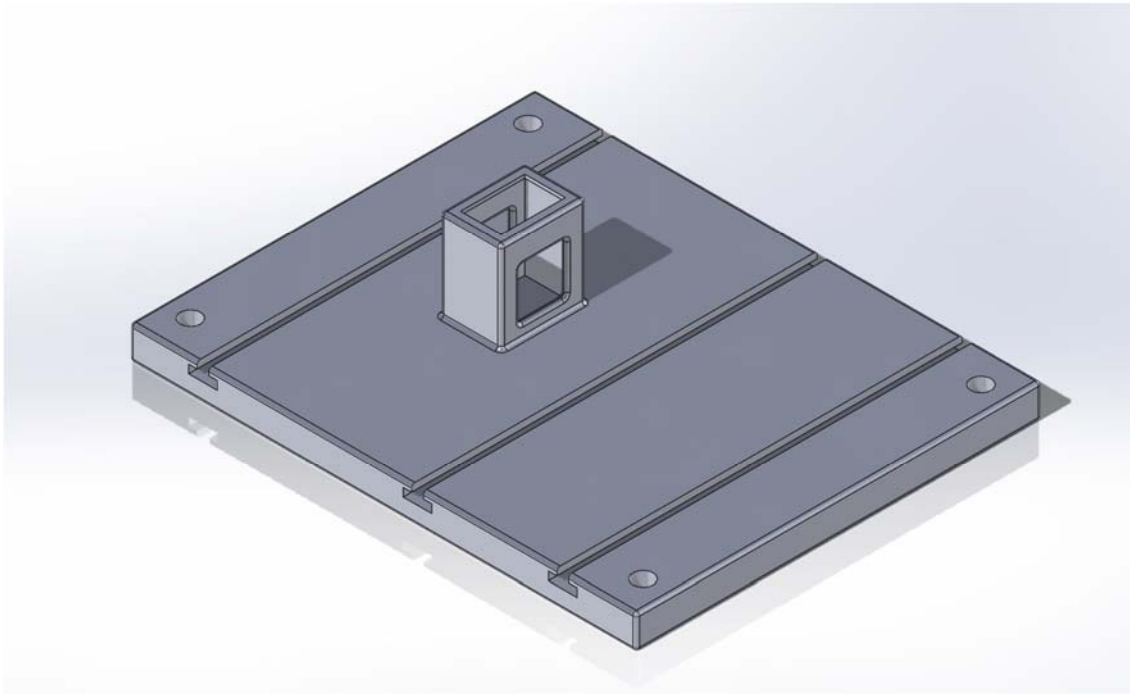


Figure 7 – Plaque pour fixer l'interrupteur et les accords électriques.

Cette plaque sera imprimée à l'aide d'une imprimante 3D. Nous allons utiliser le PLA comme matériel d'impression pour sa capacité à agir comme un isolant électrique. De plus 4 vis de 3 mm de diamètre seront requises pour fixer ce tableau au châssis.

D'ailleurs, après discussion avec Monsieur POULIQUEN, nous remarquons que les batteries ne peuvent pas être montés en parallèle car cela consiste un danger. Nous proposons alors de modifier ce schéma et d'intégrer 2 interrupteurs. Chaque batterie alimentera 2 cartes et une prendra en charge aussi la Microautobox de la dSpace.

Au temps d'écriture de ce rapport, nous avons fixé les composants comme les batteries et le tableau. Toutefois, nous n'avons pas eu le temps de fixer les câbles électriques pour alimenter les composants mais ceci doit s'achever dans les 2 prochains jours de la soumission du rapport.

Voici une photo finale montrant l'état de notre véhicule au temps de la soumission de ce rapport :

