

Simulateur à retour haptique de véhicules basé sur des modèles multicorps temps réels

Mémoire présenté par
Romain POLETTI

en vue de l'obtention du grade de Master
ingénieur civil électromécanicien

Promoteur(s)
Paul FISETTE

Lecteur(s)
Dehez BRUNO, Delannay LAURENT , Verlé AUBAIN

Année académique 2016-2017

L'importance du secteur automobile n'est aujourd'hui plus à démontrer et fait même de la voiture le mode de transport le plus utilisé. De part l'importance d'une telle demande, l'étude de la dynamique d'un véhicule se retrouve dans tout processus de conception. L'étude peut être menée sur des prototypes réduits, des prototypes à échelle réelle mais leur coût est un frein majeur à cette voie.

L'essor du développement de simulateurs de véhicule ces dernières années a répondu à cette problématique et offre un moyen rapide, fiable et peu coûteux pour la conception ou l'analyse du rapport homme-véhicule.

Remerciements

Ce mémoire n'aurait pu être réalisé sans l'aide de nombreuses personnes, je tiens à les remercier pour l'expérience qu'elles m'ont apportée.

Je remercie le Professeur Paul Fisette pour sa confiance et son soutien accordés tout au long de cette année.

Je remercie Aubain Verlé sans qui le projet n'aurait pu aboutir. Je le remercie pour son suivi et tout particulièrement pour tous ses conseils tant en mécanique, informatique qu'au niveau de la rédaction. Je le remercie également pour sa disponibilité et le temps important qu'il m'a consacré pendant la totalité du projet.

Je remercie Martin Vanderwegen pour sa disponibilité et son aide indispensable dans le développement sur micro-contrôleur.

Je remercie également Nicolas Docquier, Quentin Docquier et Olivier Lantsoght pour leur disponibilité ainsi que leurs conseils et intérêt apportés au projet.

Finalement, je remercie ma famille pour son soutien et la relecture du mémoire.

Table des matières

| | |
|--|-----------|
| Remerciements | 5 |
| Introduction | 9 |
| 1 Méthode de simulation dynamique | 11 |
| 1.1 Formalisme multicorps | 11 |
| 1.1.1 Modélisation | 11 |
| 1.1.2 Génération des équations | 12 |
| 1.1.3 Résolution des équations | 12 |
| 1.2 Véhicule étudié | 15 |
| 1.2.1 Modélisation | 15 |
| 1.2.2 Implémentation des forces | 17 |
| 1.3 Conclusion | 19 |
| 2 Simulateur de l'UCL | 21 |
| 2.1 Définition de l'haptique | 21 |
| 2.2 Vue globale du dispositif | 21 |
| 2.3 Modules du simulateur | 22 |
| 2.3.1 Module multicorps | 22 |
| 2.3.2 Module actionneur | 24 |
| 2.3.3 Module de transmission | 26 |
| 2.4 Protocoles de communication | 27 |
| 2.4.1 Communication série | 27 |
| 2.4.2 Communication CANopen | 28 |
| 2.5 Architecture de contrôle | 29 |
| 2.6 Problèmes subsistants | 32 |
| 3 Développement du simulateur | 35 |
| 3.1 Architectures de contrôle | 35 |
| 3.1.1 Analyse de l'architecture sérielle | 35 |
| 3.1.2 Architecture Multi-Thread | 37 |
| 3.1.3 Architecture avec interruptions | 42 |
| 3.2 Entrées et sorties du modèle multicorps | 46 |
| 3.3 Analyse du modèle multicorps | 48 |
| 3.3.1 Influence du type de modèle multicorps | 49 |
| 3.3.2 Influence de la relaxation du pneu | 50 |
| 3.3.3 Influence de la raideur du pneu | 52 |
| 3.3.4 Influence de la résolution du codeur | 53 |
| 3.4 Assemblage mécanique | 53 |
| 3.5 Conclusion | 55 |

| | | |
|----------|--|-----------|
| 4 | Expériences sur simulateur | 57 |
| 4.1 | Objectifs | 57 |
| 4.2 | Procédure | 58 |
| 4.2.1 | Questionnaire | 58 |
| 4.2.2 | Expériences | 60 |
| 4.3 | Hypothèses | 63 |
| 5 | Résultats | 65 |
| 5.1 | Outil d'analyse | 65 |
| 5.2 | Questionnaire | 66 |
| 5.3 | Expériences | 68 |
| 5.3.1 | Algorithme | 68 |
| 5.3.2 | Analyse | 72 |
| 5.3.3 | Conclusion | 77 |
| 6 | Perspectives et améliorations | 79 |
| 6.1 | Améliorations du simulateur sans extension de l'hardware | 79 |
| 6.2 | Améliorations du simulateur avec extension de l'hardware | 80 |
| | Conclusion | 81 |

Introduction

L'automobile est le premier moyen de transport dans le monde. Selon les derniers chiffres publiés par l'Association Européenne des Constructeurs Automobiles, 256 millions de véhicules sont en circulation en Europe et plus d'une personne sur deux possède un véhicule. L'importance de la connaissance du comportement routier du conducteur et de son véhicule est largement motivée. Cependant, certains de ces aspects restent encore méconnus.

Pour l'étudier, effectuer des expériences réelles n'est pas sans contraintes. Des coûts excessifs mais aussi la nécessité d'assurer la sécurité du conducteur limitent leur déroulement.

Dès lors, un outil de simulation de véhicule est entièrement justifié. Une fois conçu, un simulateur permet de tester indéfiniment à des coûts réduits n'importe quel type de véhicule routier dans des situations différentes.

Nicola De Lellis et *Quentin Materne* ont conçu un simulateur dans le cadre de leur mémoire réalisé à l'UCL. Ce simulateur permet à son utilisateur de conduire des véhicules en lui fournissant un retour haptique au volant. Cependant, certains problèmes subsistaient à l'issue de leur travail et un deuxième mémoire a été réalisé afin de les résoudre.

Ce travail s'articule en 6 chapitres qui peuvent être scindés en deux parties : la première caractérise et développe le simulateur et la deuxième en fait usage.

La première partie de ce travail présente la méthode de simulation dynamique indispensable pour simuler le mouvement du véhicule. En fin de chapitre, un modèle de véhicule implémenté pour le simulateur illustre les concepts théoriques.

Le deuxième chapitre présente le simulateur. Le simulateur est décrit tel qu'il a été achevé par les étudiants du premier mémoire. La mise en évidence des imperfections du simulateur termine le chapitre.

Le troisième chapitre présente les développements informatiques et mécaniques apportés au simulateur.

Le quatrième chapitre initie la deuxième partie de ce travail. Il expose une procédure d'expérience qui vise à analyser l'interaction entre l'homme et le véhicule ou plus particulièrement l'homme et le volant.

Le cinquième chapitre analyse les résultats des expériences.

Finalement, les perspectives du simulateur sont évoqués et le travail est conclu.

Chapitre 1

Méthode de simulation dynamique

La base de tout simulateur de véhicule est un logiciel qui modélise le véhicule et son mouvement. Le mouvement d'un véhicule est caractérisé par un ensemble d'équations que le logiciel doit d'une part générer et d'autre part résoudre afin de décrire précisément l'évolution temporelle du véhicule.

Différents logiciels existent à cette fin et sont repris dans l'état de l'art réalisé par les étudiants du mémoire précédent [12]. Pour le simulateur de l'UCL, une des contraintes est l'utilisation du logiciel *Robotran*.

Ce chapitre vise donc à expliquer comment *Robotran* décrit le mouvement d'un véhicule et applique cette théorie à un kart, modèle de véhicule étudié pour ce travail.

1.1 Formalisme multicorps

Robotran, développé au sein de l'UCL-CEREM, est un générateur symbolique des équations du mouvement (cinématique et dynamique) d'un système multicorps [14].

L'utilisation du logiciel est décomposable en plusieurs étapes correspondant chacune à un module. Nous allons brièvement expliciter les étapes importantes.

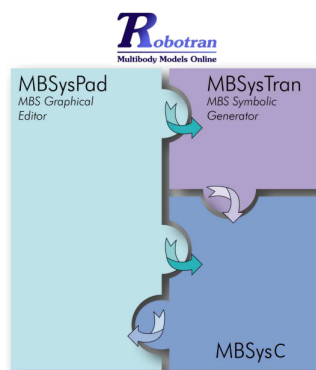


FIGURE 1.1 – Les trois modules principaux de *Robotran* ([14] modifié)

1.1.1 Modélisation

La première étape consiste à modéliser le système (véhicule, pendule, ...) duquel on souhaite utiliser les équations du mouvement : c'est le rôle de *MBSysPad* (figure 1.1). Cet éditeur graphique définit complètement *et une fois pour toutes* le système multicorps composé de corps

rigides ponctuels liés entre eux par des articulations.

Les corps ont une masse, un centre masse, une matrice d'inertie ainsi qu'un ou plusieurs points définissant leur géométrie.

Les articulations (ou joints) ont un degré de liberté (rotation ou translation) chacune. C'est la combinaison des articulations qui modélise les mouvements relatifs entre les corps ayant jusque 6 degrés de liberté.

La modélisation du kart est repris en section 1.2.

1.1.2 Génération des équations

La deuxième étape consiste à générer symboliquement les équations du mouvement à partir du modèle. C'est le rôle du sous-module *MBSysTran*. Dès que l'utilisateur est satisfait du modèle, il génère ses équations qui ne devront ensuite plus être régénérées. C'est l'une des forces du calcul symbolique avec sa puissance pour simplifier des équations (\rightarrow rapidité) et sa portabilité.

Ces équations sont générées sous un formalisme relatif : les coordonnées des points d'un corps sont exprimées à partir de la base du corps. Le détail, les avantages et inconvénients de ce formalisme sont discutés dans la référence [1].

Avant d'entamer la troisième et dernière étape, il convient de rappeler les équations du mouvement d'un système multicorps :

$$M(q)\ddot{q} + c(q, \dot{q}, \text{frc}, \text{trq}, g) = Q(q, \dot{q}) + J^t \lambda \quad (1.1)$$

A cette équation, sont ajoutées les contraintes en position, vitesse et accélération pour fermer complètement le système multicorps :

$$h(q) = 0 \quad (1.2)$$

$$\dot{h}(q, \dot{q}) = 0 \quad (1.3)$$

$$\ddot{h}(q, \dot{q}, \ddot{q}) = 0 \quad (1.4)$$

Pour un système multicorps à N_{coord} articulations et N_{contr} contraintes, les termes de l'équation sont :

| Nom | Détail | Taille |
|-----------|--|------------------------------|
| M | Matrice de masse | $N_{coord} \times N_{coord}$ |
| q | Vecteur des coordonnées généralisées | $N_{coord} \times 1$ |
| c | Vecteur dynamique non-linéaire et des forces/couples extérieures | $N_{coord} \times 1$ |
| Q | Vecteur des forces/couples articulaires | $N_{coord} \times 1$ |
| J | Matrice jacobienne des contraintes | $N_{contr} \times N_{coord}$ |
| λ | Multiplicateurs de Lagrange associés aux contraintes | $N_{contr} \times 1$ |

Les coordonnées généralisées sont définies par les articulations modélisées sur *MBSysPad*.

1.1.3 Résolution des équations

La dernière étape consiste à résoudre ce système d'EDA¹ (1.1 - 1.4). Les \ddot{q} doivent être trouvés à partir de conditions initiales q, \dot{q} . *MBSysC* implémente un partitionnement de coordonnées à cette fin.

1. Équations différentielles algébriques.

Partitionnement

L'idée sous-jacente au partitionnement est de séparer l'ensemble des coordonnées généralisées en deux sous-ensembles : 1. les variables indépendantes q_u et 2. les variables dépendantes des premières q_v .

$$q = \begin{pmatrix} q_u \\ q_v \end{pmatrix} \quad (1.5)$$

Un premier sous-module *MBSysC* sélectionne *adéquatement* ($N_{coord} - N_{contr}$) q_u où ($N_{coord} - N_{contr}$) correspond au nombre de degrés de liberté N_{ddl} du système multicorps. Les détails de cette partition sont expliqués en [8].

Réduction

A l'aide du partitionnement, les termes de (1.1) peuvent être réécrits selon (1.5) :

$$\begin{pmatrix} M_{uu} & M_{uv} \\ M_{vu} & M_{vv} \end{pmatrix} \begin{pmatrix} \ddot{q}_u \\ \ddot{q}_v \end{pmatrix} + \begin{pmatrix} c_u \\ c_v \end{pmatrix} = \begin{pmatrix} Q_u \\ Q_v \end{pmatrix} + \begin{pmatrix} J_u^t \\ J_v^t \end{pmatrix} \lambda \quad (1.6)$$

où la deuxième ligne du système peut être remplacée dans la première via l'inconnu λ :

$$M_{uu}\ddot{q}_u + M_{uv}\ddot{q}_v + B_{vu}^t(M_{vu}\ddot{q}_u + M_{vv}\ddot{q}_v) + c_u + B_{vu}^t c_v = Q_u + B_{vu}^t Q_v \quad (1.7)$$

où $B_{vu} \triangleq -J_v^{-1} J_u$.

Les contraintes (1.2-1.4) sont ensuite résolues :

- Contrainte en position : $h(q) = 0$ est non-linéaire et est résolue via la méthode numérique itérative de Newton-Raphson. A partir de conditions initiales q_u indépendantes et connues, Newton-Raphson converge vers les q_v assurant $h(q) = h(v) = 0$. La jacobienne $J = \frac{\partial h}{\partial q^t}$ et la matrice de masse peuvent alors être calculées.
- Contrainte en vitesse : $\dot{h}(q, \dot{q}) = J(q)\dot{q} = 0$ est linéaire. Connaissant les \dot{q}_u (= conditions initiales) et J , les \dot{q}_v se calculent directement comme : $\dot{q}_v = B_{vu}\dot{q}_u$.
- Contrainte en accélération : $\ddot{h}(q, \dot{q}, \ddot{q}) = J(q)\ddot{q} + \dot{J}\dot{q}(q, \dot{q}) = 0$ est aussi linéaire et les \ddot{q}_v se déduisent comme : $\ddot{q}_v = -B_{vu}\ddot{u} - J_v^{-1}\dot{J}\dot{q}$.

Grâce à ces fermetures en position, vitesse et accélération, tous les termes de l'équation 1.7 ont été parcourus à l'exception de c et Q .

Comme mentionné dans la table, Q modélise les forces/couples de joints. C'est l'utilisateur qui les implémente.

L'utilisateur peut aussi appliquer des forces extérieures sur le mécanisme étudié. Celles-ci forment en partie le terme c et sont illustrées avec l'exemple de la section suivante.

Dès lors, les équations du mouvement avec les seuls \ddot{q}_u comme inconnues s'écrivent sous la forme suivante :

$$(M_{uu} + M_{uv}B_{vu} + B_{vu}^t M_{vu} + B_{vu}^t M_{vv} B_{vu})\ddot{q}_u + (M_{uv} + B_{vu}^t M_{vv})b + (c_u + B_{vu}^t c_v) - (Q_u + B_{vu} Q_v) = 0 \quad (1.8)$$

où $b = -J_v^{-1}(\dot{J}\dot{q})$ a été défini. Sous une forme condensée, le système d'équation s'écrit :

$$M_r(u)\ddot{q}_u + F_r(\dot{u}, u) = 0 \quad (1.9)$$

1.2 Véhicule étudié

Cette section décrit un exemple d'utilisation du logiciel *Robotran*. Elle présente un modèle de kart, réalisé pour le projet du cours de dynamique des véhicule [7]. Le projet étudie l'influence des paramètres géométriques (chasse, déport,...) du train avant sur le retour de couple au volant. Les résultats de cette étude sont repris dans le rapport [10].

1.2.1 Modélisation

Comme expliqué en section 1.1.1, la première étape dans *Robotran* est de définir le mécanisme multicorps sur *MBSysPad*. A l'issue de cette étape, les dimensions, masses, centres de masse et inerties du véhicule sont définis.

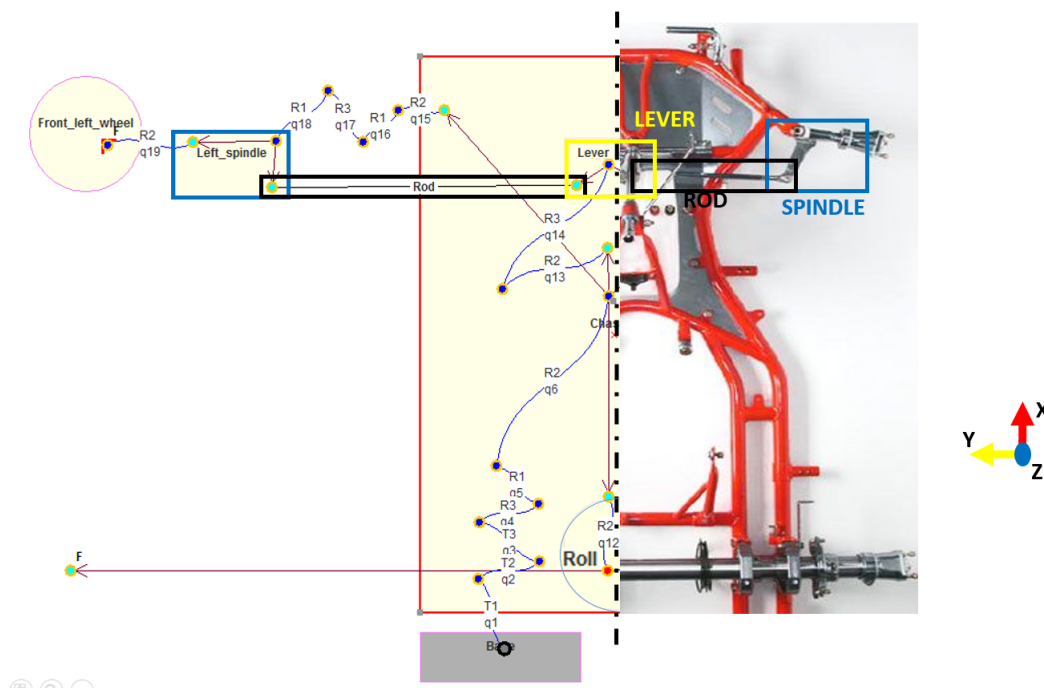


FIGURE 1.3 – Comparaison entre le modèle et un châssis réel [15]

1.2.1.1 Géométrie des corps

La figure 1.3 montre les corps du kart implémenté (symétrique). Un code couleur permet de le comparer aisément avec un châssis réel.

Remarquons que la figure ne donne aucune indication sur les coordonnées réelles de chaque point. Elle est même trompeuse à cet égard car le modèle a été réalisé tel qu'il minimise le nombre de coordonnées à encoder.

Ci-dessous, les mesures principales des corps du kart sont reprises. Certains des paramètres ont été mesurés sur un kart et d'autres, plus difficilement mesurables, proviennent d'une fiche d'homologation [16] ou d'un dessin technique [17] (aucun fabricant ne fournit le plan d'un kart complet).

- **Le châssis :**

Voie avant : $1.06[m]^3$ - Voie arrière : $1.15[m]$ - Empattement : $1.045 [m]$

3. mesuré à partir du milieu des pneus.

- **Les fusées** ("Spindle")
Longueur de la partie latérale de la fusée : 0.18 [m]
Longueur de la partie longitudinale de la fusée : 0.1 [m] (grandeurs principales dans le plan x,y)
- **Le triangle de direction**⁴ ("Lever")
Longueur, selon x : 0.015 [m]
Largeur, selon y : 0.05 [m] (grandeurs principales dans le plan x,y)
- **Les roues avant**
Rayon : 0.127 [m] - Bande de roulement : 0.117 [m]
- **Les roues arrière**
Rayon : 0.139 [m] - Bande de roulement : 0.180 [m]

La biellette de direction (liant fusée et triangle de direction : en noir sur la figure 1.3) n'est pas modélisée par un corps.

En effet, pour que *Robotran* puisse formuler les équations du mouvement 1.1 (en utilisant Newton-Euler récursif en coordonnées relative), un formalisme "tree-like" est indispensable. Il permet d'exprimer la position/orientation absolue de façon unique pour chacun des corps. Or, sur le kart réel, la biellette forme une boucle cinématique.

La biellette est donc remplacée par une contrainte de type "rod" qui impose une distance fixe entre la fusée et le triangle de direction. Cette distance garantit un pincage nul ($\alpha = 0$ [°]) et la bielle est supposée sans masse. Le développement exhaustif de cette justification se trouve dans le livre de référence [1].

La particularité du corps "**Roll**" de la figure 1.3 sera expliquée en section 1.2.1.3

Hypothèse

Outre l'hypothèse mentionnée juste ci-dessus et d'autres comme la nullité du jeu, du frottement dans les articulations, l'approximation à des formes simples,... l'hypothèse principale du modèle est la rigidité de l'arbre de transmission et celle du châssis.

1.2.1.2 Masse et Inertie

Le kart (pilote + kart) pèse 150 kg, à savoir 5 kg de plus que le minimum autorisé par la CIK, Commission Internationale de Karting [27]. Les inerties ont été approximées en ramenant les corps complexes (voir figure 1.3) à des formes simples : parallélépipèdes et cylindres.

Elles sont donc aisément calculées en appliquant les formules des matrices d'inertie pour un corps rigide. Le détail de ces calculs est repris en [10].

1.2.1.3 Articulation entre les corps

Nous nous référons ici à la numérotation telle qu'exposée en figure 1.3. "1" est la direction selon X, "2" selon Y et "3" selon Z.

Le **châssis** à 6 degrés de liberté par rapport à la base ($q_1 \rightarrow q_6$ sur la figure).

Les **roues arrière** d'un kart sont solidaires via un axe. Cette contrainte a été modélisée par un corps "Roll" ayant un degré de liberté R_2 par rapport au châssis.

A la place d'avoir deux corps "roues" liés par deux articulations R_2 à l'arbre, un unique corps modélise l'ensemble arbre de transmission et roues arrière. Cela s'assimile à un rouleau de diamètre égal à celui des roues arrière, s'étendant des deux côtés du plan de symétrie XZ du châssis. Cette modélisation est plus économe en temps de calcul car :

4. qui modélise toute la colonne de direction.

- Il y a deux fois moins de corps/articulations.
- Il y a une contrainte en moins à appliquer (position, vitesse et accélération égale des deux roues).

La **fusée gauche** est liée par 4 joints au châssis mais n'a qu'un seul degré de liberté par rapport à celui-ci.

3 rotations sont donc forcées (variables commandées forcées, équation 1.11) :

- *R2 q15* : fixe l'angle de chasse⁵ de l'axe du pivot, constant pour une configuration donnée. Pour le kart étudié, cet angle vaut 10 [°] selon les standards, par exemple en [6].
- *R1 q16* : fixe l'angle d'inclinaison de l'axe pivot, constant pour une configuration donnée. Cet angle vaut aussi 10 [°].
- *R1 q18* : réaligne Y_{local} à $Y_{inertiel}$. Cela permet de n'avoir qu'une coordonnée à encoder (selon Y) pour le point extérieur de la fusée (la "*partie latérale*"). Cette seule coordonnée garantit un carrossage nul $\gamma = 0$ [°] car le Y local est aligné avec l'Y inertiel et la roue est perpendiculaire à la fusée. Il vaut donc -10 [°].

Deux rotations servent donc à orienter l'axe de pivot de la roue, l'axe autour duquel la roue tourne lors d'un braquage.

R3 q17 est donc le seul degré de liberté local (braquage des roues).

Le **triangle de direction** est relié par deux joints au châssis. Le premier *R2* est de type commandé forcé. Il oriente la colonne de direction à 45[°] vers le pilote.

La deuxième articulation, *R3* est commandée. Cette articulation est la rotation du volant et revêt une importance capitale pour le simulateur.

Pour conclure cette présentation du modèle, le nombre de degrés de liberté (en incluant l'utilisateur, en considérant qu'il a le droit de bouger le volant) est calculé :

$$\begin{aligned} ddl &= ddl_{chassis} + ddl_{roues} + ddl_{fusées} + ddl_{volant} - \text{contrainte}_{rod} \\ &= 6 + 3 + 1 \times 2 + 1 - 2 = 10 \end{aligned}$$

En figure 1.3, des capteurs de forces sont aussi placés sur les roues. Cela fait l'objet de la section suivante.

1.2.2 Implémentation des forces

Après avoir généré les équations du mouvement du kart, c'est dans le module *MBSysC* de *Robotran* que l'équation du mouvement est résolue. L'utilisateur doit quant à lui implémenter les forces qui agissent sur le modèle.

1.2.2.1 Les forces de joint

Comme décrit au chapitre suivant, le simulateur de véhicule n'a pas de pédalier. Toutes les simulations proposées dans ce travail sont réalisées à vitesse constante grâce à un régulateur de vitesse PID qui impose un couple à l'articulation *R2 q12* du rouleau.

Ces paramètres K_p , K_i , K_d sont réglés expérimentalement de façon à avoir un temps de réponse proche de celui qui existe en kart de compétition (Kart à vitesse : $0 \rightarrow 100$ [km/h] ≈ 3 [s], Kart classique : $0 \rightarrow 100$ [km/h] $\approx 5 - 6$ [s] selon le modèle).

5. Les définitions des paramètres caractéristiques sont repris dans le travail [10]

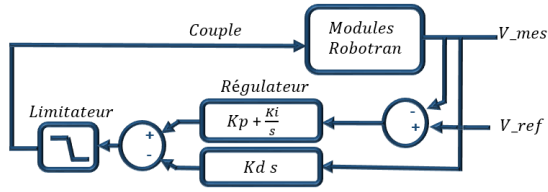
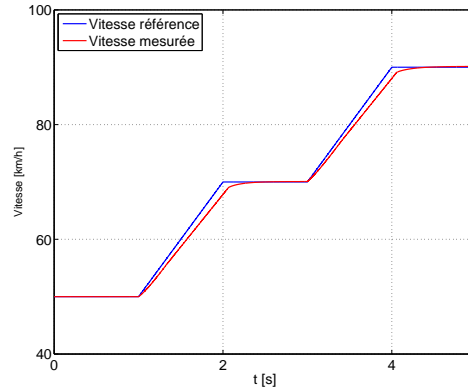


FIGURE 1.4 – Schéma bloc du régulateur

FIGURE 1.5 – Réponse de la vitesse pour des steps de 20 [km/h] sur 1 [s] en $t = 1$ [s] et $t = 3$ [s]

La vitesse linéaire du kart suit correctement les consignes sur la figure 1.5. Le temps de montée et d'établissement est faible, il y a peu de dépassement et l'erreur statique s'annule.

1.2.2.2 Les forces extérieures

Les forces extérieures se retrouvent dans le terme "c" de l'équation du mouvement 1.1. La gravité et les forces de contact pneu/sol sont les deux types de forces extérieures qui agissent sur le kart.

La gravité est directement implémentée sur *MBSysPad*.

