



**Projet S7**  
**Conception et commande d'un mini Segway**

**Etudiants**  
**Ayman MOUMMADI, Logan PAQUIN**  
**Maxence NEUS, Omar SIFA**

**Tuteurs**  
**Mr Midzodzi Pekpe, Mr Othman Lakhal**

**Janvier 2022**

## Résumé

Notre sujet se nomme “Conception et commande d’un segway”. Nous devons réaliser un segway miniature et le contrôler à distance. Autrement dit, identifier les composants nécessaires pour la fabrication du segway et le concevoir au Fabricarium de Polytech Lille.

Ce rapport présente l’avancement final de notre projet. Il reprend les parties abordées au semestre 6, en apportant ce qui à été ajouté lors du semestre 7. Il présente également ce que nous n’avons pas pu réaliser, ainsi que comment nous aurions pu effectuer ces tâches.

## Abstract

Our subject is called ”Design and control of a segway”. We have to make a miniature segway and remote it. In other words, identify the necessary components for the manufacture of the segway and conceive it at the Fabricarium of Polytech Lille.

This wiki report presents the final progress of our project. It takes the parts tackled in the past semester (S6), and adds all that have been done since. Then it describes what we could not do, how we would have done those tasks and why we didn’t.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Cahier des charges</b>	<b>4</b>
<b>3</b>	<b>Etat de l'art</b>	<b>7</b>
<b>4</b>	<b>Organisation</b>	<b>8</b>
4.1	Planning . . . . .	8
4.2	Répartition des tâches . . . . .	9
<b>5</b>	<b>Conception de la commande</b>	<b>10</b>
5.1	Modélisation du Segway . . . . .	10
5.2	Conversion Numérique . . . . .	13
<b>6</b>	<b>Conception de la maquette</b>	<b>16</b>
6.1	Concept de base . . . . .	16
6.2	Architecture matérielle . . . . .	16
6.3	Architecture logicielle . . . . .	17
6.4	Changements à apporter . . . . .	18
<b>7</b>	<b>Conclusion</b>	<b>19</b>

# 1 Introduction

## Présentation du projet

Notre projet consiste à réaliser un Segway miniature et à le contrôler à distance. Un Segway est un gyropode, soit un véhicule électrique monoplace, constitué d'une plateforme munie de deux roues parallèles sur laquelle l'utilisateur se tient debout, d'un système de stabilisation gyroscopique.

Nous allons créer les schémas électriques et identifier les composants nécessaires à la fabrication du Segway puis le concevoir au Fabricarium de Polytech. Ensuite nous allons programmer différentes commandes pour gérer le Segway. Cela inclut un contrôle à distance ,puis l'implémentation d'un programme de pilotage autonome à l'aide d'une caméra incluse sur le système.

## 2 Cahier des charges

Le système que nous devons réaliser étant une version miniature d'un Segway, les attentes sont différentes. Alors que le Segway a pour vocation d'offrir un moyen de transport personnel et agile en milieu urbain, notre système doit pouvoir être piloté par l'utilisateur à distance.

De plus, notre système est prévu pour une utilisation en intérieur rendue possible grâce à sa taille modeste, mais peut-être amené à évoluer pour une utilisation en extérieur moyennant notamment l'ajout d'une carcasse permettant une isolation contre les liquides et la poussière suffisante.

