

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département d'Automatique

Laboratoire de Commande des Processus

Laboratoire CENTRELEC R&D

Mémoire de projet de fin d'études

pour l'obtention du diplôme d'Ingénieur d'État en Automatique

Modélisation et Réalisation d'un Robot Auto Équilibrant (Self Balancing Robot)

OULARBI Housseme Eddine

Présenté et soutenu publiquement le (10/07/2019)

Composition du jury :

Président	M. Mohamed TADJINE	Prof.	ENP
Promoteur	M. Messaoud CHAKIR	MC/B.	ENP
Promoteur	M. Abdelmoumen LARFI	Ingénieur.	CENTRELEC
Examineur	M. Omar STIHI	MA.	ENP

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique



Département d'Automatique

Laboratoire de Commande des Processus

Laboratoire CENTRELEC R&D

Mémoire de projet de fin d'études
pour l'obtention du diplôme d'Ingénieur d'État en Automatique

Modélisation et Réalisation d'un Robot Auto Équilibrant (Self Balancing Robot)

OULARBI Housseme Eddine

Présenté et soutenu publiquement le (10/07/2019)

Composition du jury :

Président	M. Mohamed TADJINE	Prof.	ENP
Promoteur	M. Messaoud CHAKIR	MC/B.	ENP
Promoteur	M. Abdelmoumen LARFI	Ingénieur.	CENTRELEC
Examineur	M. Omar STIHI	MA.	ENP

Dédicace

Je dédie ce travail à mes chers parents et grands parents,

A ma famille,

A mes frères et soeurs,

A mes amis

*Et à tous ceux qui ont fait confiance en mes capacités pendant
notre parcours.*

Remerciements

Je remercie Dieu de m'avoir donné courage, santé et soutien durant les moments les plus difficiles.

Je souhaite exprimer ma sincère gratitude à mes conseillers, **M. Messaoud CHAKIR** et **M. Mohamed TADJINE**, pour avoir assumé la lourde responsabilité de m'encadrer, de m'avoir orienté et conseillé tout au long de ce travail ainsi pour la confiance qu'ils m'ont accordée.

Je souhaite aussi remercier mes professeurs d'Automatique et tous les enseignants pour les connaissances qu'ils m'ont transmises, leur disponibilité et leurs efforts.

Je remercie également **M. Abdelmoumen LARFI** pour son temps, ses précieux conseils et pour m'avoir donné la chance de travailler sur un projet d'une si grande importance.

Je suis également reconnaissant à toute l'équipe de **CENTRELEC** et plus particulièrement à **M. Hicham BOUTOUCHE** et **M. Abdelkrim TALEB** pour leur disponibilité, précieux conseils et leurs encouragements.

Je remercie sincèrement les membres du jury pour l'honneur qu'ils m'ont fait en acceptant d'évaluer mon projet.

Dernier mais pas des moindres ; je voudrais remercier mes parents, familles et amis pour leur soutien constant. Rien de tout cela n'aurait pu arriver sans eux.

ملخص

الغرض من هذا المشروع هو دراسة روبوت أحادي العجلة ذو توازن تلقائي. أولاً، تم إنشاء نموذج غير خطي وثلاثي الأبعاد. ثانياً، يتم تطبيق طريقتين للتحكم فيه بواسطة المتحكم الأمثل التريبيعي الخطي ومتحكم النمط الانزلاقي، والهدف هو تثبيت النظام وإبقائه في وضع عمودي، بالإضافة إلى أنه يمكن الروبوت من تتبع سلسلة من النقاط المتتالية. أخيراً، يتم شرح عملية بناء هيكل الروبوت لنتهي من خلال وصف الوحدات الإلكترونية اللازمة لتطبيق الدراسة النظرية.

الكلمات المفتاحية: الروبوت المتوازن، النواس المقلوب، المتحكم التريبيعي الخطي، متحكم النمط الانزلاقي، أردوينو، مقياس التسارع، الجيرو سكوب.

Abstract

The purpose of this project is to study a single-wheeled balancing robot. Firstly, a 3D nonlinear model is established using screw theory. Secondly, two methods of control are applied, the first one is the Linear quadratic controller (LQR) used to stabilize the system and keep it upright, plus, it enables it to follow a track of points. The second controller based on sliding mode is applied to fulfill the balancing at the equilibrium point. Finally, an experimental part is done about the robot's structure building process and finishing by a description of the used hardware modules.