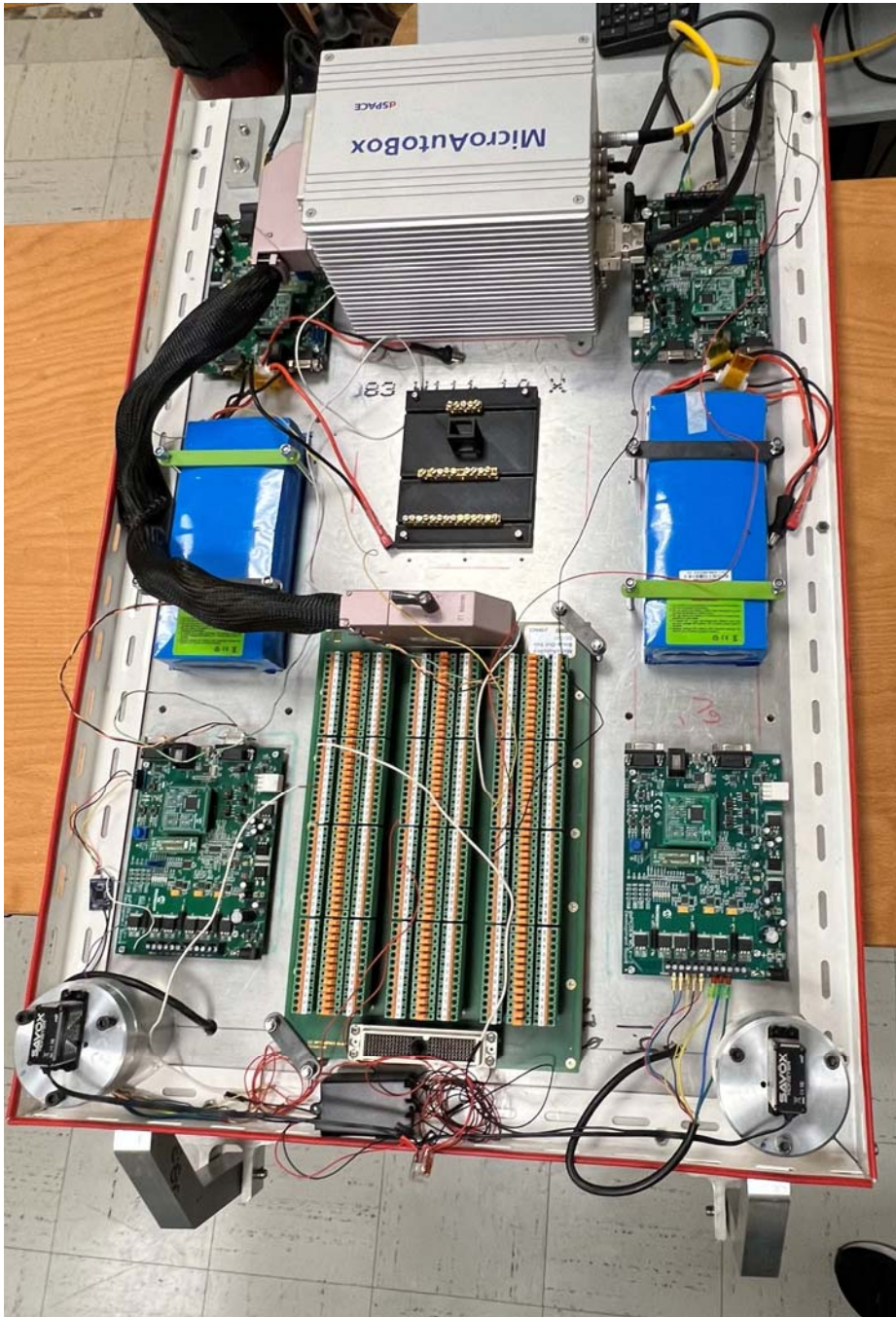


Photo 2 - L'architecture finale de notre véhicule.

Chapitre 4. Programmation de la commande

4. 1. Schéma entrée/sortie de la dSpace

Pour avoir un plan des variables mise en jeu dans notre programme, nous proposons de s'organiser en les représentant sur un schéma en divisant les entrées et les sorties.

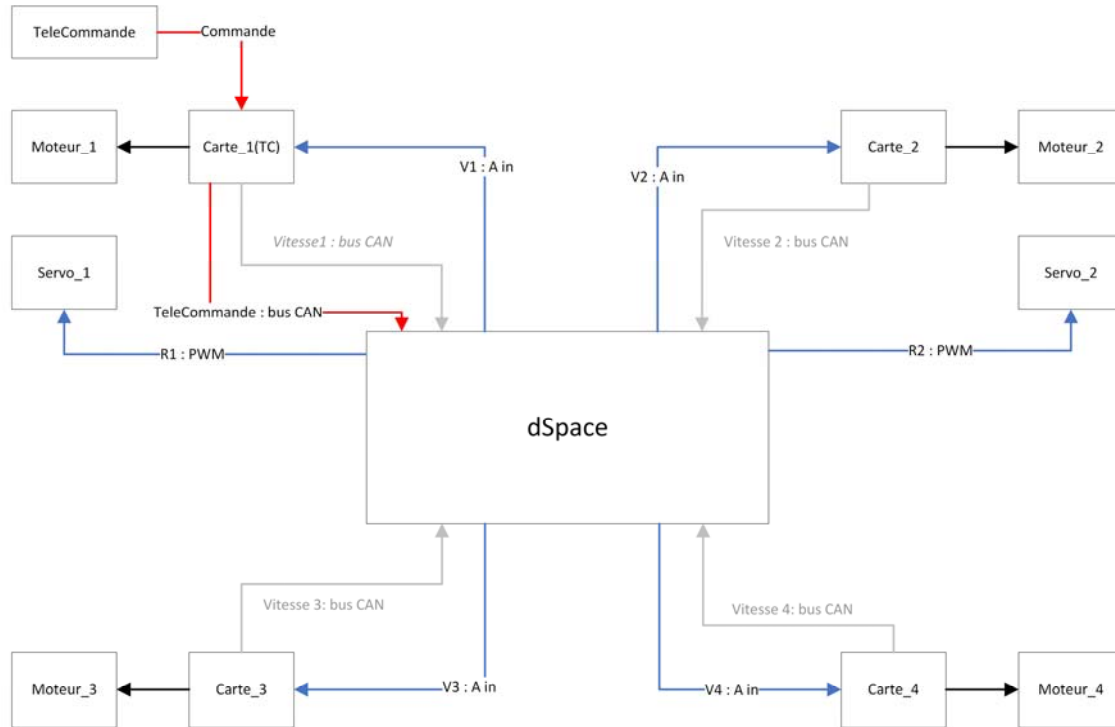


Figure 8 - Schéma des variables mises en jeu par la dSpace.

Nous notons que les servo-moteurs seront reliés directement à la dSpace et non pas aux cartes de commande comme les moteurs à courant continu.

Nous devons envoyer 4 signaux de vitesse et, aussi, récupérer 4 vitesses réelles issues de chaque moteur pour s'assurer du bon fonctionnement de notre programme en boucle fermée. Cependant, la partie de la boucle fermée n'a pas abouti pendant la réalisation de ce PRI, par indications du responsable. Pour cela, la partie de retour des vitesses est marquée en gris.