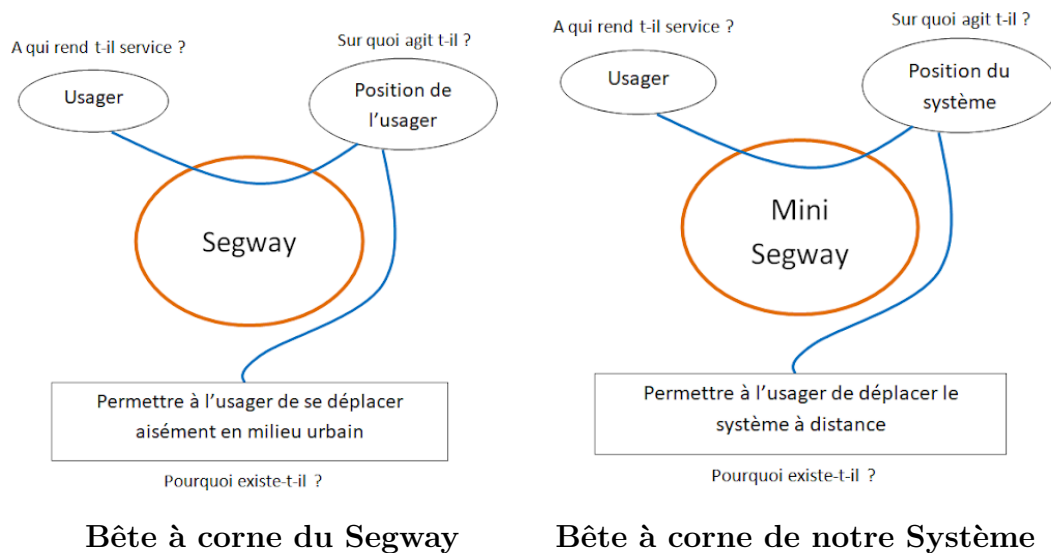
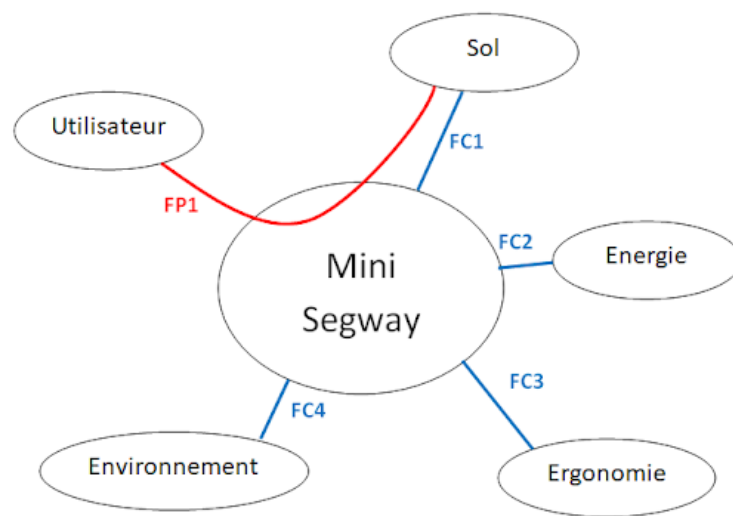


Figure 1: Bêtes à corne

Notre système évolue donc dans un environnement simple, ce qui permet de limiter la complexité des contraintes. Voici le diagramme pieuvre décrivant les contraintes auxquelles devra répondre notre système.



FP1	Se déplacer par rapport au sol conformément aux commandes envoyées par l'utilisateur à distance.
FC1	Se tenir en équilibre vertical par rapport au sol en toute circonstance.
FC2	Avoir une autonomie en énergie suffisante.
FC3	Être de taille modeste et facilement transportable.
FC4	Se déplacer en intérieur