Le pneu subit 3 forces et 3 moments extérieurs imposés par le sol :

- Force longitudinale (X)
- Force latérale (Y)
- Force normale (Z)
- Moment de renversement (X)
- Moment de résistance au roulement (Y)
- Moment d'autoalignement (Z)

Vulgairement, la force normale F_z empêche le véhicule de s'enfoncer dans le sol, la force longitudinale F_x accélère ou freine le véhicule et la force latérale F_y garde le véhicule sur sa trajectoire lors de la prise de virage.

Dans *Robotran*, l'ellipse de contact pneu/sol est remplacée par un point. Or, toutes ces 3 forces n'agissent pas nécessairement au point central de cette ellipse (bras de levier $\neq 0$). Les moments permettent donc (entre autres) de prendre en compte la surface de contact.

Calcul des forces

La force normale est modélisée comme un ressort-amortisseur où ce dernier est le caoutchouc du pneu :

$$F_z = K \cdot e - D \cdot V \quad (1.14)$$

e est la pénétration du pneu dans le sol et V la vitesse de rapprochement de la roue vers le sol.

K est la constante de raideur et D le coefficient d'amortissement. Leur valeur a un impact sur le couple transmis au volant et sera discuté en section 3.3.3.

Le restant des forces et moments de contact sont calculés à l'aide du modèle de pneu de "**Bakker**". Ce modèle semi-empirique utilise la cinématique et la force normale F_z de chacune des roues pour calculer F_y, F_x, M_z (hypothèse : $M_y \approx 0$ et $M_x \approx 0$, F_z agit au centre de l'ellipse de contact).

Relaxation

Le dernier élément à prendre en compte est le temps de réponse des forces.

Jusqu'ici aucune notion temporelle n'apparaît explicitement dans le modèle de Bakker. Or, lors de virages à haute dynamique, les forces mettent un certain temps à atteindre leur valeur de "régime" (telle que prédite par "Bakker" à l'instant précis).

Des étudiants de l'EPL dont le mémoire portait sur l'étude de la dynamique d'un véhicule à échelle réduite [13], ont étudié ce phénomène. Ils le modélisent via une équation différentielle ordinaire :

$$\frac{d\beta}{dt} = \frac{V}{\sigma}(\beta^* - \beta) \quad (1.15)$$

où :

- $\beta^* = F_{long,bakker}$ ou $F_{lat,bakker}$ ou $M_{z,bakker}$, la valeur de référence à atteindre prédite par Bakker.
- $\beta = F_{long}$ ou F_{lat} ou M_z , la valeur actuelle corrigée.
- V la vitesse linéaire du kart. σ la longueur de relaxation du pneu. Plus V/σ est élevé, plus le modèle se rapproche d'"un modèle stationnaire" selon la solution analytique de cette équation : $\beta(t) = \beta^*(1 - e^{-\frac{V}{\sigma}t})$.

L'équation 1.15 doit donc être intégrée à chaque pas de temps.

1.3 Conclusion

Cette section a décrit le fonctionnement d'un outil de simulation dynamique. L'utilisateur définit un véhicule et les forces qui agissent sur ce dernier. A partir de ces modèles, l'outil simule toute la dynamique du véhicule et se place comme élément indispensable du simulateur de l'UCL présenté au chapitre suivant.

Chapitre 2

Simulateur de l'UCL

Le simulateur de l'UCL est un outil pédagogique conçu par *Nicola de Lellis* et *Quentin Marterne*. Le but premier du simulateur de véhicule est de proposer à son utilisateur une expérience de conduite réaliste grâce à un retour haptique issu des lois de la physique du véhicule. Avant de détailler le simulateur, il convient de définir le terme *haptique*.

2.1 Définition de l'haptique

L'haptique désigne la science du toucher [18]. Un mouvement haptique est un mouvement où la couche superficielle de la peau (perception tactile) mais aussi les couches inférieures comme les muscles, les ligaments et les articulations (perception kinesthésique) sont excitées.

La sensibilité des fréquences haptiques n'est pas facilement identifiable. La référence [19] a étudié le sujet en détail. Selon celle-ci, des stimuli tactiles sont perceptibles jusqu'à des fréquences de $0 \rightarrow 400$ [Hz] mais peuvent aller jusqu'à 10 [kHz] pour des textures très fines. La bande passante des récepteurs kinesthésiques est quant à elle limitée à des fréquences de $10 \rightarrow 30$ [Hz].

Pour le simulateur de véhicule, l'haptique est donc utilisée, par l'intermédiaire d'un volant, pour recréer les sensations de conduite d'un véhicule simulé sur un écran. Ainsi l'influence d'un paramètre du véhicule sur sa tenue de route, l'interaction entre l'homme et le véhicule peuvent par exemple être étudiées.

Précision faite, la suite de ce chapitre décrit le simulateur tel que conçu par les étudiants. Il décrit chacun des composants sans expliciter les détails de la conception. Les justifications plus complètes sont reprises dans le travail [11].

2.2 Vue globale du dispositif

En parallèle à un état de l'art complet [12], les étudiants ont réalisé un cahier des charges reprenant les fonctions et contraintes que le simulateur de véhicule doit supporter. Au compromis entre les performances désirées et les contraintes pratiques du projet, la conception électromécanique puis électronique du simulateur a été réalisée. Un premier schéma illustre la morphologie du simulateur :

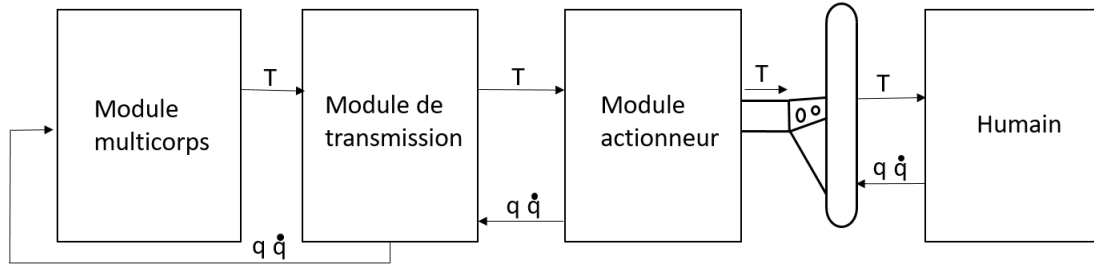


FIGURE 2.1 – Illustration des 5 composants principaux du simulateur de véhicule

Cinq blocs clefs se retrouvent sur ce schéma :

1. *Le module multicorps*
2. *Le module de transmission*
3. *Le module actionneur*
4. *Le volant*
5. *L'humain*

Au vu de la figure 2.1, le simulateur est de type "Human-In-The-Loop" : *le module multicorps* modélise un véhicule complet : de la roue à la colonne de direction.

Le simulateur fonctionne en dynamique inverse :

Le module multicorps calcule le couple à envoyer au *volant* grâce aux positions et vitesses de ce dernier. Ce couple transite par le *module de transmission* qui le cède au *module actionneur*. Ce module s'occupe de le mettre en consigne au *volant*.

En bout de chaîne, *l'humain* tenant le volant entre ses mains, reçoit le couple. *Le volant* est donc **l'interface haptique** entre la dynamique du véhicule passant par les modules mécatroniques et *l'humain*.

La figure peut aussi se parcourir en sens inverse, l'ensemble formant une boucle : *l'humain* impose une certaine position et vitesse au *volant*. Ces informations sont récupérées par le *module actionneur* pour faire le chemin inverse à la consigne de couple et servir d'entrée au *module multicorps*,

Remarquons que l'accélération n'est pas prise en compte dans ce modèle.

Premièrement, la suite de ce chapitre détaille chacun des modules : elle explique leur rôle et sur quel matériel ils sont implémentés (excepté pour *l'humain*). La section suivante met en évidence les communications entre les modules, deux à deux. Ensuite, la boucle de contrôle du simulateur, rassemblant tous les modules et communication, est détaillée. La dernière section établit un bilan du fonctionnement du simulateur et met en évidence les problèmes subsistants qui détériorent le retour haptique.

2.3 Modules du simulateur

2.3.1 Module multicorps

Conceptuellement, *le module multicorps* est le centre de calcul de la dynamique du véhicule simulé. Cette section décrit comment *le module multicorps* calcule le couple à envoyer à l'actionneur à partir des positions et vitesses du volant. Elle se termine en expliquant le concept du temps réel indispensable au simulateur.

Calcul du couple

Robotran est le logiciel que *le module multicorps* représente. Il est utilisé sur un ordinateur sous

Windows.

La section 1.1.3 a défini la notion de variable commandée. L'articulation du volant (voir figure 1.3 : le degré de liberté de rotation entre la colonne de direction et le châssis) est une variable commandée. En effet, *l'Humain* lui impose une position, une vitesse et une accélération (non prise en compte) à chaque instant. Ces coordonnées commandées, tout comme les indépendantes, servent de point de départ pour le premier partitionnement et la réduction du système d'équations du mouvement. Comme mentionné, elles doivent ensuite être séparés pour intégrer uniquement les variables indépendantes.

Ce deuxième partitionnement est d'un intérêt particulier pour le simulateur car, comme la référence [1] le démontre, le couple des variables commandées est contenu dans le λ_c du partitionnement (équation 1.13). Dès lors, après le calcul des \ddot{q}_u , le couple de l'articulation du volant est directement obtenu par :

$$\lambda_c = M_{rcu} \ddot{q}_{uu} + M_{rcc} \ddot{q}_{uc} + F_{rc} \quad (2.1)$$

Ce couple est donc envoyé à l'actionneur. Toutes les étapes décrites pour le calculer, à un instant donné, sont appelées *dynamique directe* (ou DIRDYN) pour la suite du travail.

Temps réel

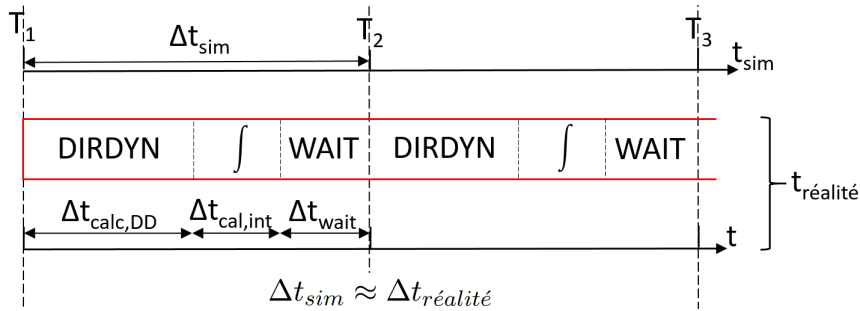


FIGURE 2.2 – Illustration du temps réel

Le temps réellement pris par l'ordinateur pour calculer le couple T est donné par :

$$\Delta t_{réalité} = \Delta t_{calc,DD} + \Delta t_{calc,int} + \Delta t_{wait} \quad (2.2)$$

où

- $\Delta t_{calc,DD}$ définit le temps pris par l'ordinateur pour effectuer la dynamique directe.
- $\Delta t_{calc,int}$ définit le temps pris par l'ordinateur pour intégrer les \dot{q}_u, \ddot{q}_u ($\Delta t_{calc,int} \ll \Delta t_{calc,DD}$).
- Δt_{wait} définit un temps d'attente de l'ordinateur.

et Δt_{sim} définit le pas de temps d'intégration de simulation, c'est-à-dire le Δt_{sim} plus "loin" en unité de temps de simulation où le couple est calculé.

Définissons le facteur temps réel v_r :

$$v_r = \frac{\Delta t_{sim}}{\Delta t_{calc,DD} + \Delta t_{calc,int}} \quad (2.3)$$

Trois cas sont possibles :

1. $v_r > 1$: le temps réel est respecté. $\Delta t_{calc,DD} + \Delta t_{calc,int} < \Delta t_{sim}$ et $\Delta t_{wait} > 0$ afin d'avoir $\Delta t_{réalité} \approx \Delta t_{sim}$.
2. $v_r = 1$: le temps réel est respecté. Ce cas désigne la limite du temps réel : $\Delta t_{wait} = 0$.
3. $v_r < 1$: le temps réel n'est pas respecté. $\Delta t_{calc,DD} + \Delta t_{calc,int} > \Delta t_{sim}$ et $\Delta t_{wait} = 0$.

Pour trouver le Δt_{sim} assurant le temps réel, un compromis existe :

D'un côté réduire un maximum le $\Delta t_{sim} = t_{i+1} - t_i$ permet d'obtenir des résultats (T, q, \dots) plus précis, se rapprochant d'une intégration continue et favorisant la stabilité numérique (fermeture des boucles et stabilité de l'intégrateur).

D'un autre côté, réduire exagérément¹ le Δt_{sim} , réduirait le facteur temps réel v_r car le temps $\Delta t_{calc,DD} + \Delta t_{calc,int} \approx cste$ pour un modèle multicorps donné.

Pour résumer : Nous avons parcouru la composition du *module multicorps* : un ordinateur hébergeant une session *Robotran*. Nous avons explicité son fonctionnement : le module calcule, à chaque pas de temps, en temps réel, le couple à envoyer au volant T grâce à la position et vitesse du volant.

2.3.2 Module actionneur

Le *module actionneur* est composé de trois entités : un servomoteur, un codeur et un servovariateur. Ils servent donc à :

- transmettre le couple au *Volant*.
- récupérer la position et la vitesse du *Volant*.

Cette section a pour but de présenter les caractéristiques et les fonctions du matériel choisi, ayant la capacité de produire un rendu haptique de qualité.

Le servomoteur

L'actionneur choisi est en *prise directe* : aucun élément de transmission ne sépare l'arbre moteur du volant. Les critères essentiels, en plus du cahier des charges, qui ont amenés les étudiants à sélectionner cette morphologie d'actionneurs (plutôt qu'un actionneur linéaire par exemple) sont :

- *Contraintes pratiques*
 - Masse/ Longueur : le simulateur doit être facilement transportable.
 - Délai de livraison : l'actionneur devait être livrable rapidement.
- *Contraintes mécaniques*
 - Jeu : l'actionneur doit satisfaire à un jeu quasiment nul. Ce facteur est important pour qu'il y ait une faible différence entre la commande de l'utilisateur et ce qui est imposé au module multicorps. C'est aussi un facteur important au vu de l'entretien que pourraient nécessiter des engrenages, par exemple.
 - Inversibilité : le couple minimum à imposer à l'arbre moteur de sortie pour tourner l'actionneur (éteint) doit être le plus faible possible.

Le servomoteur retenu est le *BMH140P12A2A* de chez *Schneider Electric* [28]. Il a été choisi, parmi d'autres servomoteurs directs adaptés, pour son ratio performance/prix (figure 2.3).

1. ce que "exagérément" signifie dépend du modèle, de l'ordinateur utilisé,...

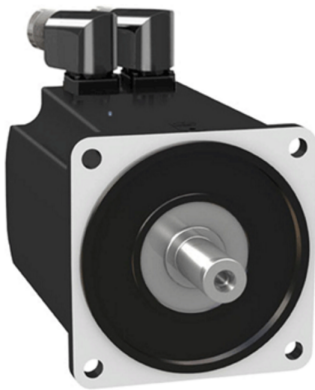


FIGURE 2.3 – Servomoteur du simulateur [28]

| Caractéristiques Générales | |
|----------------------------|------------------------------|
| Type du moteur | Synchrone 5PP |
| Alimentation | 240 [V] 3 ϕ , max 3 [A] |
| Dimension IXL | 140 X 202 [mm] |
| Masse | 8 [kg] |

| Performances | |
|--------------------|----------------------------|
| Couple de maintien | 10.3 [Nm] |
| Couple nom & max | 8.3 & 30.8 [Nm] |
| Vitesse nom & max | 2000 & 4000 [rpm] |
| Puissance nominale | 1.74 [kW] |
| Inertie du rotor | 16.46 [kgcm ²] |

TABLE 2.1 – Caractéristiques du le servo-moteur

Le codeur

Le codeur intégré au moteur est le *SinCos Hiperface* [28]. De type absolu, il mesure une position absolue au démarrage et calcule la position courante de façon incrémentale par rapport à cette position de référence.

Le codeur choisi est aussi multi-tours, cependant cette faculté n'est pas utilisée pour ce travail.

| Nom | Résolution par révolution | Nombre de tours max | Précision |
|-----------------|---------------------------|---------------------|-------------------|
| SKM36 Multiturn | 128 périodes sin/cos | 4096 | $\pm 0.022^\circ$ |

TABLE 2.2 – Caractéristiques du codeur *SinCos Hiperface*

Le servo-variateur

Le matériel retenu, car compatible avec le moteur, est le *Lexium LXM32ADM2*. Le servo-variateur a deux fonction principales.

D'une part, c'est l'organe qui pilote en temps-réel le moteur. *Le module de transmission* lui transmet un couple 2.1 qui sert de consigne au moteur lui-même solidaire au *volant*.

D'autre part, il récupère la position angulaire du moteur et calcule la vitesse angulaire de celui-ci. Si une requête lui est envoyée, il transfère ces informations au *module de transmission*.

FIGURE 2.4 – Servo-variateur *Lexium LXM32ADM2*

| Nom | Alimentation | Intervalle de couple | Précision du couple |
|------------------|--------------|----------------------|---------------------|
| Lexium LXM32ADM2 | 220 V AC | -24.9→24.9 [Nm] | 0.1% |

TABLE 2.3 – Caractéristiques du servo-variateur

Assemblage mécanique

Pour conclure cette section, l'assemblage mécanique du simulateur 2.5 est présenté. L'ensemble

servomoteur (moteur 5 + codeur) est disposé dans un cadre 3 reposant sur un pied 1. Ces deux derniers assurent la sécurité de l'utilisateur.

Autour de l'arbre de sortie du moteur, un moyeu est fixé 5.2. Le mouvement relatif de rotation est empêché par une clavette 5.1 de l'arbre moteur. Une rondelle et une vis, s'insérant dans l'alésage taraudé coaxial de l'arbre moteur, empêchent quant à eux le mouvement relatif axial. Le volant 6 est solidarisé au moyeu par des vis s'insérant dans les alésages taraudés de la face libre du moyeu.

Le volant est inclinable d'un angle de 15° à un angle de 30° . Un système de glissière en arc de cercle (voir figure 2.6) autorise le simulateur à tourner autour d'un pivot 4.

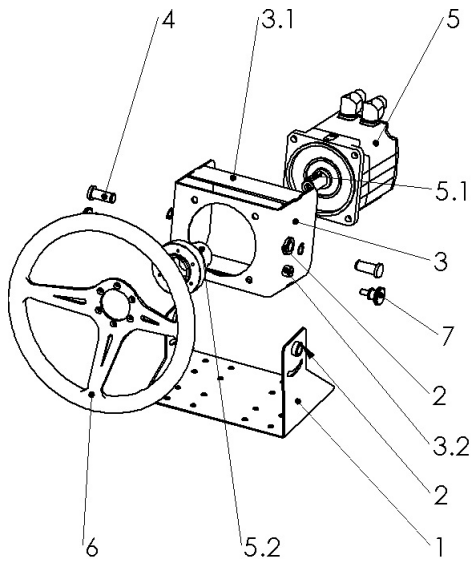


FIGURE 2.5 – Vue éclatée de l'assemblage mécanique sous *SolidWorks* [11]

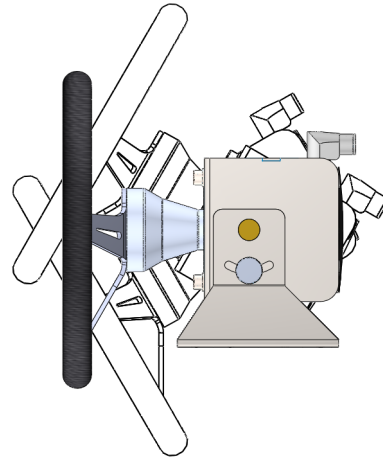


FIGURE 2.6 – Simulateur incliné de 15° à 30°

Les caractéristiques présentées pour le servomoteur, le codeur et le servo-variateur concordent avec le cahier de charges dressé par les auteurs et autorisent donc une simulation à rendu haptique réaliste.

2.3.3 Module de transmission

Le module de transmission prend son nom du rôle qu'il détient : il sert de passerelle entre le module *multicorps* et le module *actionneur*. La figure 2.7 ci-contre illustre les enjeux du module. Le module y est représenté comme un agent de circulation sur un carrefour quatre bras (les entrées et sorties du module de la figure 2.1). Il a deux flux d'informations à gérer (le Couple T et la paire q, \dot{q} représentés par les voitures) et décide quand ils peuvent passer de ses entrées à ses sorties. Le détail de cette gestion de transit d'informations est dépendant de l'architecture de contrôle et sera explicité plus loin, au cours de la section 2.5.

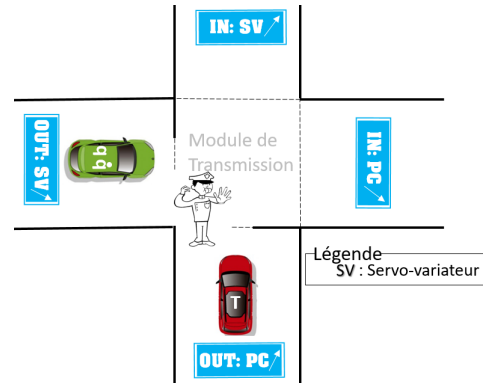


FIGURE 2.7 – Analogie du rôle du module de transmission

Pour que le transit d'information soit fluide, un hardware suffisamment puissant utilisant les protocoles de communications adaptés au *module multicorps* et *module actionneur* est indispensable.

A cette fin, une liste de critères de sélection a été dressée. Finalement, la carte de développement de *Texas Instruments* de type *Industrial Communications Engine AM3359*² a été retenue. Les composants d'intérêts sont mis en évidence sur la figure 2.8

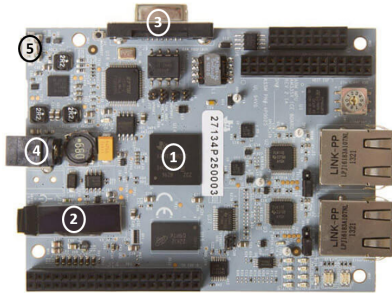


FIGURE 2.8 – Photo de la carte [29]

| Numéro | Composants |
|--------|------------------------|
| 1 | Microprocesseur AM3359 |
| 2 | Écran LCD |
| 3 | Alimentations 24 V |
| 4 | DB9 pour CanOpen |
| 5 | Micro USB pour UART |

TABLE 2.4 – Légende de la figure 2.8

Le cerveau de la carte de développement est le microprocesseur AM3359. Il contient lui-même le microprocesseur ARM Cortex-A8 ainsi qu'une série de périphéries, interfaces, mémoires,...

Comme l'indique la figure 2.8, la carte offre bien plus de ressources que son utilisation actuelle. La carte supporte les communications via Ethernet, EtherCat, mais contient aussi des LEDS, des interrupteurs, différentes mémoires,...(fonctionnalité complète en [21]).

Dans le cas présent, seulement deux protocoles de communication sont utilisés :

- L'UART pour la communication avec le PC (*module multicorps*).
- Le CanOpen pour la communication avec le servo-variateur (*module actionneur*).

Ils font l'objet de la section suivante.

2.4 Protocoles de communication

2.4.1 Communication série

La communication série fait le lien entre le *module multicorps* et le *module de transmission*. C'est

2. appelé "carte de développement" ou "carte Texas" ou encore "carte" par la suite.

le type de canal entre l'ordinateur hôte et la carte Texas.

La communication série est un type de communication où les données sont envoyées bit par bit sur une seule voie. L'UART est le composant électronique clef de la communication série. Sur le PC, il transforme une séquence de bytes en bits pour les envoyer sur le câble série. Vice-versa, sur la carte Texas, l'UART rassemble les bits en bytes pour être utilisés par le microprocesseur.

La communication UART utilisée est asynchrone. Ses paramètres de communication sont :

- Clock Baudrate : 230 400 [bps]
- Start bit : 1
- Nombre de bits par trame : 8
- Stop bit : 1
- Parity bit : 0

Aucun signal de "clock" n'est échangé. Les deux parties communicantes utilisent chacune une horloge interne à la même fréquence (le baudrate) pour pouvoir rassembler sans erreurs les bits de la trame reçue.

Le début du message est repéré par la partie réceptrice grâce au premier bit de la trame, le "Start bit".

Le "Stop bit" indique lui la fin de la trame et sert de protection pour les éventuelles erreurs de communication.

La vérification de la validité de la trame est également assurée par le "Parity bit". Il n'est toutefois pas utilisé pour ce travail.

La communication est aussi de type Full-Duplex : le câble "aller" transmettant les bits de couple et le câble "retour" transmettant les bits de position/vitesse, sont utilisables simultanément.

Concernant l'implémentation physique, c'est le standard RS-232 qui est utilisé. Il est semblable à l'UART TTL classique (une entrée RX et une sortie TX seulement) avec quelques extensions expliquées en [22].

Notons aussi la présence d'un driver pour émuler le port COM.

En effet, le standard RS-232 utilise un port série (port COM). Or, sur les PC modernes, ceux-ci ont été largement remplacés par des les ports USB. Le *Virtual Comp Port Drivers* de FTDI fait apparaître un port USB comme si c'était un port COM. Dès lors, l'accès au port USB se fait comme l'accès à un port COM et le driver s'occupe des conversions COM→USB (plus de détails en [23]).

2.4.2 Communication CANopen

La communication CANopen fait le lien entre *le module de transmission* et *le module actionneur*.

Le CANopen est une couche applicative du bus terrain CAN permettant la communication en temps réel.

Il utilise des objets comme moyen de communication entre des dispositifs différents, appelés nœuds, montés en parallèle sur un même bus CAN.

Le CANopen implémente un dictionnaire d'objets pour chaque nœud. Il sert d'interface entre l'application (le servo-variateur) et le bus (voir figure 2.9). Le dictionnaire d'objets contient un ensemble d'objets, chacun accédé par un index et sous-index qui définit les champs de l'objet en question.

Ces objets sont classés en quatre grandes catégories. Les objets *Communication Profile Area*

requièrent particulièrement notre attention et sont accédés par un protocole de communication dépendant du type de l'objet.