Enfin, un signal spécial de télécommande en termes de vitesse et de braquage doit être lu aussi pour pouvoir lancer la commande et renseigner des valeurs aux variables précédentes. Ce signal est relié à une des cartes qu'on doit extraire par l'intermédiaire du CAN. Celui-ci va donc envoyer un signal de 64 bits, que l'on divise en 4 signaux séparés de 16 bits chacun. Chaque morceau du signal représente un degré de liberté de la télécommande (donné par le

mouvement horizontal ou vertical du levier droit ou gauche du contrôleur) qui est lu comme un entier non signé compris entre environs 5 600 bits et 58 000 bits.

Ainsi, le signal du mouvement vertical du levier gauche est assigné à la vitesse du véhicule, lorsque le signal du mouvement horizontal du levier droite est assigné à l'angle de braquage. Les deux signaux restants ne seront pas utilisés pendant ce projet, mais restent comme une alternative intéressante si l'on veut contrôler de manière spéciale le véhicule.

En outre, nous configurons nos cartes de sorte que la commande envoyée vers le moteur est en fonction uniquement de la tension de commande. C'est-à-dire que pour une commande comprise dans l'intervalle $[0 ; 1.65]$ volts, le moteur est en marche arrière avec la tension 0 V correspondant à la vitesse maximale arrière du moteur et à 1.65V le moteur est à l'arrêt.

Pareillement, pour une tension comprise dans l'intervalle $[1.65 ; 4.4]$ volts, le moteur est en marche avant avec la commande de 3.3 V correspondant à la vitesse maximale et 1.65 V à la vitesse nulle.

4. 2. Mise en place d'une modélisation 3D

Pour visualiser la mise en marche de notre véhicule, nous avons décidé d'implémenter une simulation du modèle en 3D en Simulink. Pour ce faire, nous nous sommes appuyés sur un fichier d'utilisation publique de MathWorks, « VehicleDynamics3D.slx », dans lequel l'on simule la dynamique d'un véhicule de compétition étudiant, en contrôlant la vitesse et l'angle de virage (cf. Figure 9).

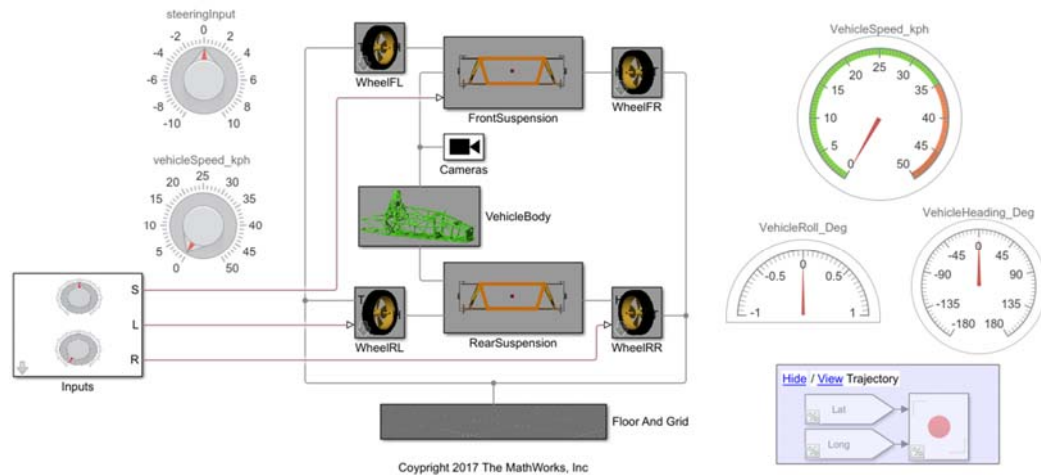


Figure 9 – Modèle Simulink de Mathworks pour observer et piloter un véhicule en 3D.

Du modèle, nous avons conservé la structure extérieure, mais nous l'avons adaptée à notre véhicule. Pour ce faire, nous avons modifié la manière dont nous contrôlons la direction de la transmission et la commande des vitesses. Ainsi, le modèle de MathWorks marchait comme un véhicule physique, c'est-à-dire, en contrôlant la distance parcourue par un essieu, ce qui provoque le mouvement des rotules de direction. En revanche, nous avons établi un modèle dans lequel nous pouvons choisir les angles que chaque roue doit parcourir selon les équations présentées en 2. 6. de sorte que nous fournissons un angle précis pour la roue droite et un autre pour la gauche.

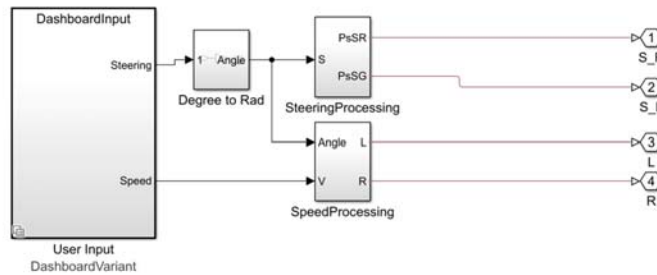


Figure 10 – Schéma de modification de l'input avec les sous-systèmes *SteeringProcessing* et *SpeedProcessing*.

Nous avons refait les pièces à l'aide du logiciel de CAO SolidWorks, en nous basant sur les plans du véhicule provenant du PRI précédent qui l'a conçu. Une fois les pièces et les assemblages réalisés, nous avons utilisé le plugin Simscape Multibody dans SolidWorks pour créer un modèle Simscape Multibody à partir des géométries et des matériaux spécifiés, notamment de l'aluminium et du caoutchouc pour les roues. Ensuite, nous avons pris les blocs correspondants aux roues, au châssis et aux connexions de rotation, et nous les avons intégrés dans la structure du fichier "VehicleDynamics3D.slx".