Figure 2: Description des besoins

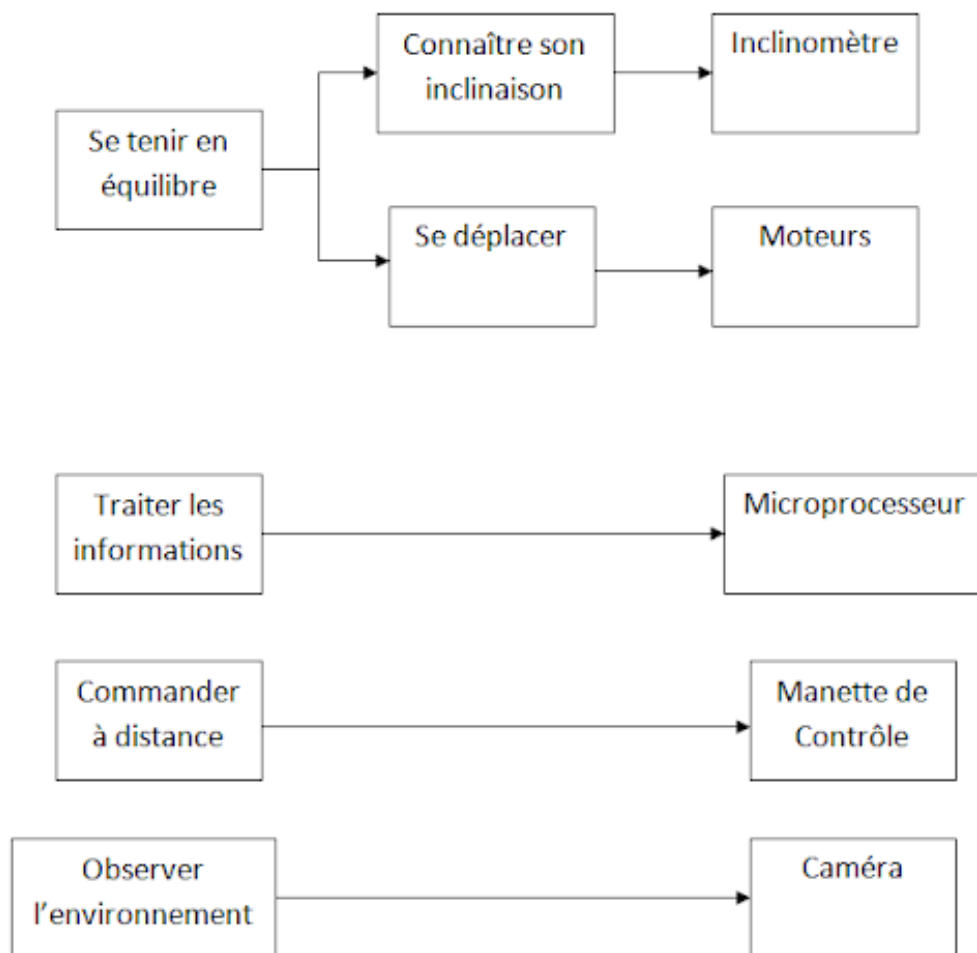


Figure 3: Schéma fonctionnel de notre système

### 3 Etat de l'art

Le début du 21ème siècle aura été chargé en innovations technologiques en tout genre. Parmi elles, le Segway semble clairement avoir fait l'une des entrées les plus fracassantes sur le marché. En 2001, le premier Segway de Dean Kamen promettait alors de révolutionner la mobilité urbaine avec une sortie de scooter à deux roues auto équilibré, capable d'être piloté par la seule force de sa pensée, ou presque. Vous connaissez le principe : le Segway peut être piloté via une simple inclinaison du corps dans un sens comme dans l'autre.

Hélas, malgré la révolution annoncée – jusqu'à s'attirer les éloges d'un certain Steve Jobs à sa présentation – le Segway n'a pas réussi à trouver sa place parmi le trafic urbain, alors que les villes l'ont peu à peu banni de leurs routes. La Segway original a alors été relégué au rang de gadget onéreux, mais n'a pas trouvé sa place chez le grand public. Malgré l'échec du premier modèle, la technologie d'auto-équilibrage apportée par le Segway original a subsisté dans bien d'autres produits : Dean Kamen, créateur de l'engin, a finalement vendu son Segway à l'entreprise chinoise Ninebot en 2015. Celle-ci n'a pas manqué de décliner le concept en une multitude de produits, et notamment, des engins moins onéreux et plutôt destinés aux jeunes, tels que ceux qu'on appelle aujourd'hui hoverboard.

<https://www.journaldugeek.com/2020/06/24/premier-segway-tire-reverence-pourquoi-manquer/>



## 4 Organisation

### 4.1 Planning

Pour pouvoir nous organiser au mieux, nous avons fait un GANTT (comme au S6) qui met des “deadlines” pour chaque partie du projet. Ci-dessous le GANTT prévisionnel qu’on a réalisé au début de semestre.

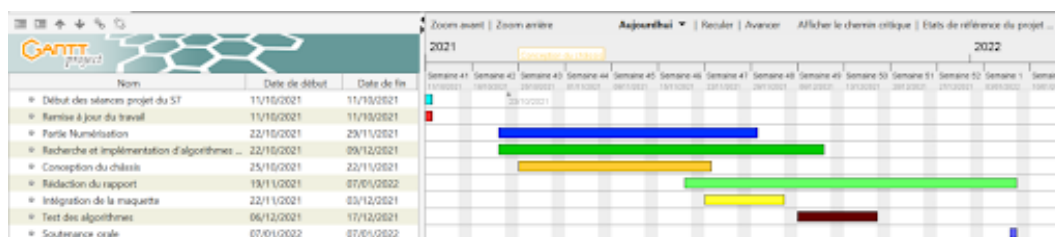


Figure 4: GANTT prévisionnel

Afin d’être critique et pouvoir expliquer les tâches faites à temps et celles où ce n’est pas le cas, nous avons fait un GANTT réel (voir ci-dessous).

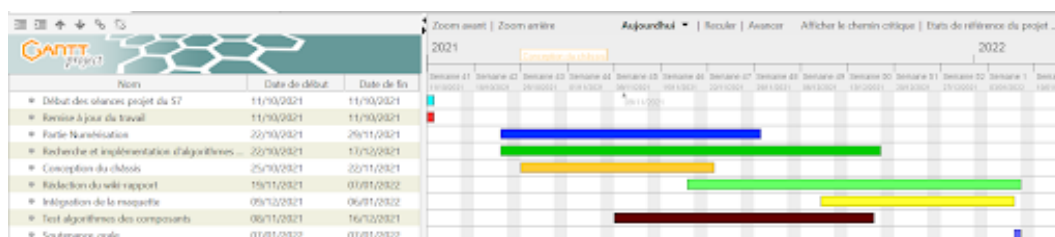


Figure 5: GANTT réel

On peut remarquer qu’on a plus ou moins respecté le GANTT fixé au début avec quelques retards dûs principalement au non fonctionnement de notre accéléromètre qu’on a dû changer et aussi les moteurs pas à pas qui ne fonctionnent pas bien.

## 4.2 Répartition des tâches

On s'est réparti les tâches au début de semestre comme suivant:

Maxence Programmation

Omar et Ayman Recherche bibliographique

Logan et Omar Modélisation du segway

Ayman Rédaction des suivis sur le WIKI

Néanmoins, à la fin de chaque séance du projet, chacun montrait son avancée et ce qu'il a pu faire durant la séance.