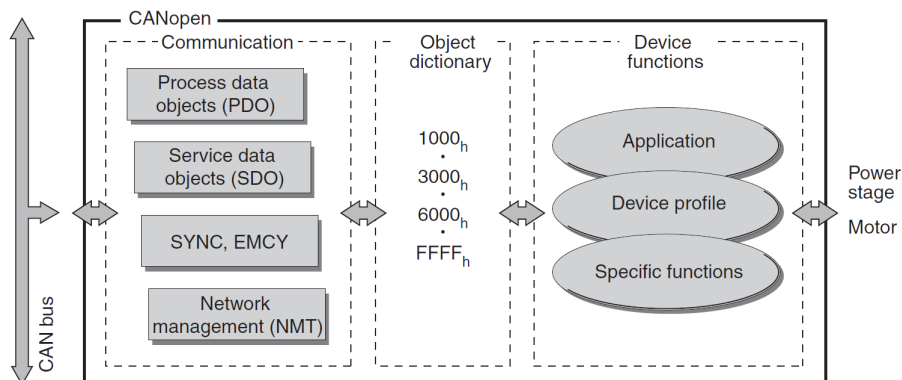


FIGURE 2.9 – Illustration du CANopen [30]

Les Process Data Object (PDO) permettent le transfert de données en temps réel. Ils s'organisent sous la forme d'un producteur qui envoie les données à un consommateur qui les reçoit et si besoin, répond à la requête du producteur. La transmission est rapide car seules les données porteuses de l'information (le couple par exemple) sont envoyées. Les PDO nécessitent donc une configuration préalable pour pouvoir adresser les champs voulus du dictionnaire. Notons encore que les PDO ne nécessitent pas de confirmation de bonne réception par l'application. Les Service Data Object (SDO) sont utilisés pour accéder à une zone du dictionnaire d'objets définie par son index et son sous-index. Ces index sont directement présents dans la trame contrairement aux PDO. Le SDO généralement utilisé pour la configuration d'une application et suit la relation d'un client/serveur. Le client envoie un message CAN au serveur qui le traite et lui répond avec un message de confirmation. Les Network Management (NMT) sont utilisés pour la gestion du réseau et définissent quatre états en fonction desquels certaines modifications des configurations PDO ne sont pas possibles.

Dans le cas du simulateur, après avoir créé et configuré le NMT du réseau, le module de transmission envoie des requêtes PDO pour recevoir q, \dot{q} . Le module envoie également le couple sous forme d'une communication PDO dans le dictionnaire d'objets. Une communication retour pour préparer l'envoi suivant est aussi réceptionnée.

2.5 Architecture de contrôle

A ce stade, en reprenant la figure 2.1 :

1. Tous les modules (les blocs)
2. Tous les types de communication entre les modules (les flèches)

ont été définis au cours des sous-sections 2.3 et 2.4.

Dès lors, la section explique comment les modules interagissent entre eux grâce aux protocoles de communication. C'est ce que nous appelons **l'architecture de contrôle du simulateur**. L'architecture de contrôle définit, de manière symbolique, la disposition et les interactions entre des modules et sous-modules informatiques.

Pour les présentations d'architectures de contrôle, le principe d'abstraction sera abondamment utilisé [3]. Bien qu'élémentaire, sa signification est rappelée.

Le principe d'abstraction

Afin de réduire la complexité, tous les détails des sous-systèmes qui ne sont pas importants pour la compréhension générale du système étudié ne sont explicités. Il est possible de définir plusieurs niveaux d'abstraction et ainsi comprendre des systèmes d'une grande complexité.

L'exemple le plus classique est la compréhension d'un processeur d'un ordinateur. Ne pas se soucier de comment se déplace un électron n'empêche en rien de comprendre comment fonctionne le "Datapath".

Pour le simulateur, dissimuler le fonctionnement du partitionnement ou les détails de communication n'empêche pas de comprendre comment le couple est transmis de module en module jusqu'au volant.

Extension du principe, nous utilisons aussi fréquemment la personnification. En voici un exemple simple : "La fonction 3 du programme 4 de la carte Texas envoie le couple" se simplifie en "la carte envoie le couple".

L'architecture implémentée par les étudiants du mémoire précédent est une architecture dite sérielle (figure 2.10).

Une notation des modules plus fonctionnelle est introduite. L'ensemble "volant + servomoteur + codeur" a été rassemblé en un élément, le volant. Le servo-variateur fait lui l'objet d'un bloc entier. Lorsqu'un des 4 blocs est énoncé, le format bloc est utilisé.

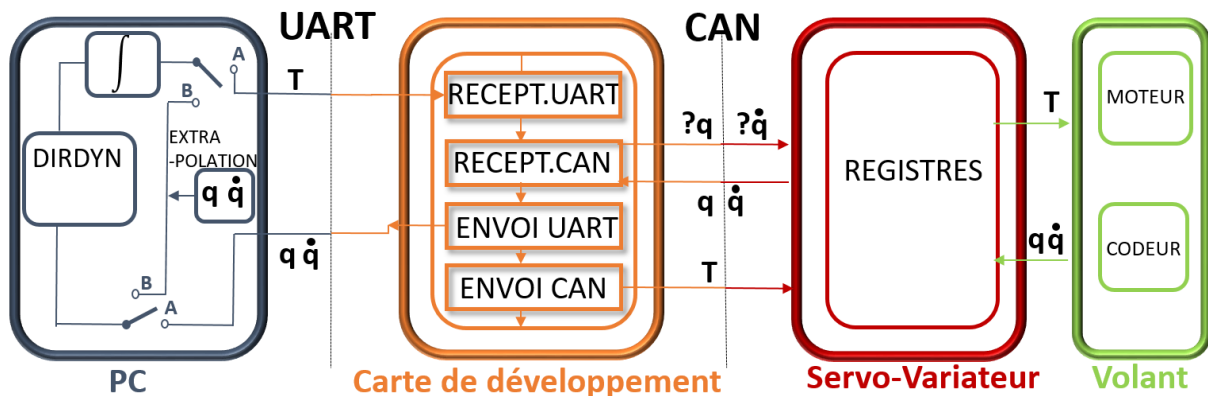


FIGURE 2.10 – Architecture sérielle du simulateur de véhicule

Les conventions sont reprises dans le tableau 2.5 et 2.6 :

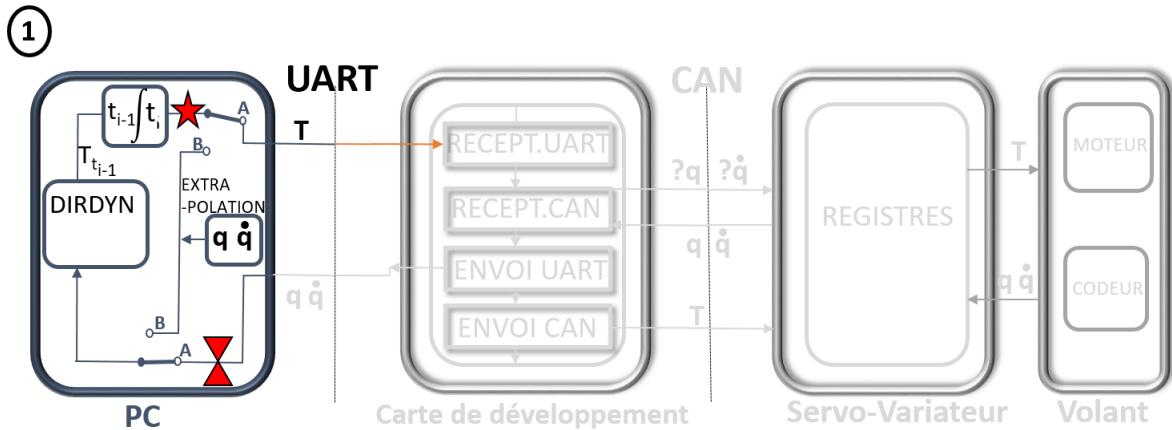
| Symboles | Significations |
|----------|----------------|
| ⌘ | En attente |
| ★ | Endroit actuel |

TABLE 2.5 – Symboles utilisés pour les diagrammes d'architectures

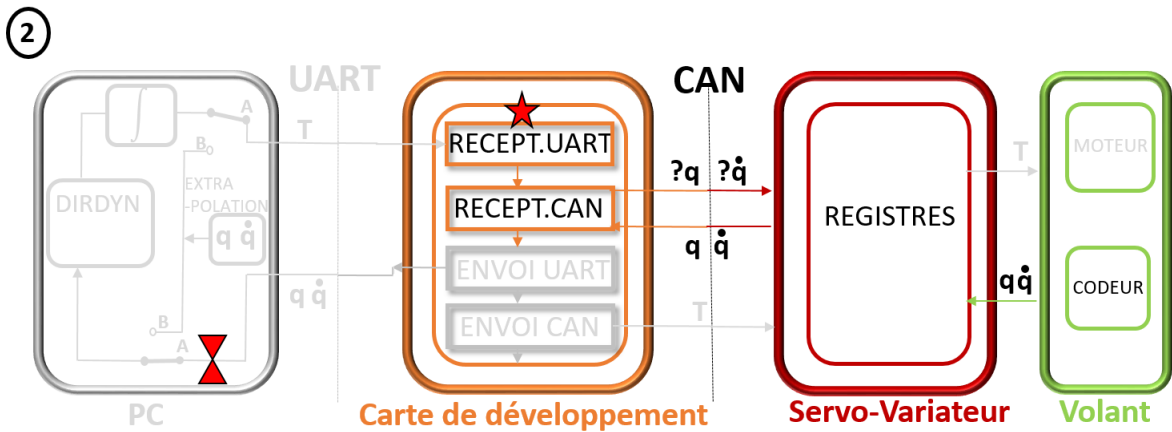
| Couleurs | Significations |
|------------|------------------------|
| ■ (bleu) | PC |
| ■ (orange) | Carte de développement |
| ■ (rouge) | Servo-variateur |
| ■ (vert) | Volant |
| ■ (gris) | Bloque non-actif |

TABLE 2.6 – Couleurs utilisées pour les diagrammes d'architectures

Après la phase de configuration des communications et du sero-variateur, considérons l'architecture à un endroit arbitraire, référencé par une ★ et suivons l'ordre des opérations indiqué par les flèches.

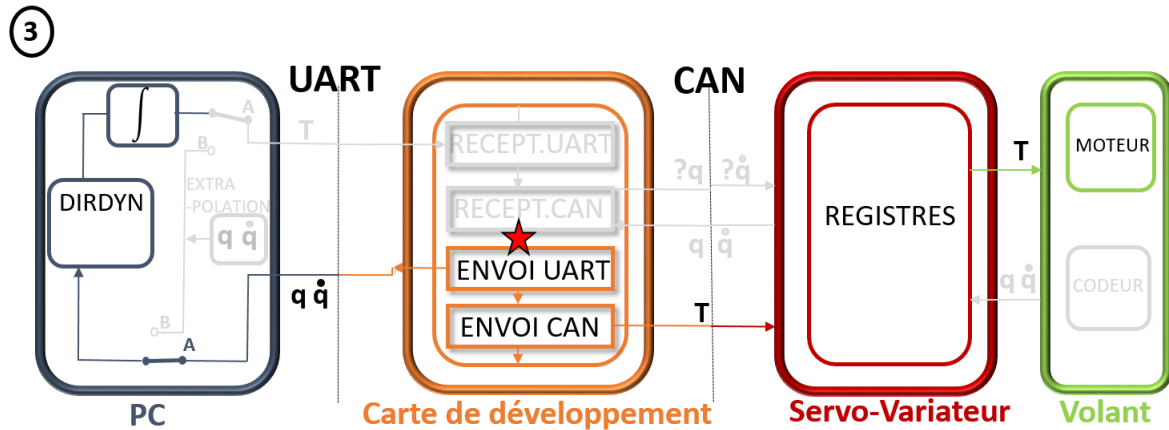


- ① En commençant à la branche supérieure A du PC.
1. Par l'intégration des équations du mouvement, les positions/vitesse indépendantes de chacune des articulations du modèle sont calculées pour le temps t_i .
 2. Le couple T du temps t_{i-1} , qui avait été préalablement calculé par la dynamique directe (tel que défini en section 2.3.1), est ensuite transmis via l'UART à la carte Texas³.
 3. Le PC est alors en attente (branche inférieure A). En effet, pour continuer et entamer la résolution de la dynamique directe en t_i , toutes les positions/vitesse indépendantes et toutes les positions/vitesse commandées du modèle sont indispensables. Il demande donc, de façon bloquante, que la **carte** lui envoie la position/vitesse du volant. Nous expliquons au chapitre suivant pourquoi ces informations ne sont pas encore disponibles. La branche B, au milieu du bloc PC, est pour l'instant totalement ignorée. L'interrupteur est toujours sur la branche A.



- ②
1. La boucle de contrôle sur la carte de développement commence par le mode "attente de réception UART" (module bloquant). Une fois que le couple envoyé par le PC s'est propagé à travers le câble et sous-couches software du PC/carte, il est stocké.
 2. La carte émet une requête, via le CAN, au **servo-variateur** pour obtenir la position et vitesse du servomoteur. Elle attend que la requête se propage et dès les q, \dot{q} reçues, les stocke.

3. $T_{t_{i-1}}$ est bien envoyé au temps t_i



③

1. Les q, \dot{q} reçus via CAN sont envoyés au PC via l'UART. Le processus *Robotran* du PC est donc enfin débloqué et peut à nouveau résoudre la dynamique directe, intégrer les \ddot{q}, \dot{q} (indépendants) et finalement revenir à l'étape ①.
2. Enfin, le couple qui avait été reçu au début de l'étape ② par la carte est envoyé via le CAN au servo-variateur qui commande en temps réel le servomoteur du Volant. Une transmission retour d'un paramètre de configuration pour le prochain envoi CAN est aussi reçu par la carte. Elle n'est pas représentée car son utilité n'est pas primordiale dans le cadre de la compréhension de l'architecture.

Remarques

La communication sur le port série se fait en envoyant uniquement des entiers. Le couple est multiplié par mille sur le PC et la position/vitesse est multipliée par mille sur la carte Texas. A la réception, ils sont tous les trois divisés par mille avant d'être utilisés.

Passer de double précision à simple précision divise par deux le nombre de bits à envoyer sur le port série et diminue ainsi les délais de communication. La perte de précision résultante n'est pas dommageable au vu de la précision du servo-variateur et du codeur (voir section 2.3.2).

2.6 Problèmes subsistants

Le simulateur complet tel que présenté est fonctionnel. L'utilisateur peut ainsi conduire en temps réel un *petit*⁴ modèle multicorps comme un kart, par exemple. L'utilisateur pilote le modèle en imposant position et vitesse et reçoit le couple de feedback dans ses bras. Néanmoins, plusieurs problèmes subsistent.

Vibrations dans le volant

Des vibrations détériorent et parasitent le rendu haptique de telle façon que la simulation proposée à l'utilisateur s'écarte de la situation physique qu'elle est supposée modéliser. En effet, pour une vitesse du véhicule simulé de 7 [m/s], ces vibrations sont de grande amplitude (> 1 [Nm]) et de relativement faible fréquence ($\approx 2 - 5$ [Hz]). Ci-dessous, deux exemples qui les mettent en évidence sont décrits :

4. avec peu d'articulations et de corps

1. Le virage stabilisé : au lieu d'avoir un retour de couple constant, il oscille avec de grandes amplitudes de telle sorte que l'utilisateur doit contracter exagérément ces muscles pour les atténuer et maintenir sa trajectoire.
2. Le recentrage du volant autour de 0° : dans un kart « réel », l'auto-alignement est grand. Si le pilote lâche le volant lors d'un virage, il se remet droit "quasi-instantanément". Dans le cas du kart sur simulateur, il y a bien ce phénomène de recentrage mais les oscillations empêchent toute stabilisation. Plus problématique encore, ces oscillations s'amplifient et rendent le volant instable.

La figure 2.11 illustre la prise d'un virage à un angle de braquage constant. Avant la prise de courbe en $t = 4$ [s], les oscillations s'atténuent grâce à une forte contraction des muscles. Par contre, pendant la prise du virage, les oscillations ne s'amortissent pas car l'utilisateur contracte ses muscles normalement (avec la même contraction que lors de la conduite d'un véhicule sur la route par exemple).

La figure 2.12 présente la situation où l'utilisateur conduit le véhicule en ligne droite et lâche progressivement le volant à partir de $t \approx 2$ [s]. Des oscillations apparaissent puis s'amplifient dans le volant. En $t \approx 4$ [s], l'utilisateur reprend le volant et le stabilise grâce à un couple important (> 5 [Nm]).

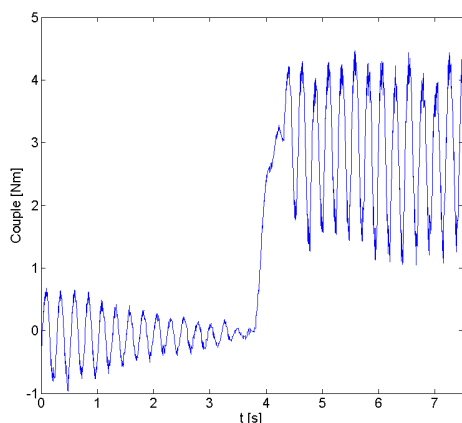


FIGURE 2.11 – Couple au volant lors de la prise d'un tournant à 7 [m/s]

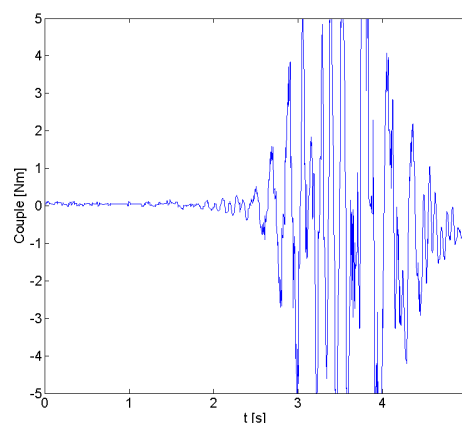


FIGURE 2.12 – Couple au volant en ligne droite à 7 [m/s]

Crashes sporadiques de communication

Après un temps aléatoire, une erreur survient. Le volant ne dirige plus le modèle multicorps et garde la dernière valeur de couple qu'il a reçue. Un redémarrage complet du simulateur est nécessaire pour simuler une nouvelle situation dynamique.

Conclusion

Suite à une méthode de conception adéquate, le simulateur de l'UCL a pris forme. Il est constitué de composants industriels comme de recherches et devrait délivrer des performances de haute qualité.

L'utilisateur peut conduire en temps réel un véhicule qui lui renvoie un couple réaliste issu de son modèle multicorps. Cependant, des vibrations détériorent le rendu haptique. Une amélioration du simulateur est indispensable pour pouvoir l'exploiter.

Chapitre 3

Développement du simulateur

Cette section développe les étapes principales qui ont permis *in fine* d'améliorer le simulateur.

La priorité a été de comprendre l'origine des vibrations. L'origine hardware est directement exclue car un (très) petit modèle multicorps donne un rendu haptique satisfaisant. De plus, changer le hardware n'était pas une possibilité de ce travail. La piste software est donc étudiée.

La première section analyse l'architecture sérielle implémentée par les étudiants du travail précédent, met en évidence ses failles et motive la nécessité d'améliorer l'architecture.

La deuxième section détaille l'implémentation de la nouvelle architecture et évalue ensuite ses performances. En conclusion, un bilan de la solution est établi.

La troisième section expose une dernière architecture, extension de la précédente aux résultats insuffisants. Ses performances sont caractérisées et un bilan à nouveau dressé.

Les vibrations n'ayant pas disparu, deux nouvelles pistes sont étudiées au cours des dernières sections. La qualité des entrées et sorties du modèle multicorps est analysée et dans un second temps, la validité du modèle multicorps est remise en cause.

Finalement, ce chapitre détaille les développements mécaniques apportés au simulateur.

3.1 Architectures de contrôle

Le servomoteur est en prise directe avec le volant. Excepté les roulements de l'arbre moteur, la transmission du couple au volant se fait sans aucune friction. De ce point de vue, le retour haptique est d'une pureté remarquable.

Par contre¹, ce très faible coefficient s'accompagne d'une très faible inertie. Le couple est peu filtré par des éléments mécaniques avant le volant.

Dès lors, une grande fréquence d'émission de couple est nécessaire afin d'éviter des « à-coups ». Au vu des vibrations présentes dans le volant, cette fréquence de rafraichissement de l'architecture sérielle semble insuffisante.

Cette section commence par mettre en évidence la limitation en fréquence de l'architecture sérielle et développe ensuite deux architectures visant à l'augmenter.

3.1.1 Analyse de l'architecture sérielle

L'architecture sérielle du chapitre précédent est reprise en figure 3.1. Au chapitre précédent, la configuration de l'interrupteur en position B a été ignorée. Cela n'a pas empêché la compré-

1. C'est également un avantage car, de façon générale, les frottements sont difficiles à modéliser avec précision.

hension du fonctionnement de l'architecture mais nous allons expliquer son rôle.

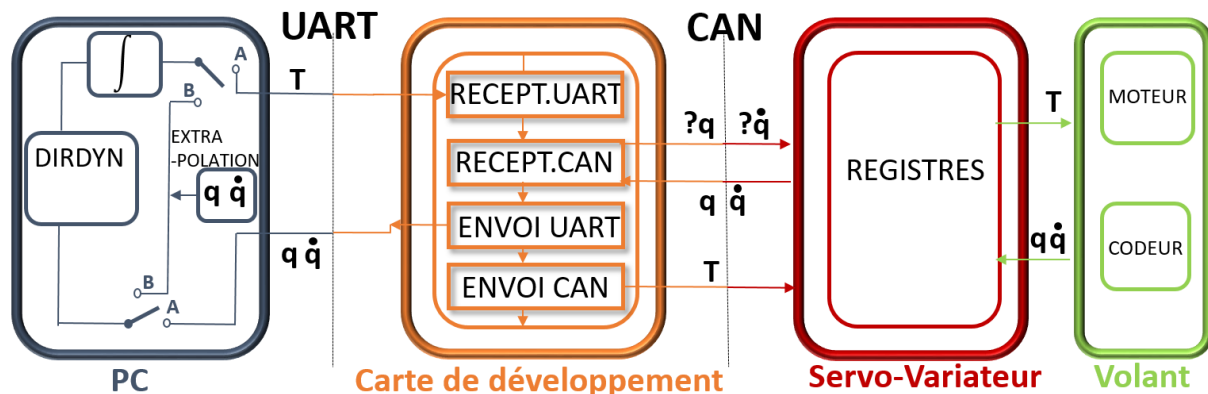


FIGURE 3.1 – Architecture N°1 : architecture sérielle

En position A, au vu de l'explication donnée en section 2.5, le processus *Robotran* reste bloqué en attente de q, \dot{q} après avoir envoyé T . Le temps que T se propage jusqu'à la carte Texas + le temps de propagation/traitement sur la carte + le temps de propagation CAN de la requête q, \dot{q} jusqu'au servo-variateur + le temps de traitement sur Lexium + le temps que les q, \dot{q} fassent le chemin retour jusqu'au PC : *Robotran* ne peut plus strictement effectuer aucun calcul (figure 3.2).

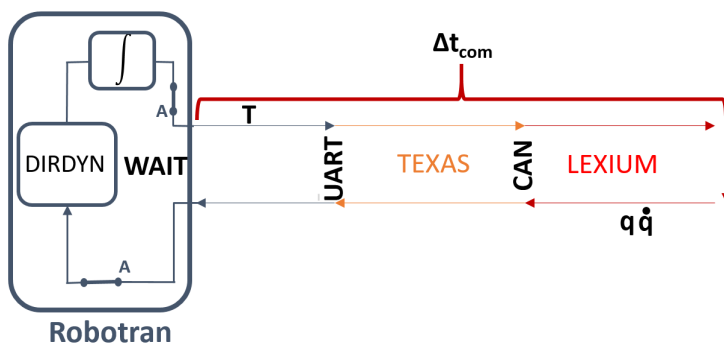


FIGURE 3.2 – Illustration du "gel" de *Robotran*

Ce temps, référencé comme Δt_{com} sur la figure 3.2, pénalise extrêmement le ratio temps réel car dans la position A :

$$v_{r,A} = \frac{\Delta t_{sim}}{\Delta t_{calc,DD} + \Delta t_{calc,int} + \Delta t_{com}} \quad (3.1)$$

où Δt_{sim} est choisi adéquatement (cfr section 2.3.1) et où :

$$\Delta t_{com} \gg (\Delta t_{calc,DD} + \Delta t_{calc,int})$$

En position B, la boucle de communication est court-circuitée : T n'est plus envoyé et les q, \dot{q} plus "demandées à être réceptionnées" : $\Delta t_{com} = 0$. En remplacement, une extrapolation sur la dernière position du volant reçue $q_{t_{last}}$ (en position A) est fournie en début de chaque dynamique directe :

$$q(t) = q_{t_{last}} + \dot{q}_{t_{last}}(t - t_{last})$$

où t est le temps courant de simulation.

Cette extrapolation diminue *en moyenne* les discontinuités de q entre deux phases A.

La position B de l'interrupteur permet donc de diminuer le nombre de ces pas de temps critiques $\Delta t_{calc,DD} + \Delta t_{calc,int} + \Delta t_{com}$ et $v_{r,série}$ de l'architecture sérielle s'écrit alors :

$$v_{r,série} = \frac{\Delta t_{sim}}{\Delta t_{calc,DD} + \Delta t_{calc,int} + f_A \Delta t_{com}} \quad (3.2)$$

où $f_A = \frac{\#(A)}{\#(A+B)}$ définit la proportion entre le nombre de fois où l'interrupteur est dans la position A pour le nombre total de fois où il est dans la position B et A.

La figure 3.3 illustre qualitativement l'idéal à trouver pour f_A .

En situation (3), la communication de couple se fait pour chaque nouveau couple calculé, pour chaque Δt_{sim} . Cela maximise la fréquence de rafraîchissement du couple au volant f_R . Par contre, comme $f_A = 1$, cela fait décroître $v_{r,série}$.

En situation (1), le nombre de communication est plus faible qu'en situation (3). $v_{r,série}$ est le plus grand des trois cas mais la fréquence de rafraîchissement du couple plus basse.

En situation (2), la situation intermédiaire est exposée.

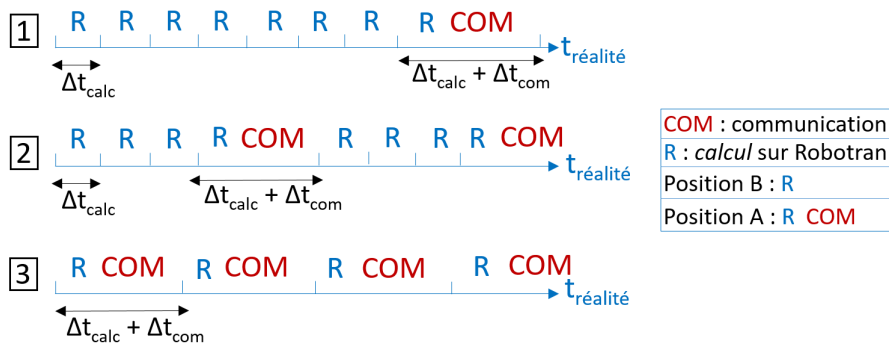


FIGURE 3.3 – Illustration qualitative du compromis entre le temps en position A et B

Pour le modèle de kart utilisé par les étudiants², f_A est fixée implicitement par :

- $\Delta t_{sim} = 0.5$ [ms] : il est choisi pour que la simulation soit précise et stable, c'est-à-dire fermant précisément les boucles du modèle et assurant la stabilité numérique de l'intégrateur.
- $f_{R,série} = 60$ [Hz] : la simulation est à la limite du temps réel $v_{r,série} \approx 1$ avec cette fréquence de rafraîchissement du couple.

Toutes les $\frac{1}{60}$ [s] l'interrupteur est placé en position A pour une seule émission de couple puis suivent 32 pas de temps sur la position B.