Pour faire fonctionner le modèle, nous avons spécifié les positions d'équilibre des amortisseurs ainsi que leur constante de rigidité ($k = 650 \text{ lbf/in}$) et de dissipation ($c \sim 100 \text{ N/(m/s)}$). Il a également été nécessaire de modifier les plans de connexion des liaisons de contact entre le sol et les roues données directement par le plugin auparavant mentionné. Pour ces liaisons de contact, nous avons utilisé les blocs « Sphere to Plane Force » de la bibliothèque de Matlab, "Contact Forces Library", modélisant les roues comme des sphères de rayon 0.1 m et le sol comme un plan. Les paramètres utilisés pour établir le contact ont été pris du modèle de MathWorks, car nous ne disposions pas des données mesurées et, pour avoir une première idée de la dynamique du véhicule, ce n'est pas une mauvaise approximation.

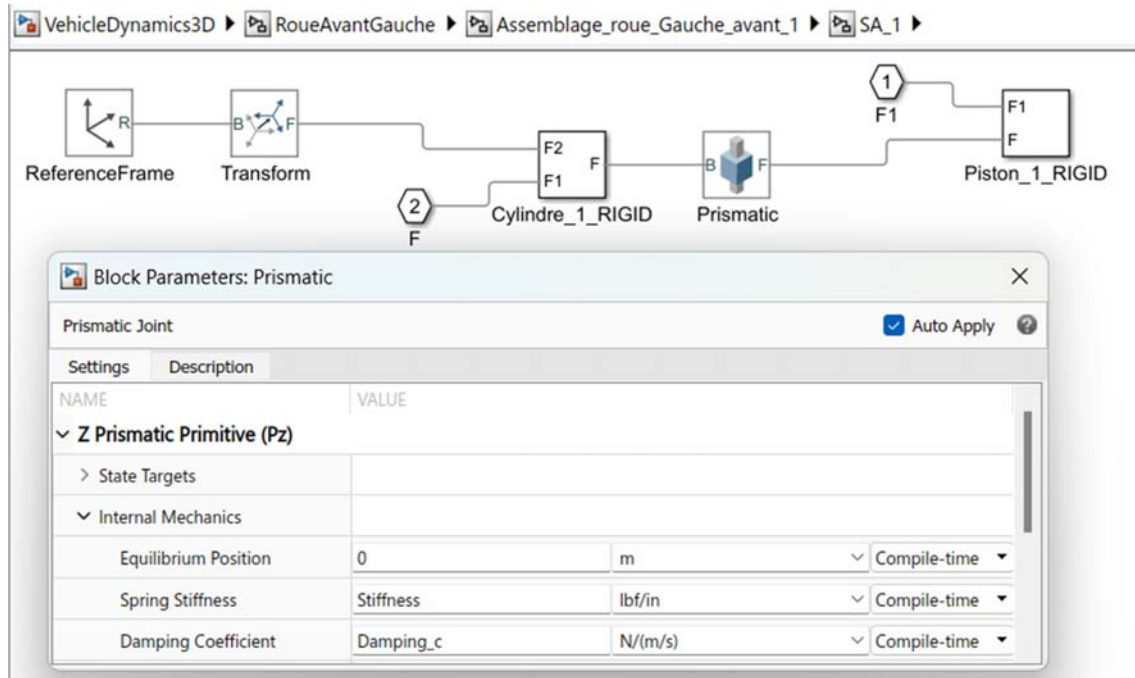


Figure 11 – Assignment des valeurs de la mécanique interne des amortisseurs.

D'autre part, le modèle de base venait avec un Mask comportant des options pour modifier le type de mouvement du véhicule en contrôlant les blocs « variants » avec une liste déroulante. Cependant, cela ne fonctionnait pas, probablement parce qu'il avait été créé avec une version antérieure de Simulink. Pour remédier à ce problème, le code du Mask a été modifié en utilisant la nouvelle configuration, en particulier en utilisant la fonction `assignin('base','variable',value)` pour affecter les valeurs aux variables dans le workspace de 'base'.

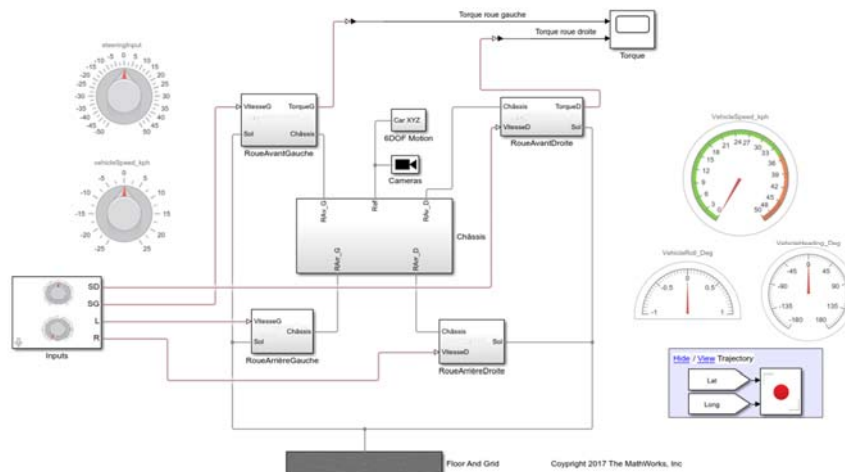


Figure 12 – Modèle 3D final du véhicule

L'on peut ajouter des capteurs des forces, couples, vitesses, etc... Nous avons ajouté un capteur de couple pour les roues avant comme démonstration, mais on peut le faire avec plusieurs capteurs en se rendant sur les paramètres du bloc de l'union correspondant et faire le traitement en Matlab.