## 5 Conception de la commande

### 5.1 Modélisation du Segway

La régulation d'inclinaison du Segway consiste à maintenir la consigne  $\psi_c(t)$  nulle. Cette régulation est réalisée si, quelle que soit l'inclinaison  $\alpha(t)$  du conducteur, la sortie  $\psi(t)$  converge vers  $\psi_c(t)$ , valeur nulle ici. Le conducteur agit directement sur la valeur  $\alpha(t)$  de pour accélérer ou décélérer. Pour le système Segway, conducteur exclu, le paramètre peut être considéré comme une perturbation. Le schéma-bloc fonctionnel du Segway est donc le suivant:

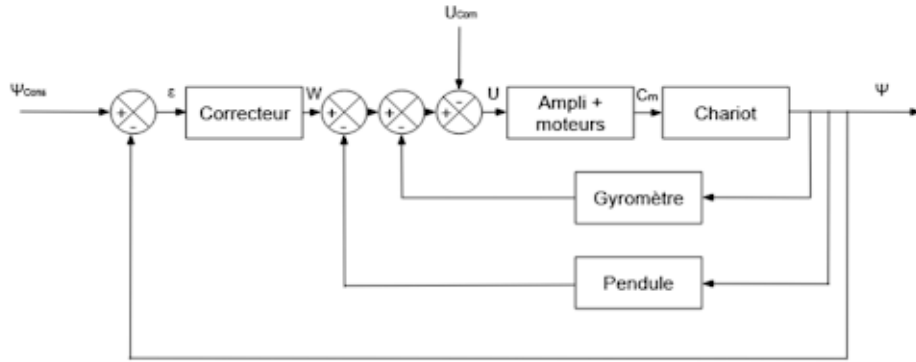


Figure 6: Schéma bloc du Segway

Il est composé de différents éléments :

Un correcteur, pour réguler la commande

L'ensemble ampli + moteurs qui délivre un couple  $C_m(p) = K.U(p)$

L'ensemble chariot + conducteur (le système mécanique) modélisé par l'équation  $\chi(p) = \psi(p) - \alpha(p) = F(p)U(p) - \alpha(p)$

avec :

$$F(p) = \frac{\frac{2(B+DR)}{RDC}}{\frac{(DA+B^2)}{DC}p^2 - 1}$$

$$A=90\text{kg.m}^2 \quad B=75\text{kg.m} \quad C=750\text{kg.m}^2/\text{s}^2 \quad D=125\text{kg} \quad R=240\text{mm}$$

(ces valeurs seront expliquées dans la partie juste après)

Le gyromètre :  $u_g(t) = k_g \frac{d\psi(t)}{dt}$  soit  $U_g(p) = pk_g\psi(p)$

Le pendule :  $u_p(t) = k_p\psi(t)$  soit  $U_p(p) = k_p\psi(p)$

Nous avons choisi de prendre comme valeurs initiales pour le segway (l'ensemble conducteur + chariot) les valeurs fournies dans ces tableaux, accompagnées des matrices d'inerties correspondantes au système étudié:

<b>Conducteur : <math>H</math></b>	
Centre de gravité $G$ avec $\overrightarrow{AG} = h\vec{z}_4$  $h = 0,95 \text{ m}$  $m_H = 80 \text{ kg}$	$I_G(H) = \begin{pmatrix} A_H & 0 & 0 \\ 0 & B_H & 0 \\ 0 & 0 & C_H \end{pmatrix}_{\vec{x}_4, \vec{y}_4, \vec{z}_4}$ $A_H = 18 \text{ kg} \cdot \text{m}^2$ $B_H = 20 \text{ kg} \cdot \text{m}^2$ $C_H = 1,2 \text{ kg} \cdot \text{m}^2$
<b>Châssis : <math>S</math></b>	
Centre de gravité $A$ $m_s = 25 \text{ kg}$	$I_A(S) = \begin{pmatrix} A_S & 0 & 0 \\ 0 & B_S & 0 \\ 0 & 0 & C_S \end{pmatrix}_{\vec{x}_1, \vec{y}_1, \vec{z}_1}$ $A_S = 0,8 \text{ kg} \cdot \text{m}^2$ $B_S = 1,2 \text{ kg} \cdot \text{m}^2$ $C_S = 1,3 \text{ kg} \cdot \text{m}^2$
<b>Roue gauche : <math>R_G</math></b>	
Centre de gravité $O_G$ , avec $\overrightarrow{AO_G} = \frac{L}{2}\vec{x}_1$ Rayon $R = 240 \text{ mm}$ $L = 650 \text{ mm}$ $m_R = 5 \text{ kg}$	$I_{O_G}(R_G) = \begin{pmatrix} A_R & 0 & 0 \\ 0 & B_R & 0 \\ 0 & 0 & C_R \end{pmatrix}_{\vec{x}_1, \vec{y}_{O_G}, \vec{z}_{O_G}}$ $A_R = 0,28 \text{ kg} \cdot \text{m}^2$ $B_R = 0,05 \text{ kg} \cdot \text{m}^2$ $C_R = 0,05 \text{ kg} \cdot \text{m}^2$
<b>Roue droite : <math>R_D</math></b>	
Centre de gravité $O_D$ , avec $\overrightarrow{DO_D} = \frac{L}{2}\vec{x}_1$ Rayon $R = 240 \text{ mm}$ $L = 650 \text{ mm}$ $m_R = 5 \text{ kg}$	$I_{O_D}(R_D) = \begin{pmatrix} A_R & 0 & 0 \\ 0 & B_R & 0 \\ 0 & 0 & C_R \end{pmatrix}_{\vec{x}_1, \vec{y}_{O_D}, \vec{z}_{O_D}}$ $A_R = 0,28 \text{ kg} \cdot \text{m}^2$ $B_R = 0,05 \text{ kg} \cdot \text{m}^2$ $C_R = 0,05 \text{ kg} \cdot \text{m}^2$