En conclusion, au vu des vibrations sur le volant, $f_{R,série}$ est jugée insuffisante. Sur cette architecture, à Δt_{sim} fixe, elle ne peut être augmentée sous peine de ne plus être en temps réel.

3.1.2 Architecture Multi-Thread

Partant du constat de la pénalité qu'ajoute le pas de temps de communication Δt_{com} sur le ratio temps réel v_r , l'architecture Multi-Thread est conçue pour affranchir le processus *Robotran* des délais de communication.

Autrement dit, l'idée mère de l'architecture Multi-Thread est de rendre le processus *Robotran*

2. différent de celui présenté en section 1.2

indépendant de tout le reste de la boucle exposée en figure 3.2 (communication + propagation/traitement de données).

Principe

Un thread (ou fil d'exécution) définit l'exécution d'un ensemble d'instructions du langage machine d'un processeur. Chaque thread a sa propre pile d'appel et ses registres [20].

Un processus définit aussi une séquence d'instruction d'exécution mais contrairement aux threads, chaque processus a des zones de mémoires physiques privées.

Un processus est dit multi-thread s'il exécute plusieurs threads *simultanément*.

L'architecture Multi-Thread (M-T) découpe le processus du PC de l'architecture sérielle en deux threads.

- Dans l'architecture sérielle : un unique thread s'occupe de calculer le couple (module "DIRDIN" + module " \int " sur le schéma 3.2) ainsi que de l'émission de T et de la réception de q, \dot{q} .
- Dans l'architecture M-T : un thread *Robotran* s'occupe uniquement du calcul de couple (module DIRDIN + \int sur le schéma 3.2) et un autre thread s'occupe de la réception de q, \dot{q} ainsi que de l'émission de T vers la carte Texas.

Ces deux threads sont chacun assignés à un cœur physique différent du PC. Ainsi, hormis la sur-couche software (l'OS par exemple), aucune dépendance ne les lie.

Conceptuellement, l'architecture définit un organe de calcul et un organe de communication.

Fonctionnement

La figure 3.4 illustre l'architecture Multi-Thread. Excepté le bloc PC, le schéma est le même que celui de l'architecture sérielle (figure 3.1).

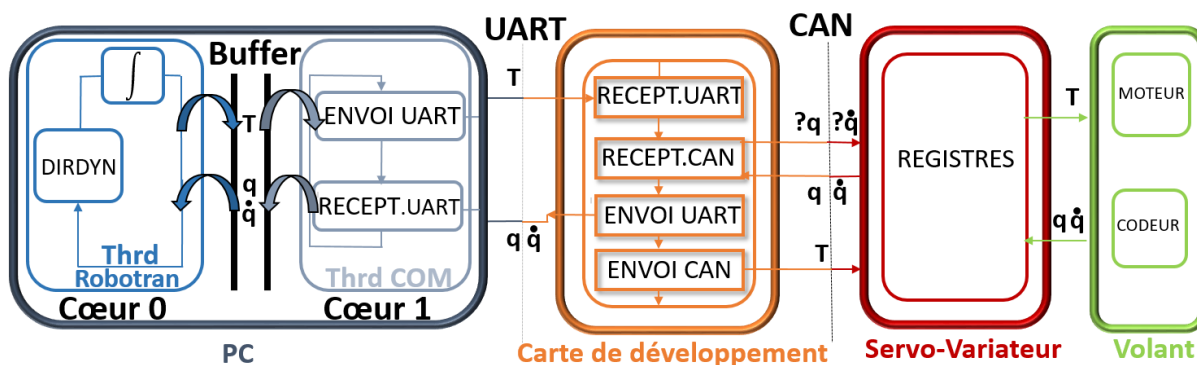


FIGURE 3.4 – Architecture N°2 : architecture Multi-Thread

Bien que les deux threads soient indépendants, ils doivent s'échanger les T et q, \dot{q} . Cet échange est assuré par une mémoire partagée, un buffer commun aux deux threads.

Le thread *Robotran* calcule le couple T au temps t_i via le bloc dynamique directe. Après l'intégration, ce couple est déposé dans un buffer et le thread *Robotran* continue son exécution. Il prend dans le buffer les q, \dot{q} du volant. Il ne vérifie pas leur validité : il prend deux variables dans deux endroits en mémoire indiqués. Il a donc toutes les variables q_u, q_c pour continuer la boucle.

Le thread de *Communication* (thrd COM sur la figure 3.4) fait les opérations inverses. Il copie le couple présent dans le buffer et l'envoie à la carte Texas via l'UART.

Il passe ensuite en mode "attente de réception des q, \dot{q} " et seulement après leurs réceptions, les dépose dans le buffer. Il revient dès lors à la première étape, prend le couple dans le buffer et ainsi de suite.

Évaluation

Le ratio temps réel du thread *Robotran* n'est plus pénalisé par Δt_{com} :

$$v_{r,MT} = \frac{\Delta t_{sim}}{\Delta t_{calc,DD} + \Delta t_{calc,int}} \quad (3.3)$$

où Δt_{sim} est le degré de liberté de l'architecture. Les simulations peuvent être exécutées avec un pas de temps d'intégration Δt_{sim} plus petit³ tout en maintenant $v_r \geq 1$. La diminution se fait à $v_r \geq 1$ tant que le processeur du cœur utilisé par le thread *Robotran* est suffisamment puissant pour résoudre les équations en $\Delta t_{réalité} \approx \Delta t_{sim}$.

La table 3.1 met en évidence un exemple de cette amélioration. Elle a été obtenue en simulant le modèle multicorps du kart implémenté par les étudiants du mémoire précédent [11].

| v_r | Architecture série | Architecture M-T |
|-----------------------------|--------------------|------------------|
| $\Delta t_{sim} = 0.5$ [ms] | ≥ 1 | ≥ 1 |
| $\Delta t_{sim} = 0.1$ [ms] | 0.36 | 0.54 |

TABLE 3.1 – Table de comparaison de v_r : valeur moyenne pour une simulation de 20 [s]

A $\Delta t_{sim} = 0.5$ [ms], les deux architectures permettent une simulation multicorps en temps réel. Par contre, à $\Delta t_{sim} = 0.1$ [ms], aucune ne garantit le temps réel mais $v_{r,MT} = 1.4v_{r,série}$. Pour le modèle utilisé, le pas de temps limite assurant le temps réel est trouvé expérimentalement de $\Delta t_{sim,lim,MT} = 0.25$ [ms].

Une capture d'écran de l'utilisation des processeurs pour ce pas de temps limite est montré en figure 3.5. Le PC⁴ a quatre cœurs dont le cœur 0 où est assigné le thread *Robotran* et le cœur 1 où tourne le thread de *Communication*.

Le processeur sur lequel tourne le thread *Robotran* est saturé. L'objectif est atteint : le thread *Robotran* tourne à la vitesse maximale que le « modèle » de PC utilisé lui permet pour le modèle multicorps simulé.

Nous observons aussi que le thread de communication n'est pas saturé. Cela s'explique par les délais d'attente de réception de q, \dot{q} , analysés en section suivante.

3. comparativement à l'architecture sérielle.

4. processeur *Intel Core i5* avec 4 cœurs logiques, 2.5 [GHz] tournant sur *Windows 10*.

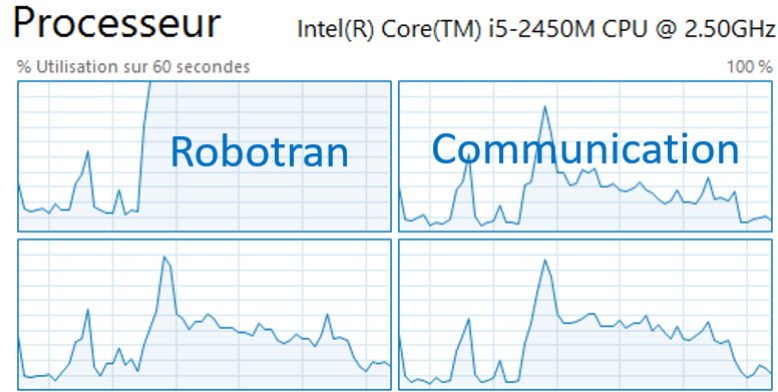


FIGURE 3.5 – Capture d'écran de l'utilisation des coeurs lors d'une simulation avec architecture M-T à $\Delta t_{sim,MT} = \Delta t_{sim,lim,MT} = 0.25$ [ms]

Bilan

L'architecture Multi-Thread améliore le simulateur car Δt_{sim} à la limite du temps réel a diminué.

Par contre, les tests de ressenti montrent que le volant vibre autant que pour l'architecture sérielle. Se référant à l'hypothèse de liaison "fréquence de rafraichissement $f_R \longleftrightarrow$ vibrations", f_R serait encore insuffisant.

Description La table 3.2 classe les deux fréquences ($\frac{1}{\Delta t_{sim,lim,MT}}$ et f_R) de l'architecture M-T.

| | Thread Robotran | Boucle de communication |
|---------------|-----------------|-------------------------|
| Fréquence f | 4 [kHz] | ~ 65 [Hz] |

TABLE 3.2 – Fréquence de l'architecture M-T

Notons trois remarques importantes sur la table :

- La fréquence de la boucle de communication $f_{R,MT}$ (thread COM + carte Texas) n'est pas un degré de liberté de l'architecture.
- Un déséquilibre de deux ordres de grandeur existe entre les fréquences des deux blocs :
 - Seulement 1 T calculé sur ~ 60 sera envoyé au moteur.
 - Une même paire q, \dot{q} est prise ~ 60 fois de suite par *Robotran* dans le buffer. Ces signaux sont donc « en escalier » et q peut être lissé à l'aide d'une extrapolation comme discuté en 3.1.1.
- $f_{R,série} \approx f_{R,MT}$: la fréquence de rafraichissement du couple n'a presque pas augmenté.

Afin d'identifier l'élément qui ralentit la boucle de communication, les six fonctions de communication sont recensées et leur temps d'exécution mesuré :

| | | | | | |
|-----------------------------------|--|-----------------|-----------|------------|------------|
| 1. Du coté PC : | | | | | |
| (a) Émission T : UART | | 1. PC | | (a) | (b) |
| (b) Réception q, \dot{q} : UART | | Δt [s] | 10^{-4} | 10^{-2} | |
| 2. Du coté de la carte Texas : | | | | | |
| (a) Réception T : UART | | 2. Texas | | (a) | (b) |
| (b) Émission q, \dot{q} : UART | | Δt [s] | 10^{-2} | 10^{-4} | 10^{-3} |
| (c) Réception q, \dot{q} : CAN | | | | | |
| (d) Émission T : CAN | | | | | |

TABLE 3.3 – Ordre de grandeur du temps d'exécution moyen pris par chaque fonction

Les fonctions les plus lentes sont donc les réceptions UART (1.b et 2.a).

Analyse Pour comprendre ces délais, analysons la figure 3.6. Les modules CAN dont le délai est comparativement petit, ne sont pas utilisés : le volant est déconnecté. De plus, le thread *Robotran* n'est non plus nécessaire pour ce schéma.

Comme pour l'explication de l'architecture sérielle, nous prenons la figure à l'endroit arbitraire référencé par ★ et suivons les flèches.

La carte Texas attend la réception des quatre bytes de l'entier T . Une fois reçu, il le stocke et passe au module suivant : l'envoi de q, \dot{q} vers le PC. Ce module s'exécute très rapidement selon la table 3.3⁵. L'information se propage alors à travers le câble série, les sous-couches du PC⁶, ... jusqu'à ce qu'il soit enfin disponible pour le thread de *Communication*. Le module de réception les dépose dans le buffer puis passe à la fonction d'émission UART à partir du couple disponible dans le buffer.

Comme ② se termine comparativement vite, de]② → ④], la carte Texas est entrain d'attendre un couple en ① qui n'a même pas encore été envoyé. Ce raisonnement est transportable pour le module de réception sur le PC.

Cette *désynchronisation* entre émission et réception explique les délais des réceptions UART de la table 3.3. A un plus haut niveau d'abstraction, cela explique aussi la faible fréquence $f_{R,MT}$ en table 3.2 : si la carte Texas attend un couple, elle ne peut pas l'envoyer par le CAN et ralentit donc le rafraichissement du couple.

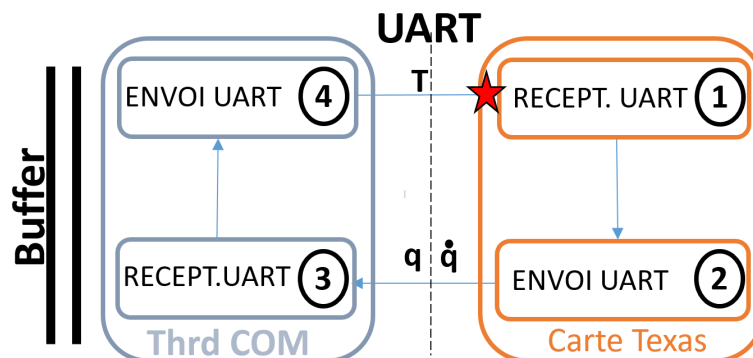


FIGURE 3.6 – Illustration de la boucle de communication sans le CAN

5. En pratique, c'est juste un dépôt dans une FIFO. La carte gère toute-seule l'émission à partir des données de la FIFO.

6. la carte Texas n'a pas d'OS.

3.1.3 Architecture avec interruptions

Répondant aux failles de l'architecture Multi-Thread, l'architecture avec interruption (architecture IRQ) augmente d'un degré l'indépendantisation des processus. Son idée fondatrice est d'indépendantiser *totalemment* les communications aller et retour sur chacun des matériels. Cette architecture permet de briser la chaîne de la figure 3.6 et de-facto d'augmenter la fréquence de rafraîchissement du couple.

Principe

Pour indépendantiser les communications sur le PC, le thread de *Communication* est séparé en deux threads, chacun assigné à un cœur différent de l'ordinateur hôte. Le thread *Robotran* reste lui inchangé. Du côté PC, l'architecture IRQ est donc une architecture à trois thread.

Pour indépendantiser les communications sur la carte Texas, plusieurs options ont été envisagées

- Le multi-thread : l'utilisation du multi-thread, tel que défini dans la section 3.1.2, n'apporterait qu'un gain en performance faible car les thread devraient se partager le seul processeur (ARM).
- Le multi-tâche : une tâche désigne un processus. Le multi-tâche lui désigne le procédé de planification et de partage du CPU (Central Processing Unit) entre plusieurs tâches [2] et donne l'illusion que les tâches sont effectuées parallèlement.

Un OS (la carte Texas est compatible avec l'OS *SYS/BIOS*) maximise l'utilisation du CPU en donnant des ordres de priorité aux tâches.

Dans le cas du simulateur, aucune ordre de priorité des tâches de communication n'est justifié.

Dès lors, l'architecture avec interruption a été choisie.

Les interruptions définissent un mécanisme hardware utilisé pour informer le CPU qu'un évènement asynchrone est survenu [2]. Le CPU sauve son espace de travail et passe à une fonction spécifique l'ISR (Interrupted Service Routine).

L'ISR est une routine, spécialement prévue par le CPU, qui traite l'évènement exceptionnel et puis fait retourner le programme à sa tâche interrompue (fonctionnement non-préemptif).

Fonctionnement

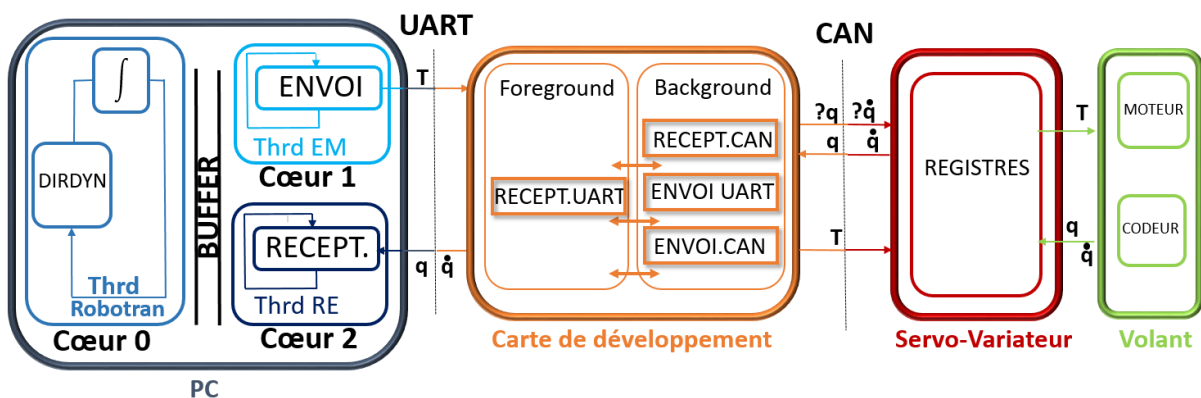


FIGURE 3.7 – Architecture N°3 : architecture avec interruptions

La figure 3.7 illustre l'architecture IRQ. Sur le PC, les trois threads sont associés à chacun des cœurs :

- *Le Coeur 0* : le thread *Robotran* calcule le couple, le dépose dans le buffer puis prend les q, \dot{q} du buffer, en boucle.
- *Le Coeur 1* : le thread *Emission* (EM) sélectionne le couple du buffer et l'envoie via l'UART à destination de la carte Texas, en boucle. Contrairement à ce qui est exposé en figure 3.7, le thread d'émission du PC n'envoie pas le couple T en continu. Un timer, dont la fréquence sera discutée à la prochaine section, limite le débit.
- *Le Coeur 2* : le thread *Réception* (RE) attend les positions/vitesses, les reçoit et les dépose dans le buffer, en boucle.

Chacun des trois threads sont indépendants des deux autres.

Sur la carte de développement, 2 domaines sont représentés sur un même processeur :

- *Le background* : La **boucle** qui réceptionne les q, \dot{q} du volant via CAN et les envoie au PC via UART (séquence principale). Elle envoie aussi T au volant, mais **seulement**, quand un nouveau couple a été reçu via l'UART (séquence secondaire).
- *Le foreground* : La **séquence** qui récupère le couple T envoyé par le *thread EM* et le rend disponible pour être envoyé au servo-variateur.

Le *background* est donc le domaine où s'effectue les opérations par défaut, alors que le *foreground* s'exécute seulement quand un nouveau couple est arrivé dans la FIFO de la carte.

La figure 3.8 précise ce partage de la carte en *background* et *foreground*. Lorsque la carte reçoit un nouveau T , une interruption se déclenche automatiquement, le programme s'interrompt et passe à l'ISR. Cette routine récupère uniquement T en (D), le copie dans un buffer, et indique à la boucle principale qu'un nouveau T a été reçu \blacksquare . Une fois cette séquence achevée, le programme retourne à l'endroit avant interruption (par exemple dans (B)). C'est seulement après avoir terminé la séquence principale du *background* que le couple est envoyé au servo-variateur (C) dans la séquence secondaire.

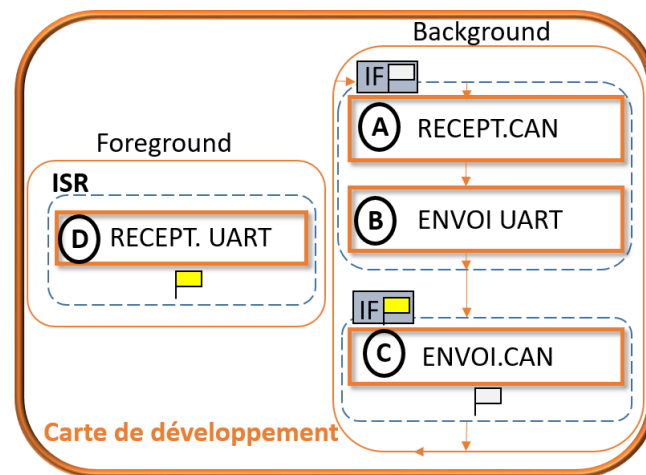


FIGURE 3.8 – Fonctionnement des interruptions sur la carte

Évaluation

L'architecture avec interruptions ajoute un degré de liberté par rapport à l'architecture Multi-Thread. Les deux *ddl* sont :

1. Le pas de temps de simulation Δt_{sim} .
2. La fréquence d'émission $f_{EM,PC}$ PC \rightarrow carte Texas (thread *Emission*)

1^{er} ddl Le premier degré de liberté est le même que celui de l'architecture Multi-Thread et en théorie, $\Delta t_{sim,lim,IRQ} = \Delta t_{sim,lim,MT}$. Pour un modèle donné, dans les mêmes conditions de simulation, il est toujours choisi pour garantir $v_{r,IRQ} = v_{r,MT} \geq 1$.

En pratique, l'ajout d'un troisième thread sur le PC impose une surcharge du travail à l'OS. Le thread *Robotran* a donc moins de "puissance de calcul" qui lui est réservé et Δt_{sim} doit être légèrement diminué.

Par exemple, pour le modèle multicorps de kart utilisé jusque là :

| | Architecture sérielle | Architecture MT | Architecture IRQ |
|----------------------|-----------------------|-----------------|------------------|
| $\Delta t_{sim,lim}$ | 0.5 [ms] | 0.25 [ms] | 0.5 [ms] |

TABLE 3.4 – Comparaison des pas de temps limites

Comme au cours de la section 3.1.2, la figure 3.9 met en évidence l'utilisation des cœurs pour l'architecture IRQ.

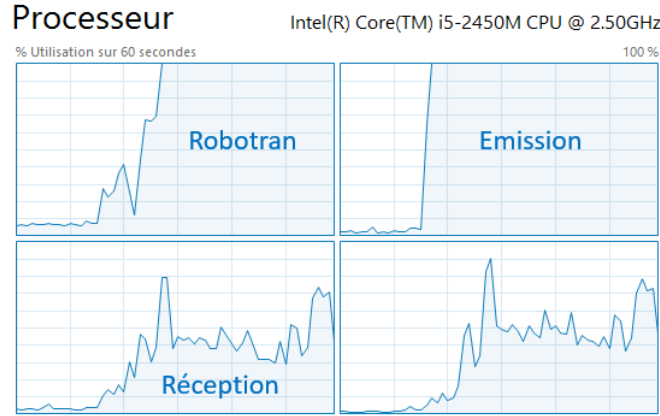


FIGURE 3.9 – Utilisation des processeurs pour l'architecture IRQ à $\Delta t_{sim,lim}$

Cette fois-ci, le coeur 0 où tourne le thread *Robotran* ainsi que le coeur 1 où tourne le thread *Emission* utilisent 100% du processeur. Le coeur 3 de réception est lui moins utilisé car il doit attendre les q, \dot{q} que la carte lui envoie (voir figure 3.8).

2^{ème} ddl Le deuxième degré de liberté est la fréquence d'émission sur le PC $f_{EM,PC}$ qui impose la fréquence de rafraichissement de couple f_R et impose aussi implicitement la fréquence de réception sur le PC.

Pour choisir adéquatement $f_{EM,PC}$, définissons la fréquence de chacun des sous-blocs de la figure 3.8 :

- f_{ISR} : la fréquence d'exécution de l'ISR.
- $f_{back,q}$: la fréquence d'exécution de la séquence principale du *background*.
- $f_{back,T} \triangleq f_R$: la fréquence d'exécution de la séquence secondaire du *background*.

Afin de simplifier l'analyse, nous posons $f_{ISR} \approx f_{EM,PC}$. Trois cas sont possibles en se référant à la figure 3.8.

1. $f_{EM,PC} < f_{back,q}$ et $f_{EM,PC} \approx f_R$: Tous les couples envoyés par le PC sont envoyés par la carte au volant.

- Avantage : La boucle principale du background est peu interrompue. Il y a donc un grand débit de communication des q, \dot{q} du volant vers le PC.
 - Inconvénient : Il y a plus de communications carte \rightarrow PC que de communications PC \rightarrow carte \rightarrow volant. f_R pourrait donc être plus grand en augmentant $f_{EM,PC}$.
2. $f_{EM,PC} = f_{back,q}$ et $f_{EM,PC} \approx f_R$: Limite où tous les couples envoyés par le PC sont envoyés par la carte au volant.
 - Avantage : Configuration qui équilibre le plus les trois fréquences de la carte et où f_R est max.
 - Inconvénient : Du coté PC, $f_{Robotran} = \frac{1}{\Delta t_{sim}} \gg f_{EM,PC}$ comparativement au cas 3. Autrement dit, certains couples que calcule le PC ne sont pas envoyés à la carte (même chose pour le cas 1).
 3. $f_{EM,PC} > f_{back,q}$ et $f_{back,q} \approx f_R$: Plus de couples sont envoyés par le PC à la carte que par la carte au volant.
 - Avantage : $f_{EM,PC}$ se rapproche de $f_{Robotran} = \frac{1}{\Delta t_{sim}}$ et $f_R \approx \max$.
 - Inconvénient : Le CPU est souvent interrompu : la séquence principale du background a tendance à mettre plus de temps à s'effectuer.

Nous remarquons que f_R sature en configuration 2. C'est pour atteindre cette fréquence, où tous les couples T qui déclenchent une interruption sont envoyés au volant que $f_{EM,PC} \approx 500 [Hz]$ est choisi.

La figure 3.10 illustre ce choix. Elle a été obtenue en moyennant le nombre d'envois de couple ((C) figure 3.8) sur 10 secondes de simulation, et ce pour 8 $f_{EM,PC}$ différentes.

Remarquons qu'en théorie, à saturation $f_{EM,PC} \approx 500 [Hz]$, le cas 2 prédit $f_{EM,PC} \approx f_R$. Sur la figure 3.10 par contre, $f_R < f_{EM,PC}$. Cette différence provient en majeure partie de l'approximation $f_{ISR} \approx f_{EM,PC}$ utilisée pour définir les trois cas (en réalité : $f_{ISR} < f_{EM,PC}$).

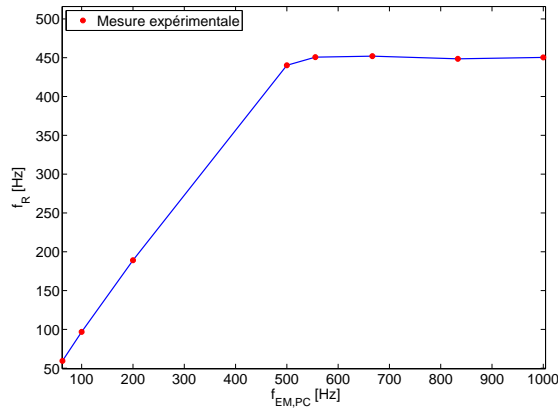


FIGURE 3.10 – Évolution de la fréquence de rafraîchissement du couple en fonction de la fréquence d'émission sur le PC

Bilan

L'architecture IRQ améliore le simulateur. Elle diminue les temps d'attente et augmente la fréquence de rafraîchissement. S'appuyant sur l'hypothèse de " $f_R \longleftrightarrow$ vibrations", le simulateur devrait donner de meilleurs sensations "haptiques".

Cependant, les vibrations restent toujours dominantes de telle façon qu'il n'est *quasiment pas* possible de "ressentir" une amélioration.