Enfin, sur Model Properties » Model pre-load function, on a ajouté un appel au fichier Plateforme_VE_DataFile à la liste qui contenait déjà le script VehicleDynamicsInitialization. Ce fichier permet d'obtenir toutes les valeurs d'inertie, de masse et taille des composants 3D importés depuis SolidWorks, et il est donné automatiquement par le plugin.

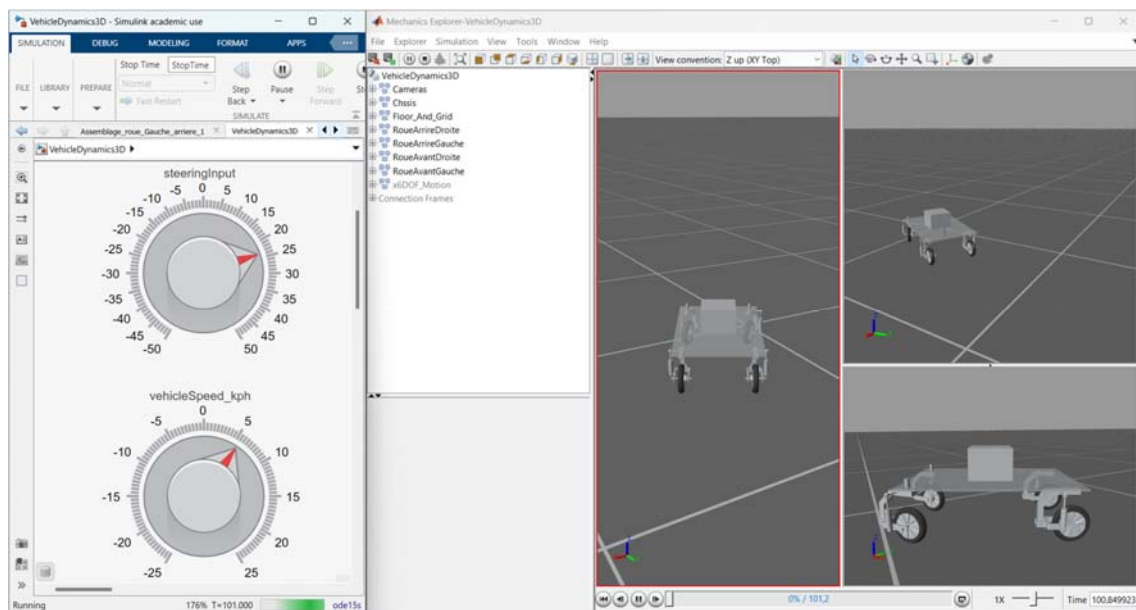


Figure 13 – Véhicule en 3D en train d'être contrôlé par l'utilisateur du Simulink

4. 3. Modèle final retenu

4. 3. a. Quelques modifications

Pour finaliser notre projet, nous effectuons quelques modifications sur la plateforme qui sont :

- Nous nous contentons d'utiliser la marche arrière uniquement pour motoriser notre véhicule. C'est-à-dire que les 2 cartes avant seront déconnectés des roues et nous utiliserons une d'entre elles pour récupérer les signaux de la télécommande grâce au CAN pour les exploiter ultérieurement.
- Nous allons utiliser uniquement 2 batteries de 36V pour alimenter nos cartes et la MicroAutobox. D'autres batteries peuvent être ajoutés ultérieurement.

4.3. b. Composition de notre modèle de commande Simulink

Nous construisons alors notre modèle Simulink pour la commande des roues arrière et du braquage. Voici une capture d'écran de celui-ci qui sera expliqué :

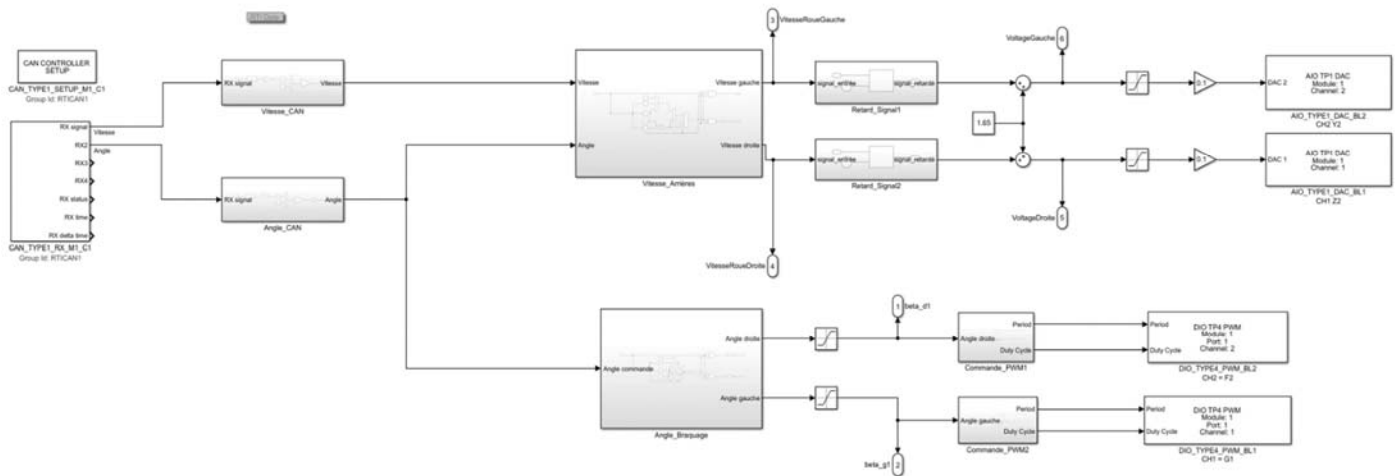


Figure 14 - Modèle Simulink final pour la commande du véhicule.