La masse et l'inertie de la motorisation et de son réducteur associé sont supposées négligeables. Le moto-réducteur fait partie du châssis.

C'est ainsi que l'on trouve les valeurs de A, B, C, D et R (rayon d'une roue) pour l'ensemble du système mécanique modélisé par l'équation  $\chi(p) = \psi(p) - \alpha(p) = F(p)U(p) - \alpha(p)$

avec :

$$F(p) = \frac{\frac{2(B+DR)}{RDC}}{\frac{(DA+B^2)}{DC}p^2 - 1}$$

On a en effet les équations suivantes:

$$\begin{cases} A\ddot{\Psi}(t) - B\dot{V}(t) = -2C_S(t) + C(\alpha(t) + \Psi(t)) \\ B\ddot{\Psi}(t) - D\dot{V}(t) = \frac{2C_S(t)}{R} \end{cases}$$

En prenant les hypothèses suivantes

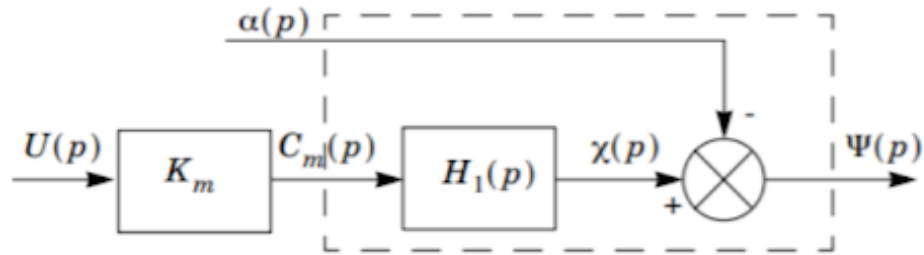
$$\begin{aligned} \sin(\alpha + \psi) &= \alpha + \psi \\ \cos(\alpha + \psi) &= 1 \end{aligned} \quad \text{et les termes en } \dot{\psi}^2 \text{ négligeable devant ceux en } \dot{\psi}$$

..

Puis, on en déduit :

$$\begin{aligned} A &= A_H + A_S + m_H h^2 & B &= m_H h & C &= m_H h g \\ D &= m_H + m_s + 2m_R + 2 \frac{A_R}{R^2} \end{aligned}$$

$$\begin{aligned} \text{A.N. : } A &= 18 + 0,8 + 80.0,95^2 = 91 \text{ kg} \cdot \text{m}^2 & B &= 80.0,95 = 76 \text{ kg} \cdot \text{m} \\ C &= 80.0,95.9,81 = 745,5 \text{ N} \cdot \text{m} & D &= 80 + 25 + 10 + 2.0,28 / 0,24^2 = 124,7 \text{ kg} \end{aligned}$$



Nous étudions la fonction de transfert de :  $F(p) = \frac{\Psi(p)}{U(p)}$

La régulation d'inclinaison du Segway® est réalisée par :

- un moto-réducteur qui permet de délivrer un couple  $C_m(t) = K_m u(t)$  où  $u(t)$  est une grandeur de commande et  $K_m = 24 \text{ N} \cdot \text{m} \cdot \text{V}^{-1}$
- le système mécanique dont les équations ont été déterminées et qui peuvent, dans le cas où l'angle  $\alpha(t)$  n'est pas supposé constant, se mettre sous la forme :

$$\begin{cases} \dot{V}(t) = \frac{1}{D} \left( B\ddot{\chi}(t) + 2\frac{C_m(t)}{R} \right) \\ (DA - B^2)\ddot{\chi}(t) = 2\left(\frac{B}{R} + D\right)C_m(t) + DC\chi(t) \end{cases} \quad \text{ou} \quad \begin{cases} A = 90 \text{ kg} \cdot \text{m}^2 \\ B = 75 \text{ kg} \cdot \text{m} \\ C = 750 \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-2} \\ D = 125 \text{ kg} \\ R = 240 \text{ mm} \\ \chi(t) = \alpha(t) + \Psi(t) \end{cases}$$

Par commodité de signe, la notation  $C_m(t) = -C_S(t)$  est utilisée dans les équations ci-dessus. Les conditions initiales sont toutes nulles.