La fréquence utilisée mise en perspective avec le domaine de fréquences haptiques sensibles

présentées en section 2.1, infirme l'hypothèse : les oscillations problématiques du volant ne sont pas directement liées à la fréquence de rafraîchissement du couple. D'autres causes sont étudiées dans les sections suivantes de ce chapitre.

3.2 Entrées et sorties du modèle multicorps

Cette section s'inscrit dans le développement du simulateur et dans la recherche de l'origine des oscillations du volant.

Au cours de cette section, les signaux d'entrées et de sortie du *module multicorps* sont présentés. Quelques traitements basiques de ces signaux sont ensuite développés et leurs effets sur le couple au volant sont analysés.

Qualité du Signal

Les figures 3.11 et 3.12 montrent les signaux de positions q et vitesses \dot{q} du volant pour une simulation où l'utilisateur prend des virages de dynamiques variés. Les signaux tels que présentés sont ceux tels qu'utilisés par le thread *Robotran* (i.e pas en sortie du codeur où les signaux seraient beaucoup plus lisses).

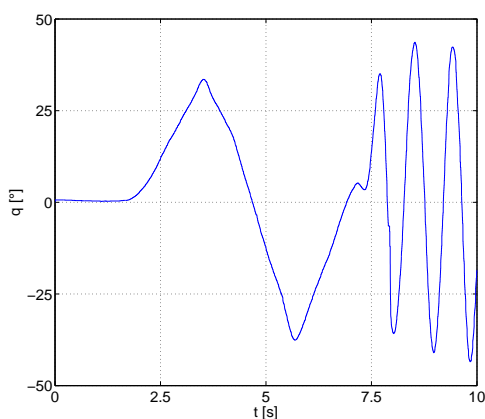


FIGURE 3.11 – Signal de la position q

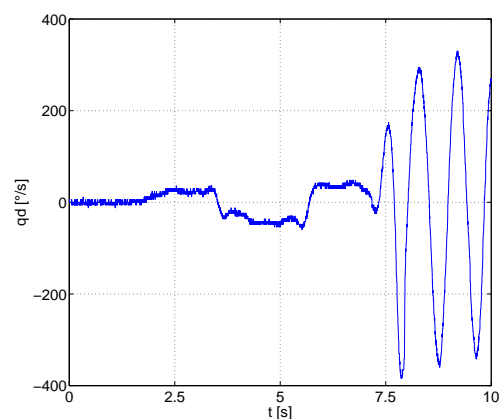


FIGURE 3.12 – Signal de vitesse \dot{q}

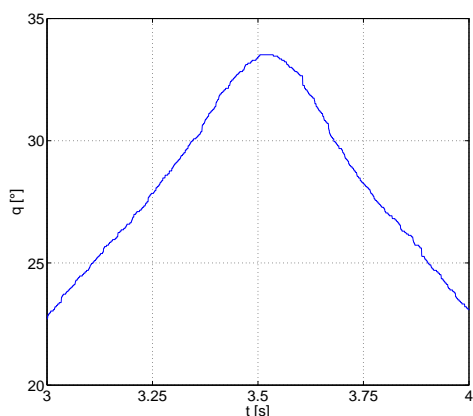


FIGURE 3.13 – Zoom de la figure 3.11

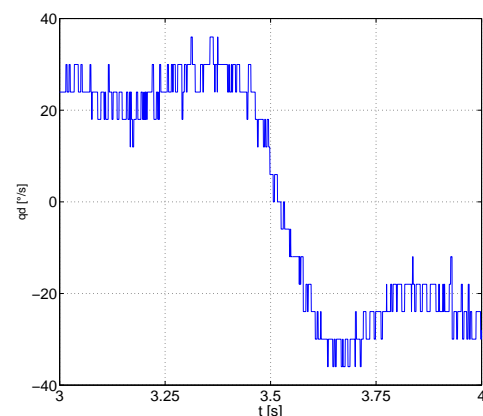


FIGURE 3.14 – Zoom de la figure 3.12

D'après les figures 3.11 et 3.12, les signaux apparaissent *de bonne qualité* : il y a peu de bruit, il n'y a pas d'outliers, q et \dot{q} sont cohérents l'un vis à vis de l'autre,... En se concentrant sur une partie de q (figure 3.13), nous observons qu'il est donc peu bruité mais

qu'il augmente ou diminue par une succession de marches de faible amplitude ($< 0.5^\circ$). Comme analysé en section précédente $f_{Robotran} > f_{réception,PC}$: le thread *Robotran* suréchantillonne les q, \dot{q} du buffer PC.

Le zoom en figure 3.14 montre que le signal de vitesse \dot{q} est constitué d'impulsions positives et négatives par rapport à une composante filtrée idéale. Ces impulsions sont en moyenne de $+/- 6$ [$^\circ/s$] d'amplitude et détériore fortement le signal comparativement à q .

Filtres du signal

Pour que le modèle multicorps calcule un couple moins bruité, les signaux d'entrée doivent être filtrés, l'intégrateur Newton-Euler explicite de *Robotran* supportant mal les signaux formés d'impulsions.

Nous détaillons ci-dessous les filtres appliqués au signal \dot{q} .

Deux filtres ont été implémentés sur le PC :

1. Un filtre passe-bas du premier ordre
2. Un filtre anti-impulsions

La fonction de transfert d'un filtre passe-bas est donnée par :

$$H(s) = K \frac{1}{1 + s\tau} \quad (3.4)$$

τ désigne la constante de temps et est le degré de liberté du filtre. Selon le diagramme de Bode, plus τ augmente et plus la bande passante se réduit.

Sélectionner le $\tau_{idéal}$ demande un compromis entre :

- Les virages à hautes dynamiques où le $\tau_{idéal}$ est "bas" pour limiter le retard entre \dot{q} filtrée et mesurée.
- Les virages à faibles dynamiques où le τ idéal est "élevé" pour filtrer un maximum les impulsions plus présentes que dans les virages à hautes dynamiques.

La figure 3.15 illustre ce compromis. Pour des virages à $\dot{q} \approx cste$ (ex : $t = 0.75 \rightarrow 0.1$ [s]), le signal rouge filtre une partie des impulsions. Le filtre ne peut être choisi beaucoup plus sélectif auquel cas le signal rouge serait en retard dans des phases à dynamique *moyenne* (ex : $t = 0.5 \rightarrow 0.75$ [s]) et dynamique *élevée* (ex : $t = 7.5 \rightarrow 10$ [s] sur la figure 3.12).

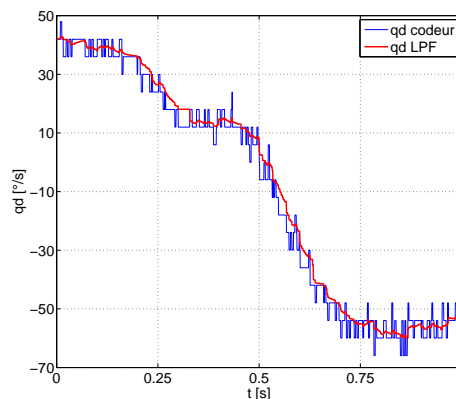


FIGURE 3.15 – Effet du filtre ($\tau = 0.014$) sur \dot{q}

Avec le τ choisi, certaines impulsions sont toujours présentes. Certaines impulsions sont problématiques, notamment celles à $\dot{q} = 0$ [$^\circ/s$]. Si le volant est centré en position stable, ces erreurs vont le déstabiliser par effet d'emballement.

Un filtre anti-impulsions est implémenté pour en éliminer la majorité. Il utilise une connaissance à priori de l'amplitude des oscillations ($\dot{q} = 6$ [°/s] ou $\dot{q} = -6$ [°/s]) autour de $\dot{q} = 0$ [°/s] et les remplace par $\dot{q} = 0$ [°/s]. Une machine d'état-fini est implémentée pour déterminer quand le signal peut être forcé à $\dot{q} = 0$ [°/s].

Effet sur le couple

Les différents types de filtres implémentés n'ont aucune influence sur les oscillations problématiques du signal de couple (tel que montré en figure 2.11). L'hypothèse "qualité des signaux \longleftrightarrow oscillations de couples" peut être infirmée. Les signaux d'entrées du *module multicorps* (figure 2.1) ne sont pas directement responsables des oscillations du volant.

Toutefois les filtres améliorent le signal de couple. Pour le mettre en évidence, nous utilisons les améliorations apportées au modèle multicorps qui elle suppriment les oscillations de couple (voir section suivante).

Dans ces conditions, en figure 3.16 et 3.17, deux enregistrements de couple pour deux simulations de prise de virages semblables sont montrés. Seule la simulation de la figure 3.17 a été faite avec un filtre "passe-pas anti-impulsions" sur la vitesse du volant.

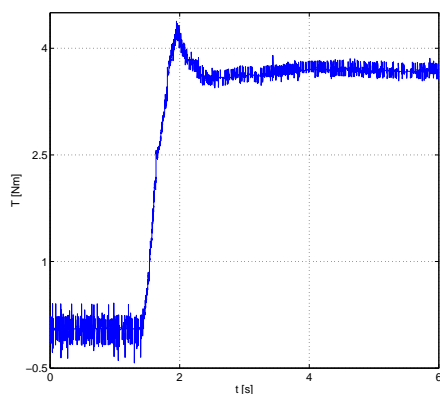


FIGURE 3.16 – Couple avec \dot{q} non-filtré

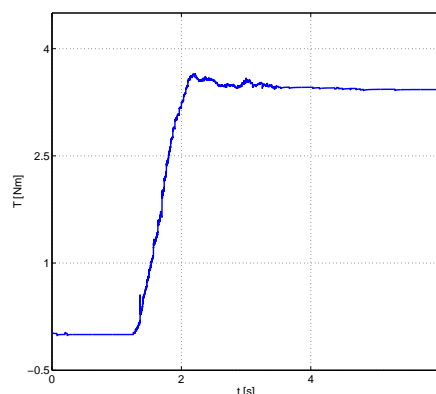


FIGURE 3.17 – Couple avec \dot{q} filtré passe-bas à $\tau = 0.014$ [s]

L'amélioration apportée par le filtre est une réduction non négligeable du contenu fréquentiel de T . Ces améliorations se ressentent au volant : elles confèrent des sensations plus « pures ». Ce type de filtre est donc retenu.

Enfin, pour obtenir un ressenti encore plus lisse, un filtre passe-bas peut être ajouté sur le couple (donc en sortie du *module multicorps*).

Limite

Le filtrage améliore le rendu haptique du volant mais doit rester faible ($\tau < 0.02$ pour notre cas). En effet, le retard résultant du passe-bas fausse la dynamique du véhicule ressentie par le couple au volant. Le ressenti est moins physique, plus "élastique".

3.3 Analyse du modèle multicorps

Cette section analyse un dernier facteur influençant la qualité du couple ressenti par l'utilisateur : le modèle multicorps. Elle développe quatre caractéristiques du modèle qui pourraient détériorer la qualité du signal de couple.

3.3.1 Influence du type de modèle multicorps

Deux modèles multicorps ont été utilisés au cours de ce travail :

1. Le modèle V1 : le modèle de kart implémenté par les étudiants du mémoire précédent [11] qui a été utilisé pour illustrer le développement des architectures.
2. Le modèle V2 : le modèle de kart présenté en section 1.2. Il a été créé pour le projet du cours de dynamique des véhicules [7] et n'était donc pas disponible dès le début du mémoire contrairement au modèle V1.

En figure 3.18 et 3.19, un tournant semblable est pris respectivement avec le modèle V1 et V2. Dans les deux simulations :

- L'utilisateur contracte *normalement*⁷ ses muscles des avant-bras.
- La vitesse du kart est $V = 7$ [m/s].
- L'utilisateur initie le braquage en $t \approx 1$ [s] et maintient son angle de braquage \sim constant jusqu'à la fin de la simulation ($t = 3$ [s]). θ_{braq} (et donc T_{moy}) est légèrement plus grand pour la simulation du modèle V2 en figure 3.19.
- Les forces de contact sont calculées grâce au modèle de Bakker ainsi qu'au terme de relaxation (section 1.2.2.2).
- Le couple maximal est limité à 5 [Nm].

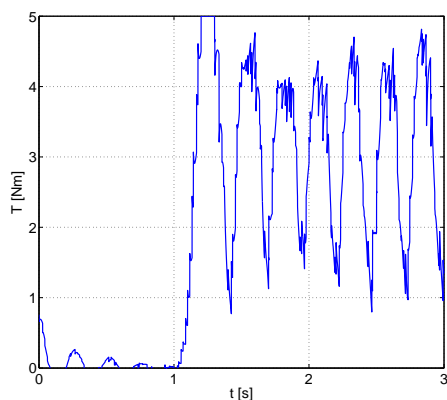


FIGURE 3.18 – Prise d'un virage avec le modèle V1

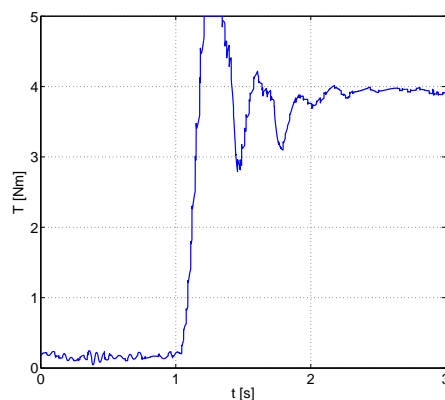


FIGURE 3.19 – Prise d'un virage avec le modèle V2

Les figures montrent distinctement que le modèle V2 améliore la réponse du couple en amortissant ses oscillations.

A première analyse, les différences entre les modèles ne semblent pas caractéristiques d'une source d'oscillation non-amortie. Les différences principales entre les deux versions sont :

- La géométrie/masse/inertie du kart : la géométrie du kart V2 a été réalisée en grande partie à l'aide de mesures prises sur kart réel.
- La formalisation multicorps : le kart V2 a été dessiné sous MBSysPad en minimisant le nombre de joints (V1 : 20, V2 : 19), de corps (V1 : 8, V2 : 7) et le nombre de coordonnées à entrer pour les points des corps (V1 : 26, V2 : 20).

Le modèle des forces extérieures est le même pour les deux modèles.

Cependant, les résultats montrent que ces différences, insérées dans la boucle fermée du simulateur ont une influence. Une analyse modale du modèle V1 pourrait mettre en évidence ces

7. de la même façon qu'un utilisateur qui ne sait pas que le couple oscille

fréquences caractéristiques instables.

Le modèle V2 est donc utilisé pour le simulateur.

3.3.2 Influence de la relaxation du pneu

Cette section analyse l'effet de la relaxation σ (présentée en section 1.2.2.2) sur les forces latérales de contact ainsi que le couple au volant qu'elles induisent.

Sur simulation

Les figures 3.20→3.23 sont obtenues avec une commande fonction du temps à la place de l'utilisateur (uniquement le PC est en fonctionnement) :

- Vitesse du kart (modèle V2) : $V = 7$ [m/s]
- Temps du braquage : $t = 1$ [s] $\rightarrow t = 1.9$ [s]
- Angle de braquage θ_{braq} : $0^\circ \rightarrow 10^\circ$ linéairement, vers la droite

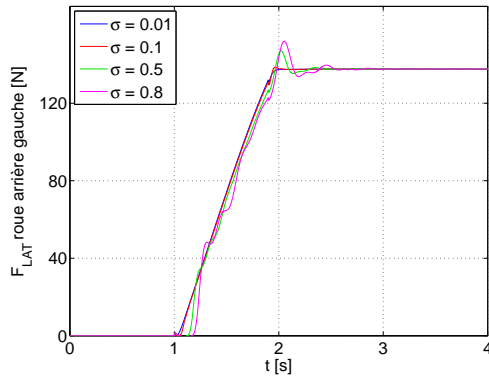


FIGURE 3.20 – Évolution de la force latérale

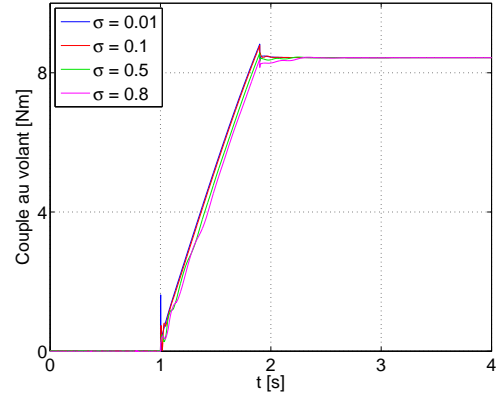


FIGURE 3.21 – Évolution du couple

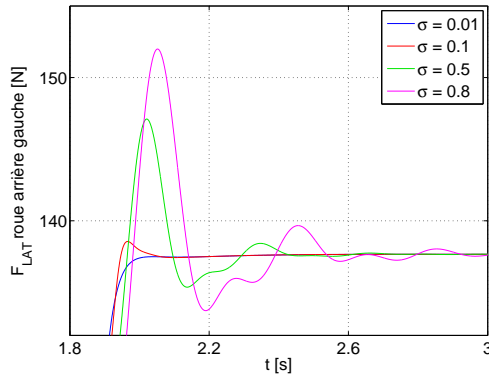


FIGURE 3.22 – Zoom de la figure 3.20

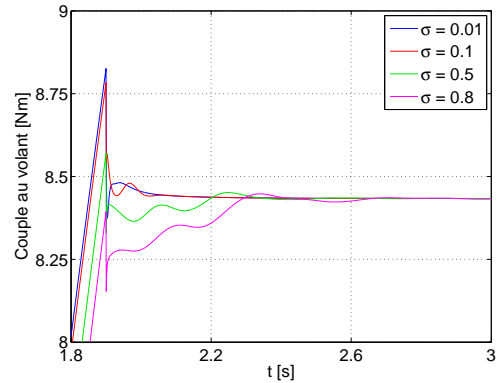
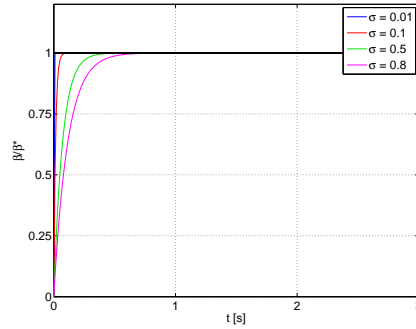


FIGURE 3.23 – Zoom de la figure 3.21

La force et le couple entrent en phase transitoire en $t = 1$ [s]. Ils croissent puis, à partir de $\theta_{braq} = \theta_{braq,max}$, se stabilisent.

La force latérale à $\sigma = 0.01$ [m] se rapproche du cas sans relaxation : $F_{lat}(t) = F_{lat,Bakker}^*(1 - e^{-\frac{V}{\sigma}t})$ où $e^{-\frac{V}{\sigma}t} \approx 0$.

Pour $\sigma = 0.1 \rightarrow 0.8$ [m], plus σ augmente plus l'overshoot, le temps de montée et le temps de réponse augmentent. Ces phénomènes ne sont pas présents dans la réponse prédite par la solution analytique (voir figure 3.24)

FIGURE 3.24 – Solution analytique pour $V = 7$ [m/s]

De plus, conjointement à l'overshoot, le signal oscille. La période et l'amplitude de ces oscillations augmentent avec σ . L'amortissement diminue lui drastiquement. Ainsi, dans les conditions mentionnées :

Pour $\sigma = 0.01$:

- $f \approx 0$ [Hz]
- Temps d'établissement : $t_r \approx 1$ [s]

Pour $\sigma = 0.8$:

- $f \approx 5$ [Hz]
- Temps d'établissement : $t_r \approx 3$ [s]

Sur le simulateur

Les figure 3.25 et 3.26 illustrent la prise d'un virage semblable avec et sans relaxation respectivement :

- La vitesse du kart (modèle V2) : $V = 7$ [m/s].
- L'utilisateur a déjà braqué à $\theta_{braq} \approx 35^\circ$ en $t = 0$ [s] et à $t \approx 2$ [s] relâche complètement le volant de ses mains.

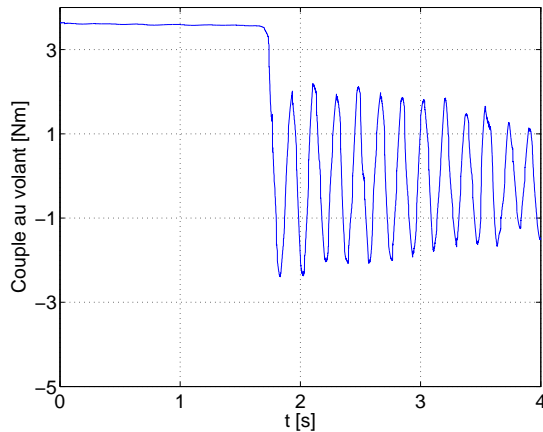


FIGURE 3.25 – Stabilisation du volant avec relaxation

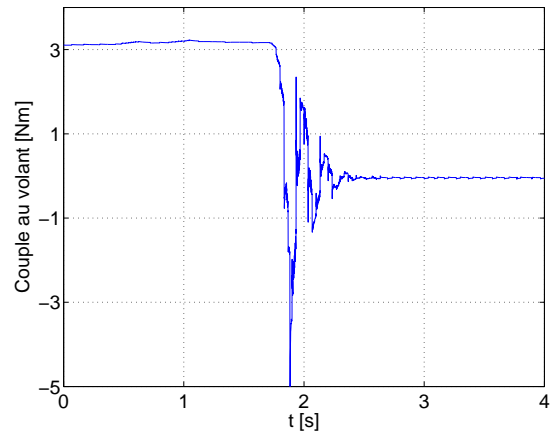


FIGURE 3.26 – Stabilisation du volant sans relaxation

Sur le simulateur avec le terme de relaxation, le volant oscille autour de 0° . Comme montré plus haut, ces oscillations s'atténuent (mais plus lentement qu'en simulation seule à cause des filtres, temps de communication, ...).

Dans ces même conditions d'expérience en supprimant uniquement la relaxation du pneu, le volant s'auto-centre et se stabilise en $\Delta t < 0.5$ [s] (à titre illustratif car ce temps dépend des conditions de simulations, filtres, ...). Le couple est plus bruité car les forces sont moins filtrées par la relaxation. Cependant la dynamique du kart est respectée.

En pratique, supprimer la relaxation du pneu sur le modèle de kart V2 a fait disparaître complètement les oscillations basses fréquences problématiques du volant.

L'origine exacte des oscillations dans le modèle de relaxation n'est pas totalement connue. Nous remarquons que le modèle de relaxation a une certaine fréquence :

$$\beta = \beta^*(1 - e^{2\pi ft})$$

mais théoriquement la réponse ne devrait être oscillante. Cependant, le modèle multicorps n'utilise pas l'équation analytique mais l'équation d'état 1.15 qui est intégrée à chaque pas de temps. Ce modèle inséré dans la boucle fermée du simulateur génère des oscillations.

3.3.3 Influence de la raideur du pneu

Cette section analyse l'influence des coefficients K et D du caoutchouc du pneu (équation 1.14 de la section 1.2.2.2) sur le couple ressenti au volant.

Les valeurs de K et D sont choisies telles qu'elles garantissent un enfoncement/amortissement réaliste du kart si il est lâché à une faible hauteur du sol (ex : 0.3 [m]).

La figure 3.27, où uniquement le PC est en fonctionnement, analyse le couple au volant avec :

- Une commande q imposée à l'articulation du volant $q(t) = A\sin(\frac{2\pi}{T}t)$
- Une vitesse du kart (modèle V2) $V = 7$ [m/s].
- Un $K = 100000$ [N/m] et un $D = 60000$ [Ns/m].

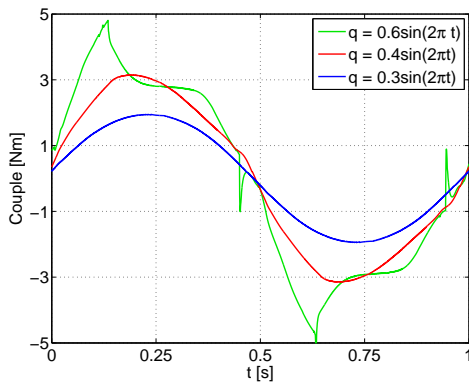


FIGURE 3.27 – Comparaison du couple pour $K = 10000$ [N/m] et $D = 4000$ [Ns/m] à $V = 7$ [m/s]

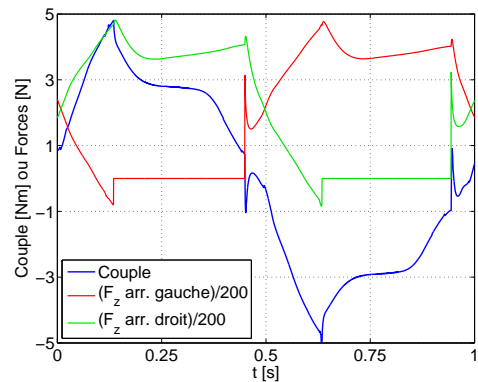


FIGURE 3.28 – Couple et forces normales pour $K = 10000$ [N/m] et $D = 4000$ [Ns/m] à $V = 7$ [m/s]

Pour des petits angles de braquage, le couple suit la forme sinusoidale que la position impose. Par contre, à plus grands angles de braquage, le signal de couple est parasité par des discontinuités qui apparaissent sur la partie ascendante et descendante du sinus.

Ces chutes de couple ont une origine physique. En effet, l'angle de chasse combiné au déport crée un mouvement vertical opposé sur les deux roues du train avant lors d'un braquage. La roue intérieure a tendance à rentrer dans le sol et l'extérieure à se soulever (elle ne se soulève pas car il y a aussi un transfert de charge arrière/avant). Par l'intermédiaire du châssis rigide, les roues arrière font le mouvement inverse : la roue arrière intérieure est délestée et l'extérieure chargée. La roue arrière intérieure se soulève même pour des grands angles de braquage. Cette perte puis reprise de contact est à l'origine des creux de couple comme montré en figure 3.28 via les forces normales. Un développement plus complet est présenté en [10].

Pour limiter l'effet de ces piques qui sont ressenties au volant et détériore l'expérience de l'utilisateur, K est réduit. Cela diminue la rigidité du kart complet et permet de plus grands angles de braquage sans pertes de contact.

3.3.4 Influence de la résolution du codeur

Le codeur a une résolution limitée au millième de radian (section 2.3.2). Afin d'analyser son influence sur le ressenti nous procédons en deux étapes :

Sur simulation, sinus q au millième

L'utilisateur est remplacé par une loi sinusoïdale sur la position $q(t) = A\sin(\omega t)$ (uniquement le PC est en fonctionnement).

De plus, le sinus imposé a la particularité de n'avoir une résolution que du millième. Mathématiquement, cela s'écrit :

$$q_{full}(t_i) = A\sin(\omega t_i)$$

$$q_{\%1000}(t_i) = \begin{cases} q_{full}(t_i) & \text{si } |q_{full}(t_i) - q_{t_i}| > 0.001 \text{ [rad]} \\ q_{\%1000}(t_{i-1}) & \text{sinon} \end{cases}$$

où $q(t_i) = q_{\%1000}(t_i)$ est l'angle de rotation de l'articulation du volant. La figure 3.29 montre le couple résultant d'un sinus à résolution maximale $q_{t_i} = q_{full}(t_i)$ comparé au sinus de résolution limitée au millième $q_{t_i} = q_{\%1000}(t_i)$ (décalé de +0.5 [s]).