Pour une meilleure présentation, nous divisons notre modèle en plusieurs sous-systèmes. Nous allons les détailler dans la suite de cette partie.

- Bloque : « Vitesse_CAN » et « Angle_CAN »

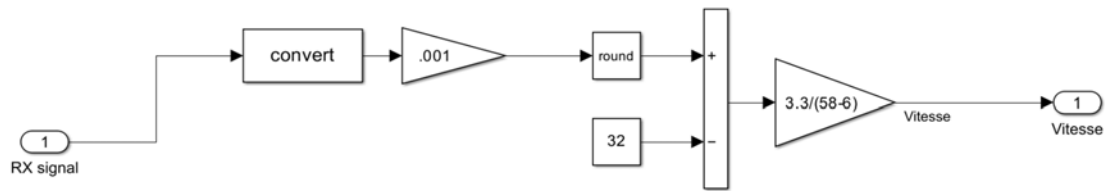


Figure 15 – Sous-système « Vitesse_CAN ».

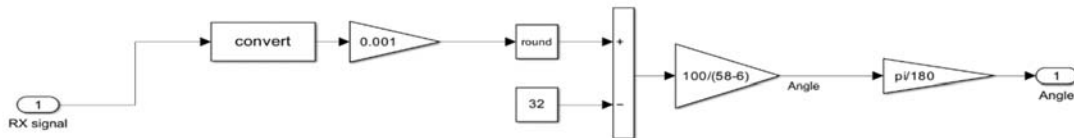


Figure 16 – Sous-système « Angle_CAN ».

Ces 2 blocs sont utilisés pour convertir le signal reçu de la télécommande à travers le CAN en des signaux de commande pour la vitesse et l'angle de braquage.

Nous recevons un signal du CAN compris entre 6 000 et 58 000 bits à peu près. Nous les arrondissons à l'unité près en utilisant round et en « centrant » la valeur sur 0. Le signal correspondant au recul du véhicule sera compris entre 6 000 et 32 000 bits. Celui correspondant à l'avancement du véhicule entre 32 000 et 58 000 bits. Le « zéro de vitesse » sera donc à la valeur de 32 000 bits.

Ensuite nous convertissons ce signal en valeur efficace : une valeur représentant la tension entre 0 et 3.3V pour la vitesse globale du véhicule et un angle général de commande entre -50 et 50 degrés.

- Sous-système « Vitesse_Arrières » :

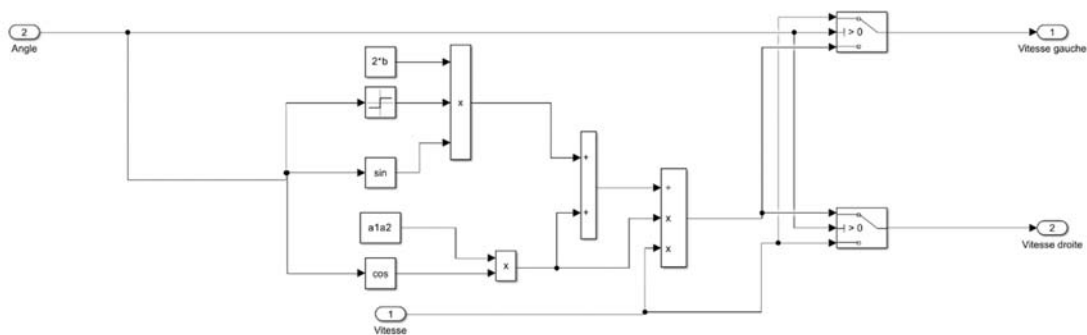


Figure 17 - Sous-système "Vitesse_Arrières".

Ce système contient l'effet de l'ajout d'un différentiel pour assister le braquage du véhicule. Il permet de faire varier la vitesse de commande de chaque roue en fonction de l'angle de braquage. Lorsque le véhicule roule avec un rayon déterminé, la roue interne au virage circulera moins rapidement que la roue externe. Nous attribuons alors la vitesse maximale de commande à la roue externe et celle de la roue interne est déterminée à l'aide des calculs établis dans la partie dynamique du véhicule.

Il permet également de maintenir la valeur de la vitesse de commande égale dans les 2 roues lorsque le rayon de braquage est nul. Ces 2 variables auront une valeur identique lorsque le véhicule circule en ligne droite.

Ce sous-système va donc avoir en sortie 2 variables encaissant une valeur de la tension de commande de chaque roue en fonction de l'entrée de la télécommande.