Les conditions sont toutes nulles, nous obtenons à partir la seconde équation différentielle dans Laplace(ci-dessus), l'équation suivante :

$$((DA - B^2)p^2 - DC)\chi(p) = 2\left(\frac{B}{R} + D\right)C_m(p)$$

Nous avons  $H_1(p) = \frac{\chi(p)}{C_m(p)}$  d'après le schéma bloc ci-dessous.  
Nous en déduisons :

$$H_1(p) = \frac{2\left(\frac{B}{R} + D\right)C}{(DA - B^2)p^2 - DC} = \frac{\frac{2\left(\frac{B}{R} + D\right)C}{DC}}{\frac{DA - B^2}{DC}p^2 - 1} = \frac{2\left(\frac{B + DR}{RDC}\right)}{\frac{DA - B^2}{DC}p^2 - 1}$$

Le schéma bloc se justifie alors en écrivant :

$$\psi(p) = \chi(p) = K_m H_1(p) U(p)$$

Le système d'entrée  $u(t)$  et de sortie  $\psi(t)$  est instable car le dénominateur de la fonction de transfert possède un pôle réel positif.

$$p = \pm \sqrt{\frac{DC}{DA - B^2}}$$

avec

$$\frac{DA - B^2}{DC} = 0.06 > 0$$

Nous déterminons les conditions sur  $K_v$  et  $K_p$  pour que le système soit stable cette fois-ci avec  $\alpha = 0$  (Voir schéma ci-dessous)

$$\frac{\psi(p)}{W(p)} = \frac{\frac{K_m H_1(p)}{1 + p K_v K_m H_1(p)}}{1 + K_p \frac{K_m H_1(p)}{1 + p K_v K_m H_1(p)}} = \frac{K_m H_1(p)}{1 + p K_v K_m H_1(p) + K_p K_m H_1(p)} = \frac{K_s}{\frac{p^2}{\omega^2} + p K_v K_s + K_p K_s - 1}$$

C'est un système de degré 2 qui est stable si et seulement si les coefficients des puissances de  $p$  sont différents de 0 et sont de même signe.

Ceci implique :  $K_p > \frac{1}{K_s}$  et  $K_v > 0$  pour que le système soit stable.

## 5.2 Conversion Numérique

Dans cette partie nous avons ajouté un CAN dont le but de modéliser notre segway miniature

Nous allons maintenant vérifier le comportement de notre Segway miniature afin de nous assurer de son bon fonctionnement. Nous allons pour cela analyser sa réponse pour une consigne d'angle de 1.57 rad (soit un angle de 90°), puis essayer de la corriger avec un régulateur

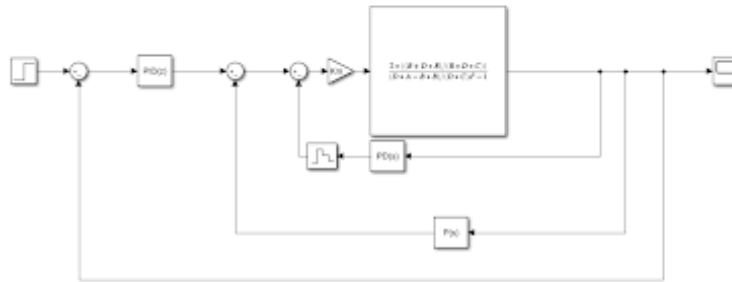


Figure 7: Schéma bloc du mini Segway



Figure 8: Sans correcteur

Nous pouvons voir que l'erreur est très grande, en effet le système ne démarre pas.