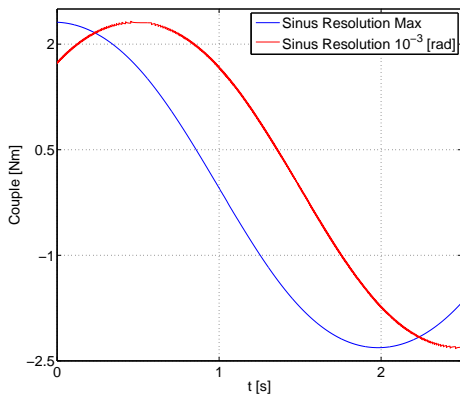


FIGURE 3.29 – Couple dans l'articulation du volant pour $q_{full} = 0.2\sin(\frac{2\pi}{4}t)$

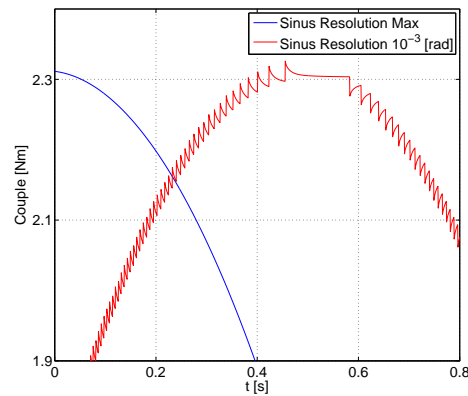


FIGURE 3.30 – ZOOM de la figure 3.29

L'effet de cette baisse de résolution est un caractère "dents de scie" du couple. Ces montées et descentes sont toutefois de faible amplitude comparativement à l'amplitude du sinus.

Émission du couple au volant

Dans cette seconde étape, nous utilisons à nouveau tout le simulateur complet.

Le couple "dents de scie" obtenu dans les conditions du paragraphe précédent est envoyé au volant. C'est uniquement un vecteur de couple qui est envoyé à chaque pas de temps (le modèle physique n'est donc plus utilisé dans cette deuxième expérience).

Après plusieurs expériences avec des sinus de fréquences et d'amplitudes différentes, nous concluons que ces parasites ne sont pas ressentis et que la résolution au millième n'affecte pas le rendu haptique.

3.4 Assemblage mécanique

Après avoir développé les améliorations informatiques de l'architecture en 2.5, des signaux en 3.2 et du modèle multicorps en 3.3, cette section décrit les améliorations mécaniques apportées

au simulateur.

A l'issue du mémoire précédent, l'assemblage mécanique du volant était fixé sur une table haute au moyen de deux serre-joints. L'écran de simulation était quant à lui placé sur une table légèrement plus haute permettant à l'utilisateur de conduire assis, avec la nuque droite. Le reste du matériel de simulation (PC, carte de développement) était disposé sur cette même table. La figure 3.31 montre le simulateur tel que vu par l'utilisateur.

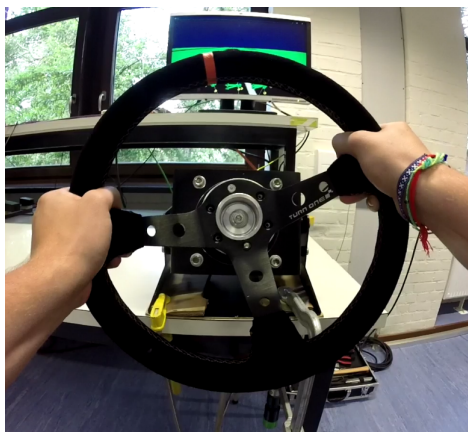


FIGURE 3.31 – Caméra embarquée sur le torse de l'utilisateur [11]

Pour ce travail, le simulateur a été placé dans une structure contenant un siège baquet (voir figure 3.32). Notons que la majeure partie de la structure porteuse avait déjà été montée par *Aubain Verlé* dans le cadre de sa thèse. De plus, l'insertion du simulateur a été réalisée avec sa collaboration.

La structure (1) est constituée d'un ensemble de profilés carrés *Bosch-Rexroth* en aluminium. Les quatre faces des profilés ont une rainure centrale où une tête de vis en T peut être glissée. Tous les profilés de l'assemblage sont solidarisés perpendiculairement deux à deux en :

1. Insérant une vis dans une rainure de chacune des deux barres.
2. Plaçant les taraudages d'une équerre en aluminium sur les vis.
3. Serrant l'équerre contre les barres au moyen d'une rondelle et d'un écrou.

Sur la partie inférieure du cadre, deux profilés horizontaux sont fixés. Les rails du siège baquet *Sabelt* (2) s'insèrent dans leur rainure. Un degré de liberté de translation selon X est permis en levant une poignée du siège (3).

Le pied de l'assemblage mécanique du volant (voir figure 2.5) est fixé sur un profilé transversal supérieur de la structure (4), toujours au moyen de deux vis en T (traversant les taraudages du pied) et de deux écrous/rondelles. Grâce au pivot du cadre du volant (voir figure 2.6), le volant est légèrement incliné selon Y vers l'écran.

Similairement, l'écran (5) est fixé à un profil vertical (6) via un connecteur *Bosh*. L'écran est :

- centré selon Y par rapport au siège.
- placé selon Z à la même hauteur que la tête d'un utilisateur de taille moyenne (1m75) assis dans le siège.
- placé selon X à une distance adéquate⁸ qui est modifiable en translatant le siège.

Le servo-variateur (7) et son interrupteur (8) de puissance sont eux aussi fixés à la structure. Enfin, le PC (9) et la carte de développement (10) reposent sur une table, placée à côté de la

8. distance conseillée par les centres de permis de conduire pratique.

structure.

En résumé, le simulateur dans sa structure confère une position de conduite réaliste à l'utilisateur.

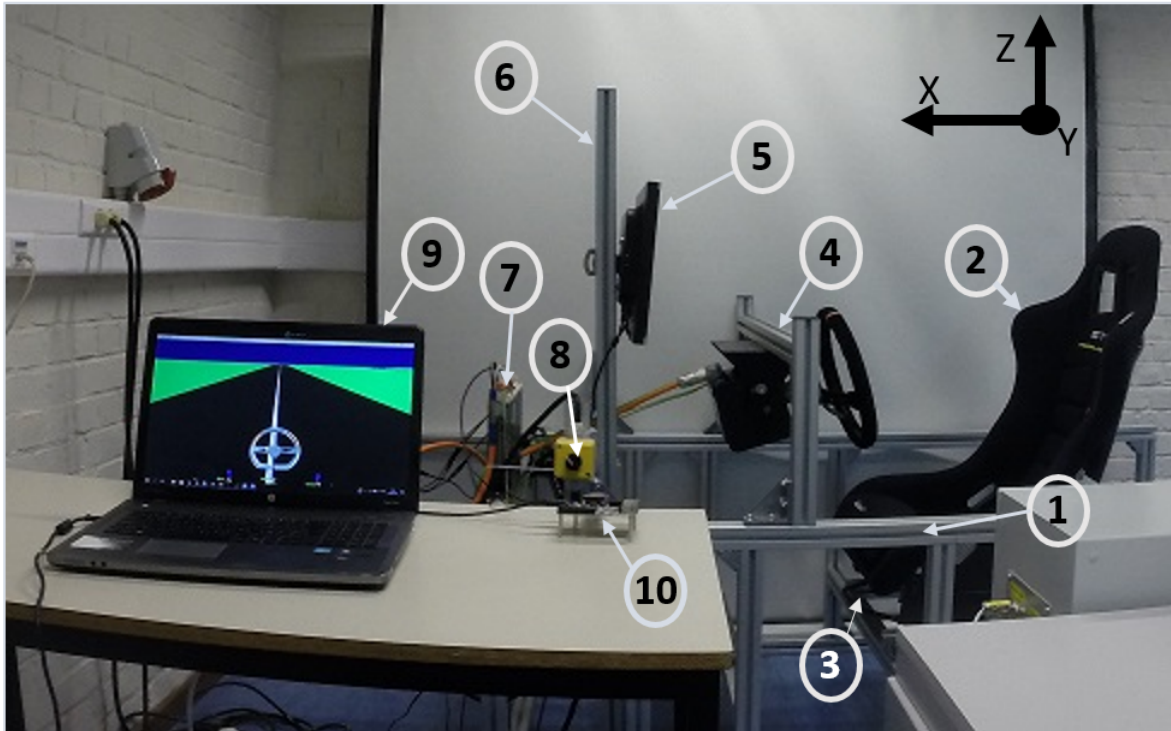


FIGURE 3.32 – Vue latérale du simulateur

3.5 Conclusion

Des améliorations informatiques et matérielles ont été présentées dans cette section.

Du point de vue informatique, le simulateur a une nouvelle architecture de contrôle : l'architecture avec interruptions. Cette architecture permet la transmission d'un couple au volant avec une fréquence de rafraîchissement plus grande que celle ressentie haptiquement par un humain.

Le couple est calculé en temps réel avec une grande précision. Ce couple provient de la dynamique d'un modèle multicorps réaliste d'un kart qui prend en entrée la position et la vitesse filtrée que l'utilisateur impose au volant.

Du point de vue matériel, le simulateur est constitué d'un volant mais aussi d'une structure avec siège baquet.

Grâce à toutes ces améliorations, le simulateur offre une expérience de conduite réaliste où aucune vibration parasite n'est ressentie. Le fonctionnement du simulateur est totalement validé.

Chapitre 4

Expériences sur simulateur

Le chapitre précédent a décrit le développement d'un outil, le simulateur de véhicule. A l'issue du chapitre, le fonctionnement du simulateur à retour haptique a été validé : il est donc exploitable dans le cadre d'une recherche scientifique.

Pour l'utilisation d'un simulateur validé, le fonctionnement n'est plus important et par abstraction, la figure 2.1 se simplifie (voir figure 4.1).

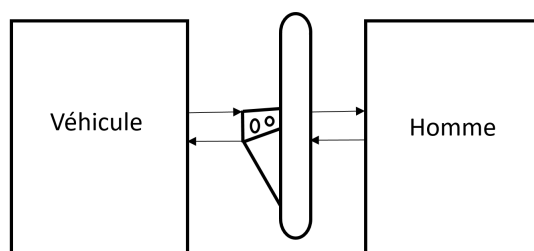


FIGURE 4.1 – Simulateur de véhicule à retour haptique

Ce chapitre décrit le type d'expériences ayant été réalisées sur le simulateur à retour haptique. Premièrement, les objectifs et les limites des expériences menées sont présentées. Deuxièmement, la procédure d'expériences séparée en deux parties est décrite.

4.1 Objectifs

L'objectif principal des expériences est d'étudier l'interaction Homme-volant grâce au simulateur. Elles étudient dans des conditions données, comment l'Homme utilise l'interface haptique Homme-machine, support de toute la dynamique du véhicule (voir figure 4.1).

Cet objectif est général et pourrait faire l'objet de plusieurs travaux à part entière. En effet, l'homme et le véhicule peuvent être étudiés dans des conditions différentes, sur base de critères différents,... Par exemple, le véhicule s'étudie dans :

- Des situations quasi-statiques.
- Des situation dynamiques.
- Des situations inattendues (réflexes).

Concernant le côté *Homme* de l'interface haptique, différentes influences pourraient être étudiées

- L'influence du type de conducteur.
- L'influence de l'état de fatigue (pour un même type ou plusieurs types de conducteur).
- L'influence de la direction du regard.

- L'influence de la position d'assise du pilote.
- L'influence de la distraction du pilote.

En raison des contraintes temporelles du travail, notre étude de l'interaction Homme-volant se limite *principalement* au suivi de trajectoire dans des conditions de faibles dynamiques, celui-ci mis en perspective avec le type de conducteur.

Pour que chaque utilisateur soit dans les mêmes conditions, nous adoptons une approche systématique : la procédure d'expériences. Chaque utilisateur/cobaye remplit un questionnaire et ensuite, suit une procédure stricte d'expériences au volant du simulateur.

4.2 Procédure

4.2.1 Questionnaire

Le but du questionnaire est de caractériser chaque utilisateur.

Un profil est défini à l'aide de trois critères généraux : l'identité (âge, sexe), les caractéristiques de conduite routière (nombre de km/année, ...) et l'intérêt pour les sports moteur (nombre de fois par an, ...).

Les questions proposées lors des journées d'expériences sont reprises ci-dessous.

Identité

- Age
 - Menu déroulant de l'année 1940 → 2010
- Sexe
 - Homme
 - Femme

Identité routière

- Type du permis du permis
 - A : Motocyclette
 - B : Voiture
 - C : Camion
 - Je ne possède pas de permis
- Année d'obtention du permis
 - Menu déroulant de l'année 1940 → 2010
- Utilisation du véhicule par année
 - <5000 km/an
 - <10000 km/an
 - <20000 km/an
 - <50000 km/an
 - <100000 km/an
 - >100000 km/an
- Type de voiture conduite¹
 - Mini citadine : Smart, 500, C1 ...

1. selon la segmentation automobile [26]

- Citadine polyvalente : A1, C2/3, Twingo/Clio, Fiesta, Polo, ...
 - Compacte : A3, C4, Série 1/2, Mégane, Focus, Astra, Golf, ...
 - Familiale : A4/5, C5, Série 3, Passat, ...
 - Intermédiaire ou routière : A6, C6, Série 4/5, Classe E, ...
 - Berline de luxe : A7, Série 6/7, Classe S, ...
 - SUV/4x4 : Q5, X5, Land Rover, ...
 - Monospace : Renault Espace, Galaxy, Sharan, ...
- Type de transmission préféré
 - Aucune préférence
 - Traction
 - Propulsion
 - 4 roues motrices
 - Style de conduite
 - Ecoresponsable
 - Douce
 - Intermédiaire
 - Sportive
 - Très Sportive

Sport Moteur

- Intérêt pour le sport Moteur
 - Complètement désintéressé(e)
 - Désintéressé(e)
 - Curieux(se)
 - Intéressé(e)
 - Passionné(e)
- Pratique d'un sport Moteur
 - Jamais
 - Annuellement
 - Trimestriellement
 - Mensuellement
 - Hebdomadairement
- Type du sport Moteur
 - Réponse courte

Quelques remarques peuvent être faites sur ce questionnaire

Remarque 1 Le questionnaire n'a pas vocation à être exhaustif. Il ne définit pas le profil absolu du conducteur, mais un ensemble de caractéristiques générales des utilisateurs qui peuvent chacune être liées/corrélées avec les résultats des expériences de suivi de trajectoire.

De nombreux autres paramètres pourraient caractériser le profil d'un utilisateur. Le nombre d'accidents en tort par année, le respect des limites de vitesse, le type de boîte à vitesses utilisé, ... sont autant de paramètres qui caractérisent aussi un conducteur.

Remarque 2 Certaines questions sont sujettes à l'interprétation de l'utilisateur (le type de transmission préférée, l'intérêt pour le sport moteur et le style de conduite). Pour l'étude de

suivi de trajectoire, ce taux propre de subjectivité est ignoré car difficilement quantifiable.

Remarque 3 L'ordre d'apparition des possibilités de réponses est choisi à dessein. Les propositions commencent à un niveau de "faible d'intensité" jusqu'au niveau de "haute intensité". Par exemple, le type de transmission part de la préférence d'un conducteur lambda à celle d'un conducteur avide de sensations sur la route. Le type de voiture conduite s'étend d'une petite voiture à la voiture la plus grande,... Cet ordre croissant est choisi pour faciliter l'analyse des résultats du chapitre suivant.

Remarque 4 Chacun des critères modélise un ou plusieurs paramètres différents du profil du conducteur :

- L'âge mesure des paramètres physiologiques de l'utilisateur. L'année d'obtention du permis mesure l'expérience sur la route tandis que l'utilisation du véhicule par année modélise la familiarisation actuelle de l'utilisateur avec la conduite d'un véhicule routier.
- Le type de voiture conduite peut par exemple démontrer si un utilisateur conduisant une routière a plus de capacité à suivre une trajectoire qu'un utilisateur conduisant des voitures "passe-partout".
- Le style de conduite mesure si la relative nervosité du conducteur sur la route influence son expérience de conduite sur le simulateur. L'influence de la transmission préférée mesure si le plaisir de conduite sur la route influence le suivi trajectoire.
- L'intérêt pour le sport moteur mesure l'influence des connaissances, de la passion sur le suivi de trajectoire. La pratique du sport moteur mesure quant à elle l'avantage qu'aurait un utilisateur habitué à rouler à d'autres fins que des fins utiles (meilleure connaissance des points de braquage, point de fixation pendant le braquage, de la trajectoire idéale, ...). Le type de sport moteur peut par exemple montrer si un utilisateur qui fait régulièrement du kart a plus de capacités qu'un utilisateur qui roule en quad.

4.2.2 Expériences

4.2.2.1 Conditions d'expérience

Conditions de simulation Les conditions de simulation sont celles décrites en section 3.5. Le modèle multicorps *Robotran* que l'utilisateur pilote est le kart détaillé en section 1.2.1. Les forces de contact pneu/sol suivent le modèle de Bakker (sans relaxation) et le kart est maintenu à vitesse constante grâce au régulateur PID de la section 1.2.2.1.

Conditions pratiques L'utilisateur est assis sur le siège du simulateur tel que montré en 3.32. Systématiquement, il est proposé à l'utilisateur d'adopter une position d'assise similaire à celle adoptée dans son véhicule. Cela limite les erreurs de trajectoire dues à un inconfort ou à une mauvaise vision de l'écran situé en face de lui.

Le déroulement de la simulation est communiqué avant son exécution mais pas son objet. Durant la simulation, un minimum d'échange verbal intervient afin de ne pas déconcentrer l'utilisateur. Certaines conditions sont discutées dans la section 4.3.

4.2.2.2 Familiarisation

Pour l'expérience de familiarisation, un circuit a été dessiné sur SolidWorks à partir de formes simples : des lignes droites ou des arcs de cercle (voir figure 4.2).

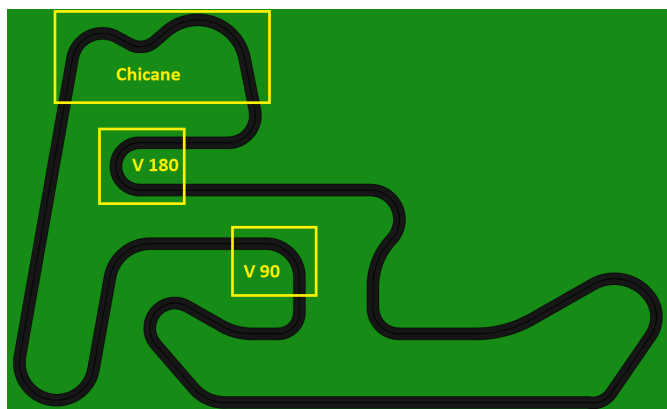


FIGURE 4.2 – Circuit dessiné sur SolidWorks®

Ce circuit est une réplique du *Circuit International CIK Genk* (1.3 [km] de périmètre). Les dimensions générales sont approximativement respectées en utilisant une illustration à l'échelle [25]. Une ligne blanche de 0.08 [m] de large est aussi placée au centre de la route de 6 [m] de large.

L'utilisateur fait un tour de ce circuit (≈ 140 [s]) et son but est de suivre la ligne blanche, au milieu. L'utilisateur juge lui-même où il repère le milieu du kart (voir figure 4.3).

L'objectif de cette première expérience est donc de familiariser l'utilisateur au simulateur. L'utilisateur s'accommode au retour haptique, à la sensibilité du kart ainsi qu'à l'environnement de visualisation *Java 3D*.

4.2.2.3 Réactivité

Pour cette deuxième expérience, l'utilisateur ne dirige plus le kart et le kart ne lui fournit plus de couple : l'écran pourrait donc être éteint. A la place, un profil de couple dépendant du temps est transmis au volant.

Au début de l'expérience, aucun couple n'est transmis dans le volant (voir figure 4.4). En $t = 6$ [s], un couple de 4 [Nm] est envoyé au volant et en $t = 13$ [s], le couple retourne à 0 [Nm]. La simulation se termine en $t = 15$ [s].

Le but de l'utilisateur expérimentant ce profil est de maintenir le volant droit durant toute la simulation (ruban rouge à sa position la plus haute, voir figure 3.31).

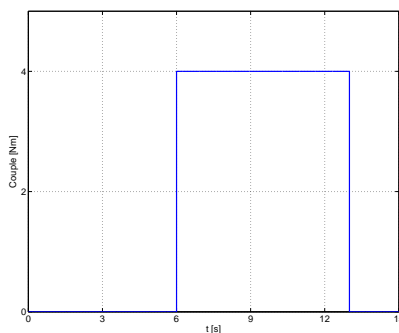


FIGURE 4.4 – Évolution temporelle du couple imposé au volant

Par sécurité, l'utilisateur est informé qu'il va recevoir des échelons de couple mais le moment auquel ils surviennent n'est pas communiqué.

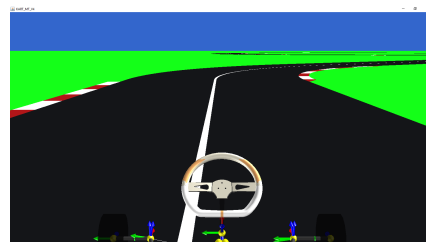


FIGURE 4.3 – Visualisation d'une simulation

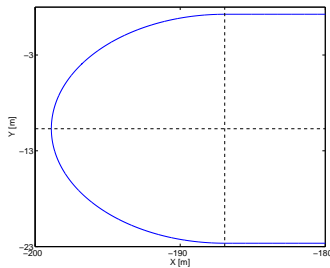
L'objectif de cette expérience est donc de caractériser la réponse de l'utilisateur, sa réactivité à un couple auquel il s'attend.

Cette réponse est mesurée à l'aide un facteur d'amortissement et d'une fréquence des oscillations de la vitesse du volant, propre à chaque utilisateur. L'algorithme pour les calculer ainsi que les résultats sont expliqués en section 5.3.1.1 et 5.3.2.

Comme pour toutes les expériences décrites à la section suivante, chaque simulation est faite deux fois pour pouvoir moyenner les résultats et ainsi réduire les facteurs parasites (effet de surprise, ...) propres à une expérience isolée.

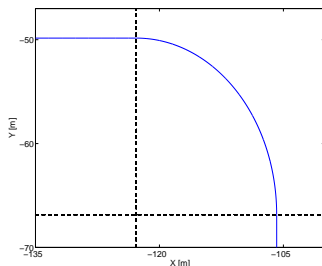
4.2.2.4 Prise de virages

Pour ce troisième type d'expériences, le circuit de la phase de familiarisation est réutilisé. Trois virages sont isolés (voir figure 4.2) et servent chacun pour trois simulations différentes. L'axe de symétrie des virages ainsi que leurs caractéristiques principales sont présentés ci-dessous.



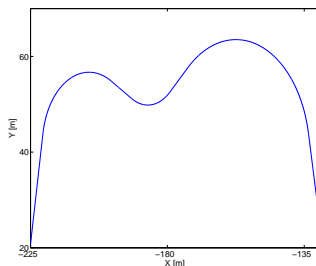
Virage à 180°

- Temps de simulation : $t = 17$ [s]
- Vitesse du kart : $V = 5$ [m/s]
- $a_g = \frac{V}{Rg} = 0.213$



Virage à 90°

- Temps de simulation : $t = 10.5$ [s]
- Vitesse du kart : $V = 8.5$ [m/s]
- $a_g = \frac{V}{Rg} = 0.433$



Chicane

- Temps de simulation : $t = 24.5$ [s]
- Vitesse du kart : $V = 7$ [m/s]
- $a_g = \frac{V}{Rg} = [0.325; 0.452; 0.206]$

Avant la prise des virages, il est rappelé le but de l'utilisateur : suivre la ligne centrale. Le sens des tournants lui est aussi indiqué.

L'objectif de cette expérience est de mesurer la capacité de l'utilisateur à suivre une trajectoire imposée. Cela se mesure grâce à la position du véhicule mais d'autres caractéristiques de la trajectoire sont aussi enregistrées (vitesse et accélération du véhicule, position, vitesse et accélération du lacet du véhicule, ...).

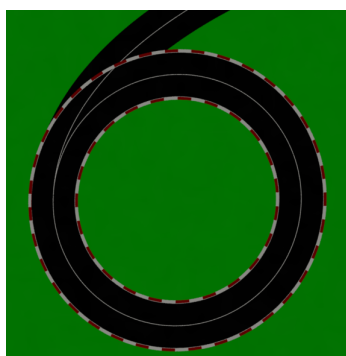
Contrairement aux virages à 180° et 90°, la prise du tournant de type chicane est simulée quatre fois. Les deux premières dans les conditions explicitées plus haut et le deux dernières sans

retour de couple. Cela est simulé afin de (in)valider l'effet facilitateur du retour haptique sur le suivi de trajectoire.

L'ensemble des analyses est exposé à la section 5.3.2 du chapitre suivant.

4.2.2.5 Prises d'un rond-point

Pour ce quatrième et dernier type d'expériences, nous simulons la prise d'un rond-point. La prise de ce virage est facilitée par une rampe de lancement en quart d'ellipse. Elle diminue progressivement le rayon de braquage (voir figure 4.5). Les conditions du rond-point sont décrites ci-dessous.



Rond-point

- Temps de simulation : $t = 28.5$ [s]
- Vitesse du kart : $V = 10$ [m/s]
- $a_g = \frac{V}{Rg} = 0.596$

FIGURE 4.5 – Rond-point sous SolidWorks®

La particularité de cette expérience est que nous étudions l'influence d'un paramètre dynamique sur le suivi de trajectoire.

Lorsque l'utilisateur conduit à $\theta_{braq} \approx cste$ sur le rond-point, nous réduisons instantanément de 30% les forces latérales des roues arrières :

$$F_{lat,arrG} = F_{lat,arrG} \cdot 0.7 \quad F_{lat,arrD} = F_{lat,arrD} \cdot 0.7 \quad \text{pour } t : 17 \rightarrow 28.5 \text{ [s]}$$

Le kart a alors tendance à tourner d'avantage sans modifier la commande au volant.

Ceci peut s'expliquer intuitivement. Avant la perturbation, les F_{lat} sont réparties de façon à maintenir le véhicule à un rayon de braquage $\theta_{braq} \approx cste$. Les accélérations centripètes des roues sont donc également "balancées". Rappelons que l'accélération centripète est l'accélération qui maintient le véhicule sur sa trajectoire. Donc, si $F_{lat,arr} \downarrow, a_{arr} \downarrow$ et le kart est moins "retenu" sur sa trajectoire à l'arrière qu'à l'avant. Il "pivote" et rentre vers l'intérieur du rond-point.

Ce type de comportement est dit sur-vireur contrairement au sous-virage où c'est le train avant qui "tire-tout droit".