- Sous-système « Angle_Braquage »

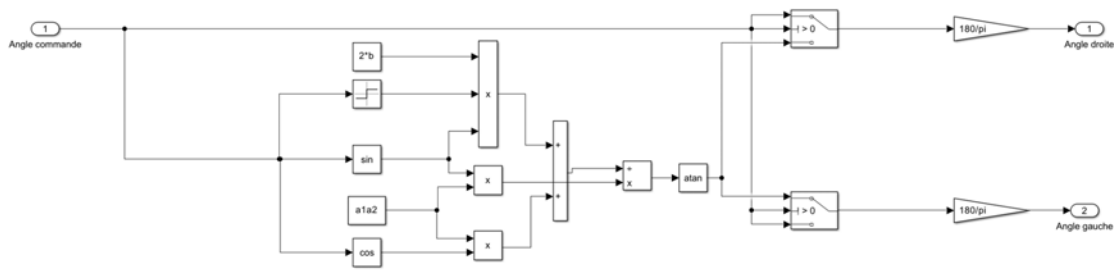


Figure 18 – Sous-système « Angle_Braquage ».

En utilisant les équations trouvées lors de l'étude de la dynamique du véhicule (partie 2. 6.) nous construisons ce sous-système. La particularité de celui-ci est que l'angle maximal de braquage ne peut pas dépasser les 50 degrés dans les 2 sens. Nous devons donc attribuer la valeur maximale à l'angle de la roue de la direction correspondante et une valeur diminuée pour l'angle de braquage de la roue opposée. Ce dernier est trouvé à partir des équations de la dynamique du véhicule présentées dans la partie 2. 6. Ainsi, les équations modifiées résultantes sont :

$$\beta_{Ext} = \beta_{commande} \quad (24)$$

$$\beta_{int} = \frac{\beta_{ext} \cdot a_1 a_2 \sin \beta_{ext}}{a_1 a_2 \cos \beta_{ext} + \text{sign}(\beta_{ext}) \cdot 2b \sin \beta_{ext}} \quad (25)$$

- Sous-système « Commande_PWM »

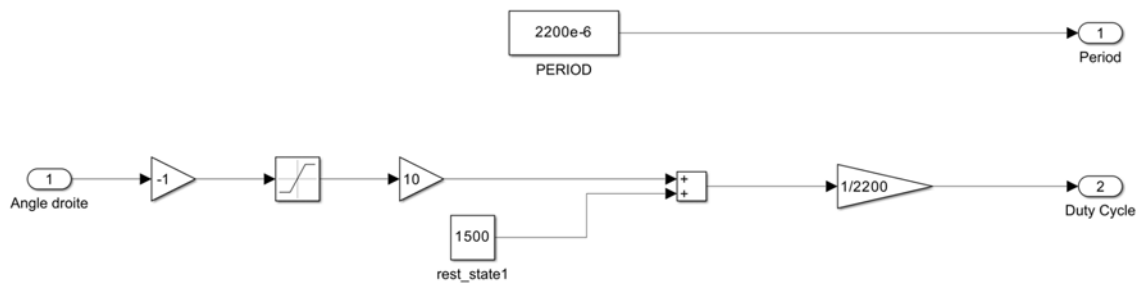


Figure 19 – Sous-système « Commande_PWM ».

Dans ce bloc, nous nous contentons de transformer la valeur de l'angle calculée en un signal PWM pour pouvoir l'envoyer au servo-moteur pour tourner les roues. C'est le même bloc pour les 2 roues.

- Sous-système « Retard_Signal » :

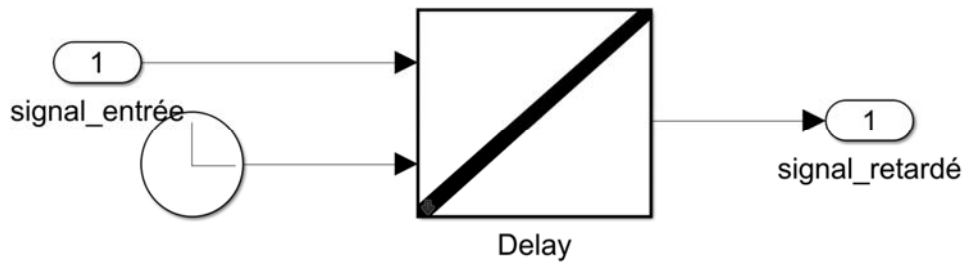


Figure 20 – Sous-système « Retard_Signal ».

Ce bloc incorpore une fonction qui a pour but d'imposer un retard au signal de commande pour éviter les à-coups lors du changement brusque du sens de roulement du véhicule. Ceci permet d'avoir une accélération plus fluide du véhicule et un ralentissement moins brusque, évitant donc le glissement du véhicule. Ainsi, la fonction est composée de deux parties, celle de l'arrivée à zéro et celle de la limitation du taux de changement (similaire au bloc « Rate Limiter »). Nous avons décidé d'utiliser une décroissance exponentielle pour ralentir les roues pendant un temps prudent pour qu'elles s'arrêtent. Pour cela, nous avons créé un Mask pour pouvoir changer les paramètres de la fonction sans changer le code. Spécifiquement, les paramètres que l'on peut changer sont :

- Rate : Valeur qui sert à faire une descente plus vite (Rate plus grande) ou plus lente (Rate plus petite).
- Time at 0 : Le temps que la roue reste à 0 après s'être presque arrêtée (avec une valeur d'approximativement 0.016 V d'écart par rapport au voltage d'arrêt).
- Rampe : Valeur absolue maximale de la pente pour la commande des roues, remplace les variables « Rising slew rate » et « Falling slew rate » du bloc *Rate Limiter*.