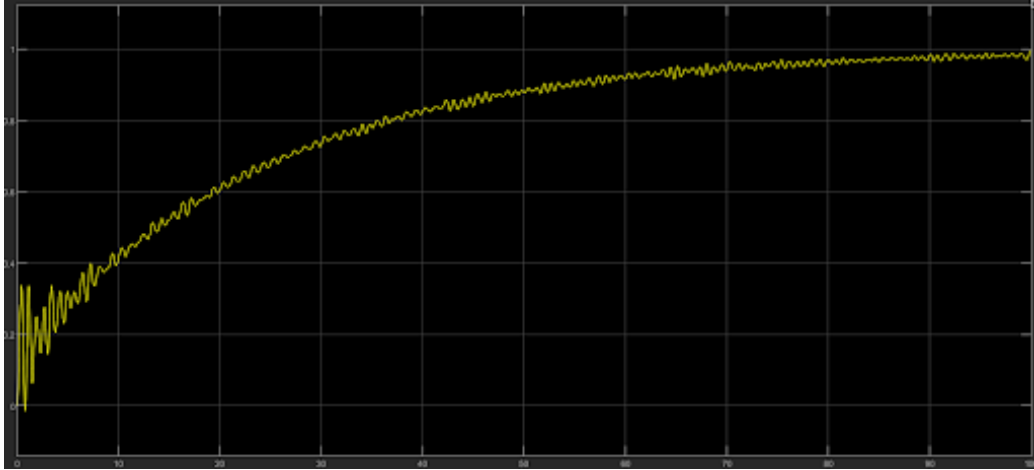


Figure 9:  $P=1.9$  et  $I=0.9$

Nous remarquons que le système a une erreur plus petite mais le temps de réponse est grand, des oscillations dans les 5 premières secondes.

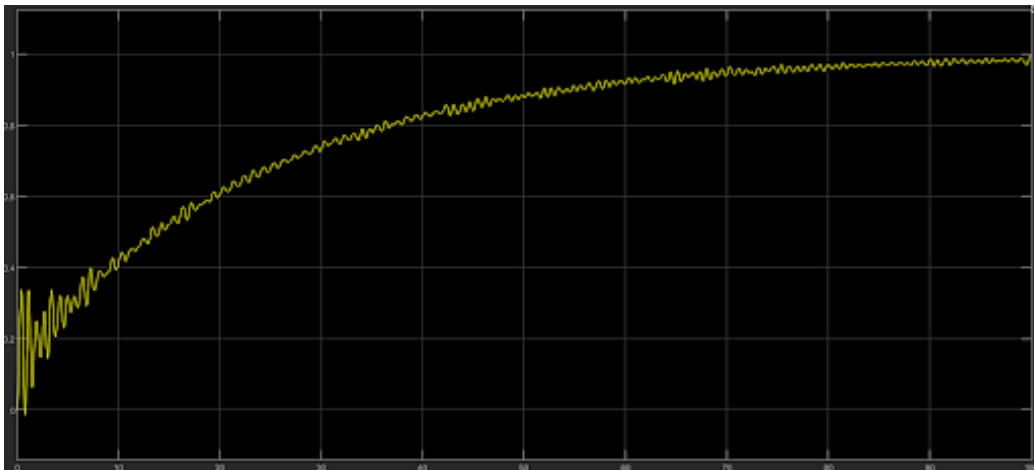


Figure 10:  $P=1.2$  et  $I=0.5$  (Optimisé par Matlab)

Nous remarquons que le temps de réponse est réglé ainsi la diminution des oscillations.



## 6 Conception de la maquette

### 6.1 Concept de base

Dans sa forme la plus simple, un segway est composé de trois éléments:

- un moyen de déterminer l'angle entre le segway et la verticale.
- une motorisation permettant un déplacement horizontal.
- un algorithme de commande qui réalise un asservissement en position.

Pour notre application, on se focalise d'abord sur la stabilisation autonome du segway et le déplacement de celui-ci est prévu dans un second temps. Le but est de développer un MVP (Minimum Viable Product) avant d'augmenter la complexité du projet par la suite pour le pousser un peu plus loin.

### 6.2 Architecture matérielle

Notre planification pour la conception du segway a quelque peu changée depuis le rapport du S6, en effet nous nous sommes rendus compte que l'utilisation d'une raspberry pi pour gérer l'asservissement ne sera pas possible à cause de l'architecture du processeur qui n'est pas temps réel. Cela rends toute tentative de commande par ce biais irréalisable.

Nous avons donc fait le choix d'abandonner le code existant pour passer sur une architecture d'Arduino utilisant FreeRTOS comme on a pu l'utiliser dans le cadre du module de Systèmes Temps Réels. Nous détaillerons les implications de ce changement dans la partie 6.3.

En ce qui concerne la mécanique du segway, nous utilisons toujours deux moteurs pas à pas pour entraîner les deux roues du segway. Ceux-ci sont fixés dans un châsis réalisé par impression 3D.