Avant la prise du rond-point, il est de nouveau rappelé à l'utilisateur de suivre la ligne centrale durant toute l'expérience. Il lui est aussi indiqué qu'un évènement particulier va se produire pendant la simulation, sans plus de détails.

Cette expérience est réalisée trois fois.

4.3 Hypothèses

Nous posons ici certaines hypothèses simplificatrices sur l'influence de conditions de simulations qui échappent à notre contrôle ou que nous avons ignoré par simplicité.

- *Conditions extérieures de l'expérience* : les facteurs extérieurs à la simulation ne perturbent pas la concentration de l'utilisateur (bruit, non-confinement du simulateur, ...).
- *Vue périphérique* : le manque de vue périphérique n'influence pas, ou influence avec la même intensité tous les utilisateurs.
- *Trajectoire de référence* : la ligne centrale que l'utilisateur doit suivre est supposée sans épaisseur.

Chapitre 5

Résultats

Ce chapitre analyse les résultats de la procédure d'expériences expliquées au chapitre 4. Premièrement, ce chapitre décrit la population que forment les 38 utilisateurs qui ont répondu au questionnaire. Deuxièmement, il étudie leur capacité à suivre une trajectoire donnée en fonction de leur profil.

5.1 Outil d'analyse

Avant d'analyser la population d'utilisateurs, nous introduisons brièvement un outil statistique : le coefficient de corrélation.

Nous définissons uniquement le concept, sans preuves mathématiques. Des justifications complètes sont présentées en [4].

Le coefficient de corrélation Le coefficient de corrélation de deux variables Y_1 et Y_2 mesure l'intensité de leur dépendance linéaire. Il est donné par :

$$\rho = \frac{Cov(Y_1, Y_2)}{\sigma_1 \sigma_2} \quad -1 < \rho < 1 \quad (5.1)$$

où σ_1, σ_2 sont les écarts types. Ce coefficient est donc la normalisation de la covariance, grandeur qui évalue l'écart conjoint de variables par rapport à leur moyenne, pour N observations de chaque variable Y :

$$Cov(Y_1, Y_2) = \frac{1}{N-1} \sum_{i=1}^N (y_{1i} - \bar{y}_1)(y_{2i} - \bar{y}_2) \quad (5.2)$$

avec la notation \bar{y}_1 pour définir la moyenne sur l'ensemble des données de la variable Y_1 .

Trois cas sont possibles :

- $\rho > 0$: La relation entre Y_1 et Y_2 est croissante et d'autant plus linéaire que $\rho \rightarrow 1$.
- $\rho = 0$: Y_1 et Y_2 sont indépendants
- $\rho < 0$: La relation entre Y_1 et Y_2 est décroissante et d'autant plus linéaire que $\rho \rightarrow -1$.

L'échelle de corrélation la plus populaire est celle définie par Jacob Cohen [5] : 0.1 \rightarrow 0.3 : corrélation faible, 0.3 \rightarrow 0.5 corrélation modérée, 0.5 \rightarrow 1 : forte corrélation. De nombreuses autres classifications existent en fonction de l'application. Au vu des coefficients calculés dans le cadre des expériences sur le simulateur, l'échelle d'intensité de Cohen est sensiblement raffinée :

| $ \rho $ | Intensité |
|-----------|---------------------------|
| 0.0 → 0.1 | Très Faible ou Inexistant |
| 0.1 → 0.3 | Faible |
| 0.3 → 0.5 | Modérée |
| 0.5 → 0.7 | Forte |
| 0.7 → 1 | Très Forte |

TABLE 5.1 – Échelle d'intensité de corrélation

5.2 Questionnaire

Cette section rassemble et commente les réponses des 38 utilisateurs au questionnaire (description en section 4.2.1).

Elle permet d'avoir un esprit critique quant à la validité des analyses présentées à la section suivante.

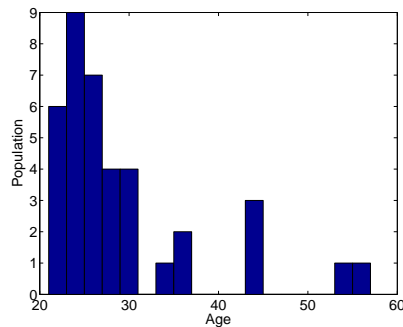


FIGURE 5.1 – Age

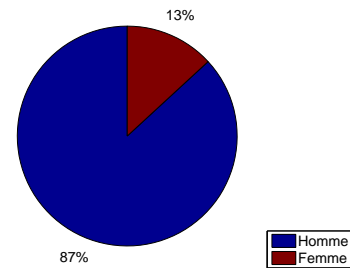


FIGURE 5.2 – Sexe

L'histogramme en figure 5.1 montre que la majorité des participants ont entre 20 et 30 ans : l'âge moyen est de 29 ans. Quelques quadragénaires et quinquagénaires ont aussi participé et le range s'étend de 21 → 57 ans.

La figure 5.2 illustre le faible taux de participation des femmes : seules 5 femmes ont participé.

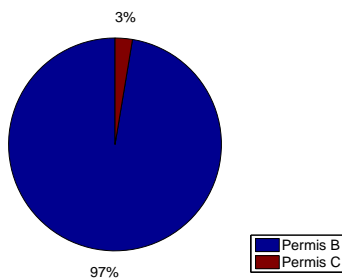


FIGURE 5.3 – Type de permis

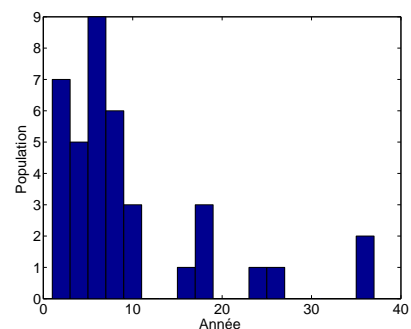


FIGURE 5.4 – Nombre d'années de permis (AnP)

En figure 5.3, le diagramme circulaire montre qu'un seul utilisateur possède le permis camion, le reste de la population est titulaire du permis voiture. La figure 5.4 illustre le nombre d'années depuis l'obtention du permis de conduire. Nous voyons que l'histogramme en figure 5.1 lui est très similaire. Ce point est étudié à la section suivante.

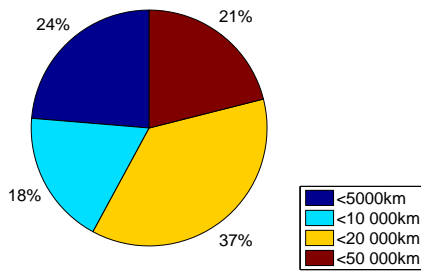


FIGURE 5.5 – Nombre de km/an

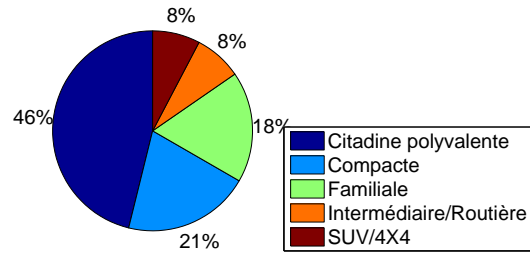


FIGURE 5.6 – Type de voiture conduite

Le diagramme circulaire de la figure 5.5 est plus équilibré. La majorité des utilisateurs roule plus de 10000 [km] par an mais aucun plus de 100000 [km]. La population est *relativement* habituée à conduire un véhicule.

Les résultats du diagramme en figure 5.6 montrent que la plupart des utilisateurs roule dans des voitures conçues pour tout type de circulation bien que plus adaptées pour les villes (compacte ou citadine polyvalente). De plus, les modalités pratiques (circulation facile en ville, prix, nombre de places) sont plus représentées que le confort ou le plaisir de conduite (8% de routières et aucune berline de luxe).

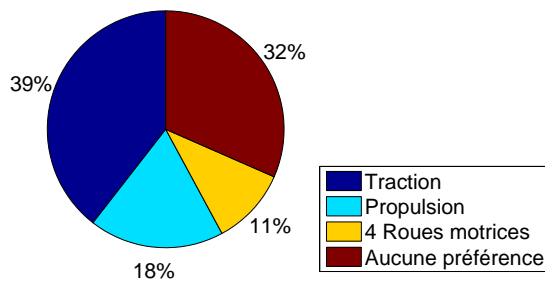


FIGURE 5.7 – Type de transmission préféré (TTr)

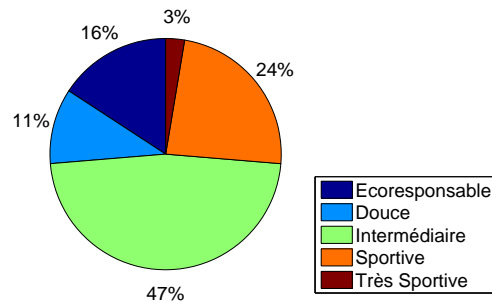


FIGURE 5.8 – Style de conduite (StC)

De la figure 5.7, il apparait que la large majorité des utilisateurs préfère les tractions ou n'a aucune préférence. Ce type de transmission est souvent préféré pour sa plus grande sécurité par rapport à la propulsion.

A la vue de la figure 5.8, le style de conduite est jugé intermédiaire par le plus grand pourcentage des utilisateurs. Seule une personne juge sa conduite très sportive.

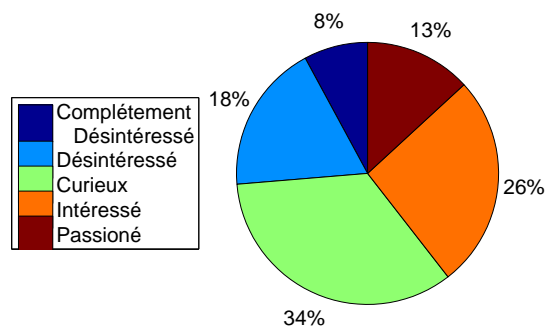


FIGURE 5.9 – Intérêt pour le sport moteur (ISPM)

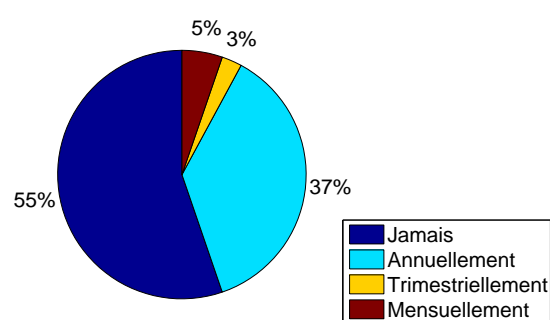


FIGURE 5.10 – Pratique d'un sport moteur (PSPM)

Le diagramme de l'intérêt pour le sport moteur est séparé de façon bien plus égalitaire que

celui lié à sa pratique (qui est majoritairement réalisée à très faible fréquence).

Le diagramme du type de sport moteur n'est pas montré car il est redondant avec celui de sa pratique. Les utilisateurs pratiquant un sport moteur font tous du karting.

Indépendance des questions

Afin de vérifier si les questions modélisent un facteur différent du profil du conducteur, les coefficients de corrélation entre questions sont calculés.

Comme mentionné, les différentes tranches d'âge et tranches du nombre d'années de permis ont approximativement la même population. Cela se vérifie avec $\rho(\text{Age}, \text{Année de Permis}) = 0.96$. Il est au niveau de *très forte* corrélation (voir table 5.1). Ainsi, les analyses du suivi de trajectoire faites avec l'un ou l'autre critère donnent des facteurs ρ similaires.

Il existe aussi une corrélation *forte* ($\rho = 0.57$) entre l'intérêt et la pratique d'un sport moteur (SPM, par la suite). Au vu des données, les utilisateurs qui portent plus d'intérêt ont *tendance* à plus en pratiquer.

Quelques coefficients de corrélation *modérés* existent entre d'autres questions :

- $\rho(\text{Age}, \text{Pratique SPM}) = 0.35$: les utilisateurs plus âgés font, de façon *modérée*, plus de sport moteur.
- $\rho(\text{Style de conduite}, \text{Intérêt SPM}) = 0.41$: les utilisateurs qui jugent qu'ils ont un comportement sportif sur la route ont tendance à porter plus d'intérêt pour les sports moteurs.
- $\rho(\text{Intérêt SPM}, \text{Type de Transmission préférée}) = 0.45$: les conducteurs préférant une transmission propulsive ou à quatre roues motrices ont tendance à préférer les sport moteur que ceux privilégiant les tractions ou n'ayant aucune préférence.

Le reste des critères est totalement ou très fortement décorrélés.

| ρ | $\rho(\text{Age}, \text{AnP})$ | $\rho(\text{ISPM}, \text{PSPM})$ | $\rho(\text{ISPM}, \text{TTr})$ | $\rho(\text{StC}, \text{ISPM})$ | $\rho(\text{Age}, \text{PSPM})$ |
|--------|--------------------------------|----------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Valeur | 0.96 | 0.57 | 0.45 | 0.41 | 0.35 |

TABLE 5.2 – Coefficient de corrélation entre questions

En conclusion, la plupart des questions ne modélise pas totalement une même caractéristique du profil de l'utilisateur. Il est donc justifiable de les analyser chacun avec les erreurs de suivi de trajectoire en section suivante.

5.3 Expériences

5.3.1 Algorithme

5.3.1.1 Réactivité

Comme mentionné en 4.2.2.3, le but de l'expérience de réactivité est de mesurer la réponse de l'utilisateur suite à un échelon de couple.

La réponse est calculée à partir de la vitesse du volant \dot{q} qui, une fois le couple transmis ($t = 6$ [s]), subit des oscillations (généralement centrées autour de zéro) et ensuite s'annule (contrairement à q).

Ce signal est dès lors caractérisé par :

- Une fréquence : f
- Un taux d'amortissement : D

Ils sont calculés à l'aide d'un algorithme cherchant les maxima/minima de \dot{q} . La méthode doit être valable pour des signaux de fréquences très différentes ($f = 2 \rightarrow 13$ [Hz]) où la quantité de bruit et de pics parasites est fonction de l'utilisateur.

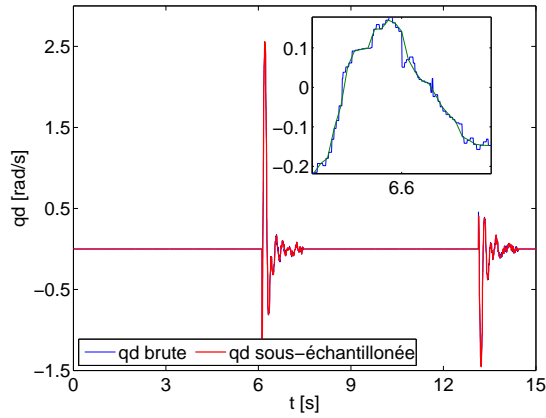


FIGURE 5.11 – Premier traitement de la vitesse du volant

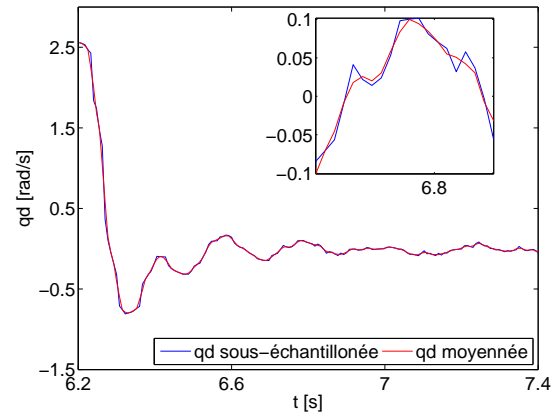


FIGURE 5.12 – Deuxième traitement de la vitesse du volant

La figure 5.11 montre \dot{q} tel qu'enregistrée au cours d'une expérience d'un des 38 utilisateurs. Afin de limiter les oscillations hautes fréquences, le signal est sous-échantillonné.

La figure 5.12 met en évidence la réponse signal suite à un échelon de couple positif. Il est isolé du signal complet (figure 5.11) puis traité grâce à une moyenne glissante. Chaque point est remplacé comme par moyenne de ce point et de ces deux voisins.

Cela réduit l'amplitude des pics parasites du signal tout en lui permettant de garder sa forme, comparativement au sous-échantillonnage.

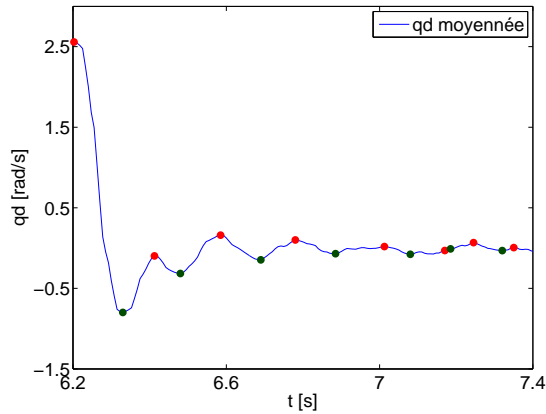


FIGURE 5.13 – Maximas et minima sur la vitesse en moyenne glissante

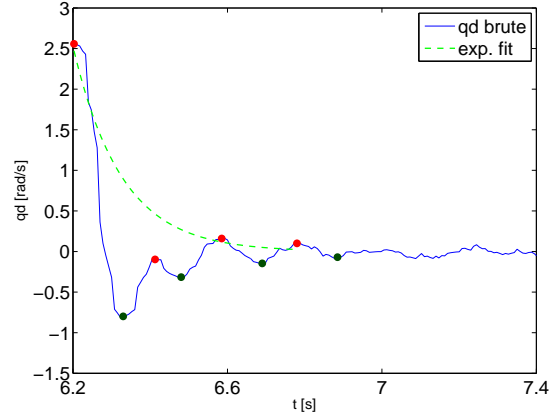


FIGURE 5.14 – Traitement des maxima et minima

La figure 5.13 illustre l'algorithme de recherche de maxima/minima appliqué au signal moyenné. Le critère qui définit un maxima est :

$$MAX = \max(\dot{q}_{i:i+N,moy}) \quad si \quad \dot{q}_{i,moy} > moy(\dot{q}_{i:i+N,moy})$$

Graphiquement, à chaque point \dot{q}_i est associé un intervalle de N points qui le suivent $\dot{q}_{i:i+N,moy}$ où l'indice moy désigne les points après le traitement par moyenne glissante. Si la vitesse $\dot{q}_{i,moy}$ est plus grande que la moyenne des vitesses de cet intervalle $moy(\dot{q}_{i:i+N,moy})$, l'intervalle contient un maxima. Une fois un maxima trouvé, l'algorithme cherche un minima de façon similaire et ainsi de suite. La taille de l'intervalle N est le degré de liberté et est définie expérimentalement pour que peu de pics parasites soient retenus.

Après les itérations sur tous les points du signal, certains maxima/minima non-représentatifs des mouvements de l'utilisateur sont trouvés. Un algorithme de traitement de maxima est utilisé. Les maxima/minima sont considérés non-valides si :

$$|MAX_i - MIN_i| < 0.1 \text{ [rad]} \quad \&\& \quad |MAX_{i+1} - MIN_i| < 0.1 \text{ [rad]}$$

ou

$$|MIN_i - MAX_i| < 0.1 \text{ [rad]} \quad \&\& \quad |MIN_{i+1} - MAX_i| < 0.1 \text{ [rad]}$$

où le premier sommet de \dot{q} est toujours un maxima ($t_{MAX_1} - t_{MIN_1} < 0$) et où 0.1 est fixé expérimentalement. Le résultat de cette sélection est montré en figure 5.14.

La fréquence est alors aisément déduite : $f = \frac{\#(MAX)}{t_{MAX_{last}} - t_{MAX_1}}$.

Pour trouver le taux d'amortissement D , une exponentielle de type ae^{Dt} est fittée sur les maxima (choix arbitraire). Elle est fittée de façon à minimiser les résidus de vitesse, au sens des moindres carrés.

Dans ce but, une méthode d'optimisation non-linéaire (Nelder-Mead) est utilisée en partant des conditions initiales a_0, D_0 d'une exponentielle passant par le premier et le dernier maxima.

5.3.1.2 Erreur de suivi de trajectoire

Pour le calcul d'erreur de suivi de trajectoire, l'axe/courbe de symétrie d'un virage est prise comme référence.

L'algorithme du calcul d'erreur de trajectoire est décomposé en deux étapes :

1. Séparation des lignes droites et des courbes (trajectoire faite de formes simples uniquement).
2. Calcul d'erreur moyenne sur chaque partie.

Pour l'illustrer, le cas simple du virage à 90° sert d'exemple.

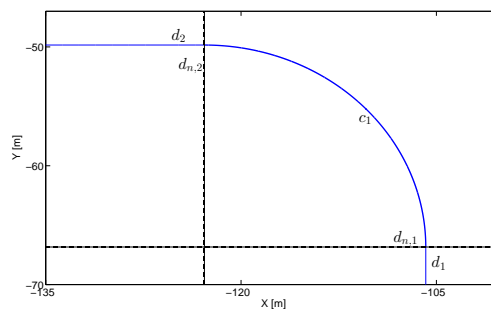


FIGURE 5.15 – Virage à 90°

Les lignes droites d_1, d_2 sont séparées de la courbe c_1 en utilisant deux droites normales $d_{n,1}, d_{n,2}$. Elles intersectent chacune un point d'intersection droite-courbe de la trajectoire (voir figure 5.15).

Cette segmentation définit, pour chacune des positions successives du véhicule¹, la référence (d_1 ou c_1 ou d_2) qui doit être utilisée pour calculer l'erreur de trajectoire. Par exemple, si le kart est à une distance plus petite que zéro de d_1 , il calcule e_{d_1} sa plus courte distance par rapport à la

1. le véhicule est représenté par le milieu de sa voie avant

ligne droite d_1 , sinon il sélectionne la courbe c_1 ou la ligne droite d_2 . C'est trivial pour le virage à 90° mais l'est moins pour la chicane et le rond-point.

La plus courte distance de séparation entre le véhicule et la droite $d_{n,1}$, mais aussi entre un point quelconque $(x(t_j), y(t_j))$ et une droite quelconque d'équation $ax + by + k = 0$ est donnée par :

$$d_{P,d}(t_j) = e_{inst,d}(t_j) = \frac{ax(t_j) + by(t_j) + f}{\sqrt{a^2 + b^2}} \quad (5.3)$$

qui est donc aussi la formule utilisée pour calculer l'erreur totale de trajectoire en ligne droite $E_{t,d_i} = \sum_{j=1}^{N_{d_i}} e_{inst,d_i}(t_j)$ pour une ligne droite d_i où N_{d_i} points de la trajectoire du véhicule sont enregistrés. Elle est définie positive si le véhicule est à l'intérieur de la courbe suivant la ligne droite d_i .

En courbe, l'erreur de trajectoire du véhicule est calculée comme :

$$d_{P,c}(t_j) = e_{inst,c}(t_j) = \sqrt{(x(t_j) - l)^2 + (y(t_j) - k)^2} - R \quad (5.4)$$

qui donne pour une courbe c_i : $E_{t,c_i} = \sum_{j=1}^{N_{c_i}} e_{inst,c_i}(t_j)$. Elle est définie positive si le véhicule est à l'intérieur de la courbe c_i .

L'erreur entière sur un type de virage (ligne droite + virage) se calcule comme :

$$E_t = \sum_{i=1}^{M_d} E_{t,d_i} + \sum_{i=1}^{M_c} E_{t,c_i} \quad (5.5)$$

pour un virage formé de M_d ligne droite et M_c courbes.

Au cours des expériences, la position du véhicule $(x(t_j), y(t_j))$ s'enregistre à une fréquence constante. L'erreur moyenne sur un virage complet se calcule simplement comme $e_t = E_t/N$ où $N = \sum_{i=1}^{M_d} N_{d_i} + \sum_{i=1}^{M_c} N_{c_i}$ est le nombre total des points de position du véhicule. L'erreur est positive si l'utilisateur est en moyenne à l'extérieur de la trajectoire imposée et négative dans le cas contraire.

Nous définissons quelques variantes de l'erreur e_t :

| Symbole | Formule | Description |
|------------------|---|--|
| e_{t-exp1} | E_t/N | L'erreur moyenne signée de la première expérience du virage étudié |
| $e_{t,abs-exp1}$ | $(\sum_{i=1}^{M_d} \sum_{j=1}^{N_i} e_{inst,d_i}(t_j) + \sum_{i=1}^{M_c} \sum_{j=1}^{N_i} e_{inst,c_i}(t_j))/N$ | L'erreur absolue moyennée de la première expérience du virage étudié |
| $e_{t,moy}$ | $\frac{e_{t-exp1} + e_{t-exp2}}{2}$ | La moyenne des erreurs moyennes signées des 2 expériences du type de virage étudié |

TABLE 5.3 – Notation des erreurs de trajectoire

Pour rappel, le virage à 180° et virage à 90° (abréviation : 180 - 90) sont pris deux fois et le virage de type chicane quatre fois (abréviation : *ch*).

5.3.2 Analyse

Deux points de l'interaction homme-véhicule sont étudiés dans cette section. Premièrement, nous analysons la réactivité de l'utilisateur et deuxièmement nous détaillons le suivi de trajectoire.

5.3.2.1 Réactivité

Au cours de cette section, nous analysons uniquement la partie de la vitesse du volant propre à l'échelon de couple positif.

La première analyse qui peut être faite concerne la variabilité des D et f . Les amortissements D varient au maximum d'un ordre de grandeur entre les utilisateurs. Les amortissements varient aussi entre la première et la deuxième expérience : un maximum d'un facteur ≈ 4 de variation est observé. Les fréquences f varient de $2 \rightarrow 13$ [Hz] entre les utilisateurs et varient aussi entre les expériences : un maximum d'un facteur ≈ 2 de variation est observé. Une hypothèse de cette variabilité est que la première expérience mesure une réponse plus naturelle que la deuxième où l'utilisateur anticipe l'échelon de couple, en contractant ses muscles. De plus, l'utilisateur se caractérise par une unique fréquence mais les oscillations de la vitesse ont des fréquences variables (voir figure 5.17).

La deuxième analyse concerne la forme du signal de réponse de l'utilisateur. Certains utilisateurs ont des réponses dont les maxima ne forment pas une exponentielle décroissante. Par exemple, certains ont des flancs montants ou descendants entrecoupés respectivement de petits flancs descendants ou montants (voir figure 5.16). D'autres ont des oscillations qui s'atténuent puis s'amplifient à nouveau (voir figure 5.17). Certains autres signaux sont oscillants pendant un temps beaucoup plus grand que le temps où ils se stabilisent,...

Par conséquent, dans certains cas l'algorithme de recherche de maxima tel que présenté n'est pas assez sélectif et dans d'autre cas, il est trop sélectif. La convergence de la méthode de minimisation des résidus utilisée pour calculer les paramètres de l'exponentielle de fit dépend elle aussi de la forme de la réponse.