- Les blocs « saturation »

Dans notre modèle nous utilisons principalement les blocs saturation pour éviter le risque de surtension de la vitesse ou de sur-commande de l'angle de braquage. Ils imposent un filtre aux signaux et laissent passer uniquement les commandes dans les intervalles indiqués : [0 ; 3.3] volts pour la vitesse et [-50 ; 50] degrés pour l'angle de commande. Nous choisissons de les ajouter après avoir fait les calculs des angles et vitesses.

Chapitre 5. Problèmes rencontrés

Durant notre projet, nous faisons face à plusieurs problèmes qui ont ralenti son avancement. D'abord, nous remarquons que certaines cartes de commande et moteurs ne fonctionnaient pas comme prévu. Ce problème a été réglé par la commande de nouvelles cartes pour remplacer les anciennes et par le remplacement d'un fil électrique dans le moteur avec l'aide du laboratoire de l'INSA pour le rendre fonctionnel.

D'ailleurs, nous avons dû effectuer une soudure des fusibles sur les cartes de commande à cause de l'apparition d'un court-circuit lors du contact avec le châssis du véhicule. Les cartes ont été ensuite élevée du châssis à l'aide d'entretoises isolantes pour palier à ce problème et aussi les fixer.

En outre, un problème avec la licence de la dSpace a retardé considérablement notre projet. En effet, nous n'avons pas eu accès à plusieurs composants de la librairie Matlab afin de commander notre véhicule. Nous ne pouvons donc pas communiquer efficacement entre la dSpace et les cartes pour les commander et lire le signal de la télécommande.

Une autre difficulté majeure qui a survenu dans notre projet était le rasage des engrenages utilisé pour raccorder les servo-moteurs aux axes des roues afin d'assurer la commande en virage des roues avant. Ceux-ci se sont abimés à la suite d'un montage précédent non idéal. Nous avons pu commander de nouveaux engrenages pour réaliser le braquage. Mais après plusieurs essais, ceux-ci n'ont pas tenu également. Il est donc envisageable d'établir une reconception du système de virage entièrement pour palier à ce problème.

Plusieurs propositions peuvent convenir pour la reconception du système de braquage notamment l'utilisation d'un train épicycloïdal au lieu d'un seul engrenage par exemple, ou l'implémentation d'un système d'engrenages et de vis sans fin.

Chapitre 6. Conclusion et perspectives

Ce projet a fait face à plusieurs problèmes qui ont ralenti son avancement mais a pu bien évoluer quand même. Une modification de l'architecture du véhicule a été faite pour accommoder une augmentation du nombre de batteries et donc de l'autonomie ainsi que de la sécurité électrique du véhicule. La commande en boucle ouverte de la dSpace nous permet de télécommander le véhicule pour l'avancer, le reculer et effectuer des virages. La simulation 3D nous a également permis de voir le comportement du véhicule sur Matlab avant de le tester réellement.

Toutefois, plusieurs pistes d'amélioration s'ouvrent dans notre projet. Nous proposons de connecter les moteurs des roues avant afin d'aider dans la traction du véhicule et d'augmenter le couple effectif fourni. Une commande en boucle fermée peut aussi être mise en place pour limiter le couple à fournir aux roues. Celle-ci peut être mise en place grâce à l'implémentation des CAN sur toutes les cartes, chose qui permet de recueillir des informations (comme le couple réel) et d'envoyer d'autre en même temps sur une carte.

D'ailleurs, une reconception du mécanisme de braquage est requise. Le matériel utilisé pour l'engrenage collé à l'axe de la roue ne supporte pas le poids de celle-ci est à tendance à se raser et donc sera inefficace dans son rôle. Nous proposons un meilleur assemblage en utilisant un train épicycloïdal ou un système d'engrenage avec vis-sans-fin.

Enfin, nous proposons de changer les servomoteurs qui sont utilisés pour braquer le véhicule. En intégrant un moteur à trois fils, nous pouvons recueillir des informations de ce dernier, ce qui permettra de limiter le couple envoyé au moteur. Ceci a pour potentiel de limiter l'endommagement des engrenages du système de rotation.

Chapitre 7. Bibliographie

- Documentation des cartes sdPICDEM MCLV-2 :
<https://www.microchip.com/en-us/development-tool/dm330021-2>
- 16 Entretoise DR 110 VO fabriqué en Nylon, diamètre intérieur 5.5mm, extérieur 10mm et de longueur 10mm.
<https://fr.rs-online.com/web/p/entretoises/1364254?gb=s>
- 16 vis de diamètre 3 mm et de longueur 15mm
<https://fr.rs-online.com/web/p/vis-a-six-pans/8229063>
- 16 écrous de diamètre 3 mm
<https://www.vis-express.fr/fr/ecrou-hexagonal-classe-classe-88-iso-4032-nfe-24032-iso-4032vs-iso4032vs4325/31109-733124-ecrou-hexagonal-m3-acier-zingue-noir-3663072087644.html>
- 8 vis diamètre 6 mm et longueur 100 mm
<https://www.leroymerlin.fr/produits/quincaillerie/cheville-vis-clou-et-boulon/vis/vis-pour-metal/tfhc-m6x100-inox-a4-ef-din-7991-88273110.html>
- 20 écrous diamètre 6 mm
<https://www.leroymerlin.fr/produits/quincaillerie/cheville-vis-clou-et-boulon/ecrou/30-ecrous-hexagonal-acier-6-mm-standers-82583005.html>
- Oi Toru, et al, « Electrical Properties of Heat-Treated Poly-Lactic Acid » , The transactions of the Institute of Electrical Engineers of Japan, vol. 131, no 5, p. 395-400. <https://doi.org/10.1541/ieejfms.131.395>.