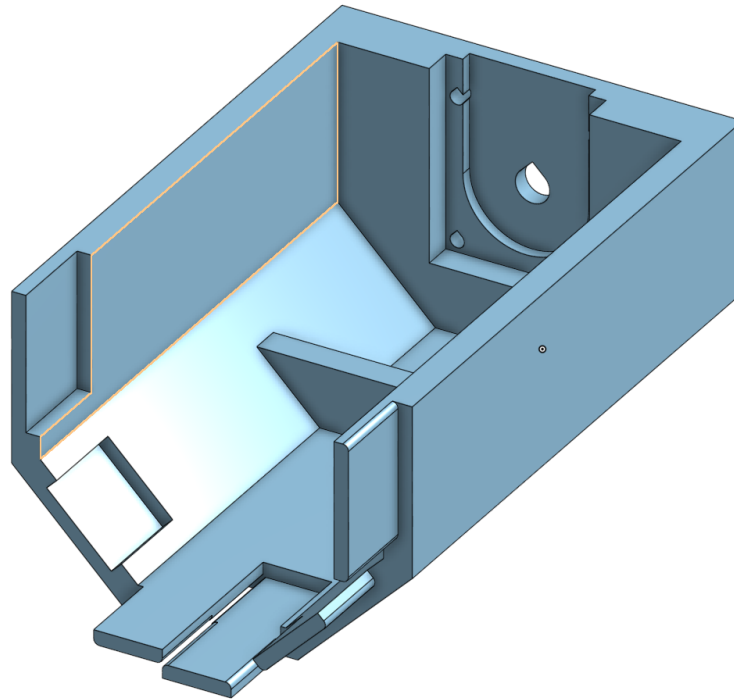


Figure 11: Modèle d'un demi châsis

L'électronique ayant beaucoup changé au cours du projet, il n'est pas prévu de fixations pour les composants dans le châsis, mais il reste possible de les y déposer pour fermer le segway.

### 6.3 Architecture logicielle

Comme discuté en 6.2 nous avons complètement repris la conception logicielle à zéro avec FreeRTOS, le but étant de créer des processus pour :

- 1 Lire la mesure d'angle de l'inclinomètre via une connexion I2C.
- 2 Calculer la commande à appliquer aux moteurs grâce aux algorithmes discutés en 5
- 3 Réaliser la commande des moteurs

Le processus 1 peut être facilement dérivé des exemples de lecture d'angle du MPU 6050 disponible avec la bibliothèque Arduino associée.

En revanche un problème apparaît pour la commande des moteurs, en effet bien que la commande des moteurs pas à pas permette de facilement faire

tourner le moteur d'un certain nombre de pas, cette commande devient difficile lorsque l'on veut réaliser des mouvements de quelques pas à intervalle réguliers.

Avec ce type de commande, il semble que le moteur ne soit pas alimenté suffisamment longtemps pour réaliser la séquence de commande permettant d'effectuer un pas. En effet les temps d'activation requis pour avoir une commande propre du moteur sont très précis et il semble que l'Arduino n'arrive pas à suivre avec un temps d'échantillonnage assez faible pour avoir une commande efficace. Et ce même avant de l'intégrer dans l'architecture FreeRTOS avec les autres processus qui prendront du temps processeur que l'on a déjà pas.

Le processus 2 de calcul de commande se base sur la représentation en transformée en Z de la chaîne d'asservissement développée en 5. La conception de celui-ci a été retardée par les problèmes survenus dans la commande des moteurs et reste donc à réaliser.

## 6.4 Changements à apporter

Il semble nécessaire pour régler les problèmes décrits en 6.3 de revoir à nouveau l'architecture pour utiliser des contrôleurs plus avancés pour les moteurs afin de réussir à les commander assez rapidement pour avoir un asservissement correct. Cela pourra passer par l'utilisation de contrôleurs différents ou bien par la séparation physique des parties de commande et de calculs par l'utilisation par exemple d'un microcontrôleur dédié à la commande de chaque moteur que l'on pourra coder en assembleur pour réduire le temps d'exécution du C. Plus de recherches à ce sujet seront nécessaires au S8.

## 7 Conclusion

Durant ce semestre, nous avons pu finaliser la modélisation de notre mini segway adapté à nos besoins, néanmoins nous n'avons pas réussi à implémenter les composants sur notre maquette. En effet, nous avons découvert au cours de nos essais sur le code que le moteur pas à pas n'est pas adapté, la commande avec un stepper n'est pas viable à cause du pas du temps (système trop lent et pas temps-réel). Un moteur à courant continu serait plus adapté dans notre cas.

Nous aurions dû choisir un moteur à courant continu mais comme nous avons choisi le moteur pas à pas depuis le S6 en trouvant un code sur internet basé sur ce type de moteur. Quand nous serons amenés à de nouveau travailler sur un projet, il pourrait être intéressant de mieux vérifier les ressources à disposition.

Finalement, nous avons réussi à réaliser:

- D'importantes recherches bibliographiques
- La compréhension du modèle de commande et de correction
- La modélisation sur matlab
- Conception 3D d'un système complet
- Etude approfondie de la commande d'actionneurs par microcontrôleur
- Prise en main du logiciel Matlab et Simulink
- Câblage des différents composants

Si nous devions continuer le projet, nous changerions le moteur pas à pas par un moteur à courant continu et nous attarderons aussi sur l'optimisation de la maquette physique.

Pour finir, et après une réflexion avec tous les membres de l'équipe, nous avons décidé de changer de sujet le semestre prochain. De plus que, deux membres de notre groupe partent à la filière SC (Systèmes communicants) et donc nous avons privilégié de travailler sur un nouveau projet avec des nouvelles notions.