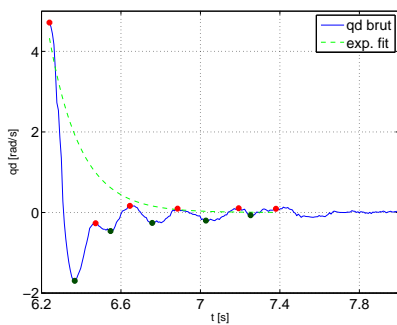


FIGURE 5.16 – Réponse en vitesse d'un utilisateur

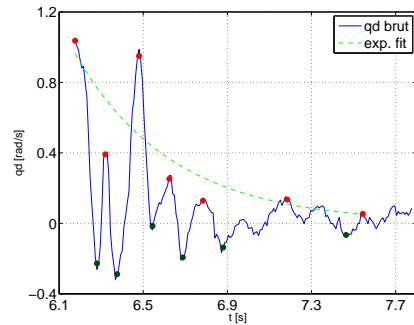


FIGURE 5.17 – Réponse en vitesse d'un utilisateur

Conscient des limites de la validité de la caractérisation des utilisateurs par un couple D/f et des limites de l'algorithme utilisé, deux types de coefficients de corrélation ρ peuvent être analysés.

Les ρ de corrélation de la fréquence/amortissement avec les critères du profil sont tous faibles ou très faibles. Aucune tendance n'est donc trouvée entre cette expérience et le questionnaire. Cette constatation est valable pour les deux expériences de réactivité.

Les ρ de corrélation de la fréquence/amortissement avec les erreurs de suivi de trajectoire sont aussi tous faibles ou très faibles. Ces erreurs de trajectoire sont analysées à la section suivante.

5.3.2.2 Erreur de suivi de trajectoire

Cette première partie de la section analyse les corrélations entre les erreurs de trajectoire et les réponses apportées au questionnaire qui définissent le profil des conducteurs.

Virage à 180°

Le tableau 5.4 présente les coefficients de corrélation du virage à 180°.

Un $\rho > 0$ modélise une situation où le critère du questionnaire "croît" avec l'erreur. Le critère augmente en intensité (Style de conduite), fréquence (km/an),... selon l'ordre défini dans la section 4.2.1. Ainsi, si $e_{t,abs-expi} > 0$ et que $e_{t-expi} < 0$, plus l'intensité du critère augmente, plus l'utilisateur a tendance à s'écarter à l'intérieur de la trajectoire de référence.

La majorité des coefficients de corrélation est de niveau *très faible* ou *faible* (défini en table 5.1).

Il y a donc peu de dépendance entre une caractéristique d'un profil et l'erreur de trajectoire lors d'un virage à 180°.

Les coefficients de corrélation faibles du type de permis avec les erreurs sont difficilement interprétables. Seul 1 titulaire du permis camion a participé à la procédure d'expériences.

Neuf coefficients de corrélation *modérés* sont également présents.

Les coefficients de corrélation modérés du sexe avec l'erreur se considère avec précaution car la population étudiée contient beaucoup plus d'hommes que de femmes. Nous pouvons simplement affirmer qu'en moyenne, les 5 utilisatrices ont tendance à s'écarter davantage de la trajectoire de référence.

Le tableau reprend aussi huit coefficients de niveau *fort*.

Comme mentionné en section 5.2, l'âge et le nombre d'années de permis sont *très fortement* corrélés entre eux. En conséquence, leur coefficient ρ avec les erreurs de trajectoire sont proches : plus l'utilisateur est âgé plus son erreur de trajectoire en valeur absolue a *tendance* à croître (corrélation forte, voir figure 5.18).

Enfin, le tableau montre que, pour des critères à corrélation forte ou modérée, l'erreur de la deuxième expérience est plus corrélée que la moyenne des erreurs sur les deux expériences, elle-même étant plus corrélée que l'erreur de la première expérience. Néanmoins les ρ demeurent similaires d'une simulation à l'autre pour la majorité des critères. Les critères de la partie inférieure du tableau restent à faible ou très faible ρ alors que la partie supérieure change de maximum un degré d'intensité de corrélation.

La variabilité des erreurs entre les expériences est analysée plus loin dans ce chapitre.

| Coefficient de corrélation ρ | e_{t-exp1} | $e_{t,abs-exp2}$ | e_{t-exp2} | $e_{t,abs-exp2}$ | $e_{t,moy}$ | $e_{t,abs,moy}$ |
|-----------------------------------|--------------|------------------|--------------|------------------|-------------|-----------------|
| Age | -0.34 | 0.51 | -0.55 | 0.65 | -0.49 | 0.62 |
| Sexe | -0.31 | 0.24 | -0.21 | 0.15 | -0.30 | 0.22 |
| Type de permis | -0.24 | 0.22 | -0.24 | 0.25 | -0.27 | 0.25 |
| Nb Années de Permis | -0.32 | 0.48 | -0.60 | 0.63 | -0.51 | 0.59 |
| Km/an | -0.17 | 0.22 | -0.31 | 0.34 | -0.27 | 0.30 |
| Type véhicule | -0.19 | 0.09 | 0.01 | -0.01 | -0.10 | 0.05 |
| Type de trans. | 0.20 | 0 | 0.11 | 0.0 | 0.18 | 0 |
| Style de cond. | 0.06 | 0.12 | -0.03 | 0.04 | 0.02 | 0.09 |
| Intérêt SPM | 0.12 | -0.09 | 0.04 | 0.01 | 0.09 | -0.05 |
| Pratique SPM | 0 | 0 | -0.13 | 0.08 | -0.07 | 0.04 |

TABLE 5.4 – Coefficients de corrélation du virage à 180°

Virage à 90°

Le tableau de corrélation pour le virage à 90° est similaire à celui pour le virage à 180°. Globalement, peu de corrélations fortes existent.

Certains ρ changent d'un niveau de corrélation mais les caractéristiques du profil de l'utilisateur les plus corrélées restent sensiblement les mêmes. Ceci peut se vérifier en analysant la corrélation d'une même erreur entre deux virages (par exemple $e_{t,abs,moy}$ en table 5.5). Pour $e_{t,abs,moy}$, l'utilisateur a une tendance forte à adopter le même comportement sur les 3 types de virages. L'intensité de cette corrélation dépend évidemment du type d'erreur analysée.

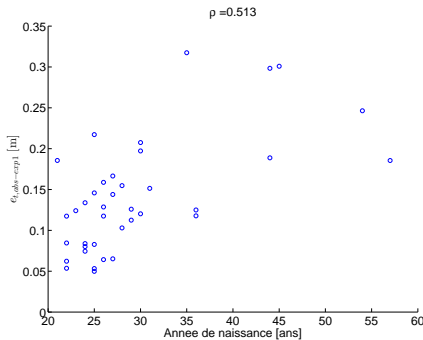


FIGURE 5.18 – Erreur de chaque utilisateur lors du premier virage à 180°

| $e_{t,abs,moy}$ | V 180 | V 90 | V Chicane |
|-----------------|-------|------|-----------|
| V 180 | 1 | 0.75 | 0.59 |
| V 90 | 0.75 | 1 | 0.63 |
| V Chicane | 0.63 | 0.59 | 1 |

TABLE 5.5 – Corrélation entre virages d'une erreur

| Coefficient de corrélation ρ | e_{t-exp1} | $e_{t,abs-exp1}$ | e_{t-exp2} | $e_{t,abs-exp2}$ | $e_{t,moy}$ | $e_{t,abs,moy}$ |
|-----------------------------------|--------------|------------------|--------------|------------------|-------------|-----------------|
| Age | -0.47 | 0.51 | -0.52 | 0.57 | -0.53 | 0.57 |
| Sexe | -0.19 | 0.18 | -0.03 | -0.03 | -0.14 | 0.12 |
| Type de permis | -0.16 | 0.14 | -0.31 | 0.30 | -0.24 | 0.21 |
| Nb Années de Permis | -0.44 | 0.49 | -0.49 | 0.55 | -0.49 | 0.55 |
| Km/an | -0.16 | 0.14 | -0.39 | 0.38 | -0.27 | 0.24 |
| Type véhicule | -0.17 | 0.12 | -0.18 | 0.12 | -0.18 | 0.13 |
| Type de trans. | 0.24 | -0.20 | 0.15 | -0.03 | 0.22 | -0.15 |
| Style de cond. | -0.07 | 0.15 | 0.04 | 0.11 | -0.03 | 0.15 |
| Intérêt SPM | 0.25 | -0.27 | -0.13 | 0.14 | 0.11 | -0.14 |
| Pratique SPM | 0.06 | -0.07 | -0.23 | 0.25 | -0.06 | 0.04 |

TABLE 5.6 – Coefficients de corrélation du virage à 90°

Virage de type chicane

La table 5.7 reprend les ρ pour le virage de type chicane. Le niveau modéré de corrélation est le niveau maximum. Les facteurs de corrélation avec l'âge sont donc moindres.

Si nous nous intéressons à la moyenne des erreurs des deux expériences (avec retour haptique), une corrélation modérée existe pour le style de conduite : plus l'utilisateur juge sa conduite sportive, plus l'erreur non-signée $e_{t,abs_{moy}}$ a tendance à croître et plus l'erreur signée $e_{t_{moy}}$ a tendance à être négative.

Le nombre de km/an est aussi modérément corrélé : plus l'utilisateur conduit par année, plus il a tendance à moins bien suivre la trajectoire de référence.

| Coefficient de corrélation ρ | e_{t-exp1} | $e_{t,abs-exp1}$ | e_{t-exp2} | $e_{t,abs-exp2}$ | $e_{t_{moy}}$ | $e_{t,abs_{moy}}$ |
|-----------------------------------|--------------|------------------|--------------|------------------|---------------|-------------------|
| Age | -0.16 | 0.28 | -0.30 | 0.44 | -0.24 | 0.38 |
| Sexe | 0.03 | -0.06 | 0.04 | 0 | 0.04 | -0.04 |
| Type de permis | -0.08 | 0.10 | -0.21 | 0.21 | -0.15 | 0.16 |
| Nb Année de Permis | -0.21 | 0.22 | -0.35 | 0.46 | -0.29 | 0.35 |
| km/an | -0.31 | 0.34 | -0.34 | 0.28 | -0.34 | 0.34 |
| Type vehicule | 0 | 0.13 | -0.08 | 0.03 | -0.04 | 0.09 |
| Type de trans. | -0.28 | 0.18 | -0.29 | 0.28 | -0.30 | 0.24 |
| Style de cond. | -0.29 | 0.37 | -0.28 | 0.38 | -0.30 | 0.40 |
| Intérêt SPM | -0.07 | 0.02 | -0.17 | 0.10 | -0.12 | 0.06 |
| Pratique SPM | 0.02 | -0.13 | -0.05 | 0.10 | -0.01 | -0.03 |

TABLE 5.7 – Coefficients de corrélation du virage de type chicane

Rond-point

Pour le rond-point, les erreurs de trajectoire peuvent aussi être utilisées. Cependant, lorsque les forces latérales sont soudainement réduites, certains utilisateurs débraquent trop brusquement et perdent le contrôle du kart.

En effet, à partir d'un certain niveau de saturation des forces latérales, le modèle de Bakker utilisé ne modélise plus correctement la physique du pneu et le véhicule devient incontrôlable jusqu'à la fin de la simulation.

Dès lors, nous utilisons un indicateur $fail = 0$ ou $pass = 1$ pour chacune des trois prises de rond-point. La moyenne des trois est calculée pour obtenir un indicateur croissant de réussite $FoP = 0$ ou $\frac{1}{3}$ ou $\frac{2}{3}$ ou 1.

La table 5.8 reprend les facteurs de corrélation entre cet indicateur et les critères du profil des utilisateurs.

Globalement, peu de corrélations de haute intensité sont présentes. L'intensité maximale est celle liée au sexe, dont la validité a déjà été discutée. Les personnes plus âgées ont aussi davantage tendance à perdre le contrôle du kart plus souvent que la population plus jeune.

| ρ | FoP | | FoP |
|----------------|-------|------------------|-------|
| Age | -0.29 | Type de vehicule | -0.13 |
| Sexe | -0.32 | Type de trans. | 0.09 |
| Type de permis | 0 | Style de cond | 0.22 |
| Nb Anne Permis | -0.29 | Intérêt SPM | 0.23 |
| km/an | 0.28 | Pratique SPM | 0.13 |

TABLE 5.8 – Coefficients de corrélation du rond-point

Deux autres caractéristiques du suivi de trajectoire sont analysées dans la suite de cette section.

Amélioration du suivi de trajectoire

Entre la première expérience et la deuxième expérience d'un même virage, la majorité des utilisateurs diminue les erreurs définies en table 5.3 (en valeur absolue). Par exemple, la table 5.9 reprend les pourcentages d'amélioration de l'"erreur absolue", en définissant ce pourcentage comme $P_{amel} = \frac{\#(e_{(t,abs)-vir-1} - e_{(t,abs)-vir-2}) > 0}{N_{id}}$,

| | Vir. 180° | Vir. 90° | Vir. Chicane |
|------------|-----------|----------|--------------|
| P_{amel} | 73.68% | 76.32% | 84.21% |

TABLE 5.9 – Pourcentage d'utilisateurs diminuant leur "erreur absolue" de trajectoire

Une interprétation possible du résultat est l'effet de surprise que génère la première simulation. Lors de la deuxième simulation, l'utilisateur connaît le déroulement de la simulation, est déjà plus habitué à la sensibilité du volant, la position adéquate à lui donner,...

Intérêt du retour haptique

Pour l'analyse suivante, un id est assigné à chacun des $N_{id} = 38$ utilisateurs.

Les expériences "1" et "2" du virage sont effectuées avec le retour en couple et les expériences "3" et "4" sans. Les définitions de l'erreur totale absolue pour les virages "1" et "3" sont alors :

$$etot_{(t,abs)-ch-1} = \frac{\sum_{id=1}^{N_{id}} e_{t,abs-ch-1}(id)}{N_{id}} \quad etot_{t,abs-ch-3} = \frac{\sum_{id=1}^{N_{id}} e_{(t,abs)-ch-3}(id)}{N_{id}} \quad (5.6)$$

Entre la troisième et première expérience, le facteur d'amélioration du suivi de l'erreur trajectoire grâce au retour haptique est défini par $\frac{etot_{t,abs-ch-exp3}}{etot_{(t,abs)-ch-exp1}} \approx 1.85$.

En analysant chaque utilisateur individuellement, 95% des utilisateurs suivent mieux la trajectoire idéale avec retour haptique plutôt que sans.

Le retour haptique diminue donc l'erreur de trajectoire de façon significative comme le montre aussi les figure 5.19 et 5.20.

La figure 5.19 montre la trajectoire moyenne des 38 utilisateurs lors d'une courbe de la chicane. La figure 5.20 montre la moyenne des erreurs instantanées sur l'ensemble de la chicane (faite aussi sur les 38 utilisateurs).

L'amélioration est d'autant plus importante lors des braquages/débraquages brusques des deux premières courbes de la chicane. En effet, pour le braquage/débraquage brusque, l'utilisateur ne subit aucune résistance dans le volant et dépasse l'angle de braquage/débraquage adapté au virage. L'utilisateur est coupé de la dynamique du véhicule et du pneu, de ce qu'il est habitué à ressentir.

Lors du long virage à angle de braquage constant, le retour haptique est légèrement moins bénéfique. L'utilisateur essaie continuellement d'ajuster la position qu'il donne au volant, ce qui lui fournit un couple différent et ainsi de suite. Sans le retour en couple, l'utilisateur pourrait placer progressivement le volant à la position idéale et le relâcher pour le restant de la courbe. Toutefois, trouver la position idéale du volant sans retour haptique est problématique au vu de la figure 5.20.

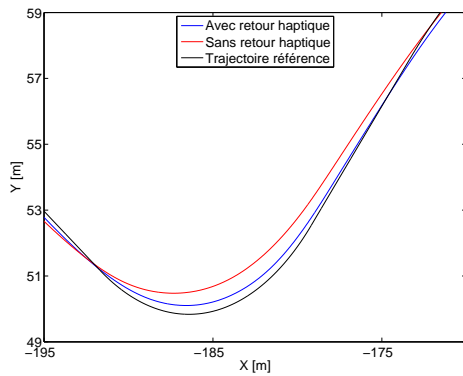


FIGURE 5.19 – Effet du retour haptique sur le suivi de trajectoire

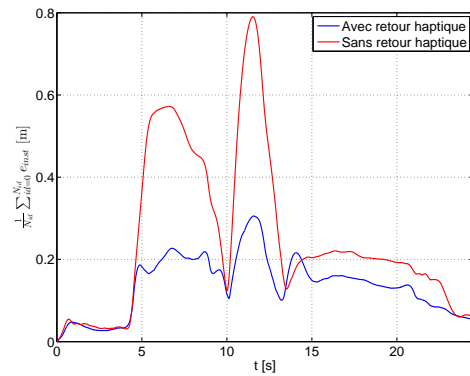


FIGURE 5.20 – Moyenne des erreurs instantanées des 38 utilisateurs

5.3.3 Conclusion

Cette section a présenté différentes approches d'analyse des données enregistrées lors du déroulement de la procédure d'expérience. Elle a validé le retour haptique comme élément facilitateur du suivi de trajectoire et exploré les corrélations qui existent entre le profil des conducteurs et le suivi de trajectoire pour une population donnée, celles avec l'âge étant les plus importantes. Elle a analysé les corrélations qui existent entre le profil et les réponses de l'utilisateur et mis en évidence l'importance de la familiarisation à une expérience sur le suivi de trajectoire.

D'autres analyses pourraient être effectuées ultérieurement grâce aux nombreuses données récoltées lors de ces expériences. Citons par exemple la vitesse de l'erreur de trajectoire, l'erreur de lacet, le degré d'amélioration des conducteurs, la position des mains sur le volant, ... tous pouvant être liés au profil des utilisateurs.

Chapitre 6

Perspectives et améliorations

A l'issue de ce travail, le simulateur de véhicule est parfaitement fonctionnel et a pu être utilisé par 38 personnes dans le cadre d'une étude statistique.

Néanmoins, certains développements futurs pourraient améliorer l'expérience de l'utilisateur. Ce chapitre détaille d'une part les développements informatiques et d'autre part les extensions matérielles qui pourraient être apportés au simulateur.

6.1 Améliorations du simulateur sans extension de l'hardware

Thread de visualisation

L'architecture avec interruptions comporte trois threads : le premier qui résout les équations du mouvement du véhicule et les deux autres qui communiquent avec la carte de développement. Le premier cité, thread *Robotran*, permet la visualisation en temps réel de l'évolution temporelle du modèle multicorps. Cette visualisation pourrait être isolée sur un quatrième thread en utilisant notamment, une mémoire qui partagerait les q, \dot{q}, \ddot{q} avec le thread *Robotran*. Cela diminuerait la surcharge du processeur et permettrait ainsi de réduire encore le ratio temps réel v_r .

Accélération

Le simulateur ne prend pas en compte l'accélération qu'impose l'utilisateur au volant. Une tentative de prise en compte a été réalisée en approximant puis dérivant deux fois le signal de position selon la même méthode exposée en [9]. Les résultats de celle-ci ne permettaient pas de conclure que les oscillations de couple diminuaient et l'algorithme a alors été abandonné.

Une étude plus approfondie de la prise en compte de l'accélération permettrait d'obtenir un signal de couple plus réaliste dans des situations à haute dynamique.

Changement de protocole de communication

Sur le PC, un driver émule le port série à partir des ports USB.

Ce driver utilise en outre un buffer interne qui sert de zone tampon entre le port USB et l'application. Il garde les données reçues et les transfère à l'application quand il a atteint sa capacité maximale ou qu'un timer interne expire.

La taille minimum du buffer étant de 64 bytes, c'est le timer qui gère la fréquence de réception de l'application. La latence minimum de ce timer est ~ 2 [ms].

La communication série pourrait être améliorée en utilisant par exemple des procédés d'« handshaking » qui suppriment la contrainte du timer. La communication pourrait aussi être remplacée par des communications, plus difficiles à implémenter mais plus rapides comme l'Ethernet avec laquelle la carte Texas est compatible.

6.2 Améliorations du simulateur avec extension de l'hardware

Comparativement aux améliorations précédentes, ces extensions nécessitent d'apporter des modifications physiques au simulateur et pourraient faire l'objet d'un ou plusieurs travaux à part entière.

Système d'exploitation temps réel

Windows est un GPOS ("General Purpose Operating System"). Il assure toutes les fonctions basiques d'un ordinateur comme par exemple la gestion du multithread mais ne garantit pas des performances en temps réel.

Prenons l'exemple du thread *Robotran* qui serait défini non-prioritaire par l'OS. Dans le cas d'un OS idéal, les q, \dot{q} sont pris dans le buffer "thread PC \longleftrightarrow threads communications" toutes les Δt_{sim} secondes. Cependant si le thread est en attente, un délai est ajouté. Parallèlement, si le thread de communication récupérant les q, \dot{q} du volant s'exécute plus rapidement, il va déposer et écraser successivement plusieurs données q, \dot{q} dans le buffer "thread PC \longleftrightarrow threads communications" sans que le thread *Robotran* puisse en faire usage. Les délais de "gel" du thread *Robotran* confèrent une réponse saccadée ("jerky") à la position/vitesse. Cela influence le couple transmis au volant.

Dans le cadre d'un outil pédagogique, ces délais sont peu problématiques : ils sont généralement courts de telle sorte qu'ils aient peu d'impact sur la boucle de contrôle du simulateur et le ressenti au volant.

Néanmoins, l'utilisation d'un *RTOS* ("Real Time Operating System") garantirait des performances déterministes et améliorerait sensiblement les performances du simulateur.

Vue périphérique

Le simulateur actuel ne possède qu'un écran placé en face du siège. Ajouter deux écrans de part et d'autre de l'écran central donnerait une meilleure vision périphérique et une expérience de conduite plus réaliste.

Pédalier

Sur le cahier des charges de conception électromécanique, le simulateur devait initialement être doté de trois pédales. Cela n'a pas été implémenté par les étudiants du travail précédent ni pour celui-ci.

Cependant, l'ajout des pédales donnerait une simulation de conduite plus réaliste : l'actionnement des pédales est couplé à celui du volant.

Il ouvrirait aussi le champ à des études statistiques de l'interaction homme-véhicule plus variées. Par exemple, des expériences de freinage et d'évitement d'urgence pourraient être analysées.

Système inertiel

Excepté le volant, tous les éléments mécaniques du simulateur sont statiques. La conception d'un mécanisme permettant à l'utilisateur de ressentir le roulis et le tangage du véhicule améliorerait son expérience sur simulateur.

Le simulateur pourrait par exemple être placé sur une plateforme mobile reliée à une base fixe au moyen de pistons.

Conclusion

Ce travail se place judicieusement dans le cadre d'une formation d'ingénieur civil mécatro-cien : les trois domaines principaux qui forment la discipline y ont trouvé application.

La mécanique d'une part pour comprendre chaque élément de la chaîne liant le pneu au volant d'un véhicule. L'informatique d'autre part, omniprésente dans le cadre du développement d'un simulateur. Et enfin l'électronique, support de toute l'informatique.

Par l'application quotidienne de ces trois branches, le simulateur est parfaitement fonction-nel.

L'utilisateur expérimente, en temps réel, des sensations de conduite réalistes grâce à un retour haptique issu d'un modèle multicorps *Robotran*.

Une exploitation du dispositif a pu être réalisée pour étudier l'interaction entre l'humain et le volant.

Le simulateur pourrait encore être amélioré pour modéliser le mouvement complet d'un véhi-cule. Cette première étude ouvre des possibilités à bien d'autres travaux sur le simulateur actuel tant le sujet de l'interaction homme-volant est complexe et inexploré.

Bibliographie

- [1] Samin Jean-Claude and Fisette Paul. *Symbolic Modeling of Multibody Systems*. Kluwer Academic Publishers, 2003.
- [2] Labrosse Jean J. *MicroC/OS-II : The Real Time Kernel*. CMP Books, 2002.
- [3] Harris Davis M. and Harris Sarah L. *Digital Design and Computer Architecture*. Morgan Kaufmann, 2013.
- [4] Wackerly Dennis D., Mendenhall III William, and Scheaffer Richard L. *Mathematical Statistics with Applications*. Thomson Brooks/Cole, 2008.
- [5] Cohen Jacob. *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates, 1988.
- [6] Mechin P and Mechin J.P. *Le guide technique du karting*. J.P.M, Juin 1993.
- [7] Pr. Paul Fisette. *LMECA2215 : Dynamique des véhicules*.
- [8] Pr. Paul Fisette. *LMECA2802 : Mécanique des systèmes articulés*.
- [9] Pr. Vincent Legat. *FSAB1104 : Méthodes numériques*.
- [10] Romain Poletti. *Projet de dynamique des véhicules LMECA2215 : Rapport de projet du cours de dynamique des véhicules : Etude de l'influence des paramètres géométriques du train avant d'un kart*.
- [11] De Lellis Nicola and Materne Quentin. *TFE : Conception et réalisation d'un simulateur à retour haptique pour l'étude de l'interaction homme/véhicule*.
- [12] De Lellis Nicola and Materne Quentin. *Rapport de projet du cours de dynamique des véhicules : Etat de l'art sur les simulateurs de conduite*.
- [13] Gier Martin and Rox Julian. *TFE : Etude dynamique d'un véhicule automobile à échelle via la simulation temps réel et l'animation virtuelle*.
- [14] http://www.robotran.be/images/files/Robotran_basics.pdf.
- [15] <https://www.pinterest.com/pin/202591683212447498/>.
- [16] www.birelart.com/sites/default/files/page/documents/BIREL%20ART-BIREL%20ART%2032-CH-20_0.pdf.
- [17] <http://www.superkart.it/fiches-techniques/mecanique/fiche-technique-fusee-referance-sk1075>.
- [18] <https://fr.wikipedia.org/wiki/Haptique>.
- [19] <http://www.tornil.me/these/These%20Bertrand%20Tornil.pdf>.
- [20] [https://fr.wikipedia.org/wiki/Thread_\(informatique\)](https://fr.wikipedia.org/wiki/Thread_(informatique)).
- [21] <http://www.ti.com/tool/tmdsice3359#1>.
- [22] http://www.ftdichip.com/Support/Documents/TechnicalNotes/TN_111%20What%20is%20UART.pdf.
- [23] <http://www.ftdichip.com/Drivers/VCP.htm>.
- [24] <http://www.ni.com/white-paper/6424/fr>.

- [25] <http://www.cikfia.com/competitions/cik-calendar/2016/genk/competition-guide.html>.
- [26] https://fr.wikipedia.org/wiki/Segment_automobile.
- [27] http://www.cikfia.com/fileadmin/content/REGULATIONS/Technical/Updates/2016/01_Bull_41_RT_Article_18-19.pdf.
- [28] http://download.schneider-electric.com/files?p_Reference=0198441113749-EN&p_EnDocType=User%20guide&p_File_Id=7350052010&p_File_Name=BMH_Manual_V2.1_EN.pdf.
- [29] <http://www.ti.com/general/docs/datasheetdiagram.tsp?genericPartNumber=AM3359&diagramId=46861>.
- [30] Schneider Electrics. *LXM32A CANopen, Fieldbus Manual*, October 2012.