Keywords : balancing robot, inverted pendulum, LQR control, sliding mode control, Arduino, accelerometer, gyroscope.

Résumé

Le but de ce projet est d'étudier un robot auto équilibrant sur une roue. Tout d'abord, un modèle 3D non linéaire est établi en utilisant la théorie des torseurs. Deuxièmement, deux méthodes de commande sont appliquées, la première est le régulateur linéaire quadratique (LQR) utilisé pour stabiliser le système et le maintenir en position verticale. De plus, il lui permet de suivre une séquence de points. Le second régulateur basé sur le mode glissant est appliqué pour réaliser le balancement au point d'équilibre instable. Enfin, une partie expérimentale est faite sur la conception du robot et se termine par une description des modules électroniques utilisés.

Mots-clés : robot auto équilibrant, pendule inversé, commande LQR, commande de mode glissant, Arduino, accéléromètre, gyroscope.

Table des matières

Table des figures

Liste des abréviations

Introduction	12
1 Historique de Ballbot	15
1.1 Introduction	16
1.2 Bref historique de ballbot	16
1.3 Modélisation	18
1.4 Commande	22
2 Modélisation du robot	26
2.1 Introduction	27
2.2 Discription du modèle	27
2.2.1 Géométrie	27
2.2.2 Entrées et sorties	28
2.3 Coordonnées	28
2.4 Hypothèses	29
2.5 Repères	29
2.6 Étude cinématique	31
2.6.1 Matrice homogène	31
2.6.2 Torseur cinématique	33
2.6.2.1 Configuration du vecteur des coordonnées généralisées	33
2.6.2.2 Configuration de torseur cinématique	33
2.6.3 Matrice jacobienne	34
2.7 Etude dynamique	35
2.7.1 Tenseur d'inertie	35

2.7.2	Énergie cinétique	36
2.7.3	Énergie potentielle	36
2.7.4	Équation d'Euler Lagrange	36
2.7.5	Contrainte de roulement	37
2.7.6	Simplification des équation de mouvement	39
2.8	Forces externes	40
2.8.1	Actionnement	40
2.8.2	Actionnement sur la sphère	41
2.8.3	Configuration des roues omnidirectionnelles	42
2.8.4	Actionnement sur le corps	43
2.8.5	Forces externes totales	44
2.9	Linéarisation	44
2.10	Simulation en boucle ouverte	45
2.11	Conclusion	48
3	Proposition de lois de commande	49
3.1	Introduction	50
3.2	Proposition de lois de commande et simulation	50
3.2.1	Commandabilité	51
3.2.2	Observabilité	51
3.3	Commande linéaire LQR	51
3.3.1	Conception du régulateur	52
3.3.2	Choix de Q et R	53
3.3.3	Test et simulation	55
3.3.3.1	Balancement du robot	56
3.3.3.2	Maintien de position	56
3.3.3.3	Commande point à point	57
3.4	Commande non linéaire : mode glissant	59
3.4.1	Le partitionnement du système	59
3.4.2	Conception du régulateur de balancement	60
3.4.2.1	Synthèse de la loi de commande	61
3.4.2.2	Choix de la surface de glissement	61
3.4.2.3	Condition d'existence du mode de glissement	62
3.4.2.4	Calcul de la commande	62

3.4.2.5	Le broutement (chattering)	64
3.4.2.6	Solution pour atténuer le phénomène de "chattering" . .	64
3.4.3	Test et simulation	66
3.5	Etude comparative de deux régulateurs	67
3.6	Singularités	69
3.7	Conclusion	70
4	Conception et implémentation	71
4.1	Introduction	72
4.2	Conception mécanique du robot	72
4.2.1	Forme du Robot	72
4.2.2	Les étagères	73
4.2.3	Les roues omnidirectionnelles	74
4.2.4	Les supports des moteurs	75
4.2.5	Les moyaux des roues	77
4.2.6	Assemblage du robot	78
4.3	Modules électronique	80
4.3.1	Le microcontrôleur Arduino 2560	80
4.3.2	Moteurs	80
4.3.3	Les pilotes des moteurs	81
4.3.4	Capteurs	82
4.3.4.1	Gyroscope	82
4.3.4.2	Accéléromètre	83
4.3.4.3	Magnétomètre	83
4.3.4.4	SparkFun 9DoF Razor IMU M0	83
4.4	Simulation graphique	84
4.5	Conclusion	85
	Conclusion générale	86
	Bibliographie	88
	Annexes	91
.1	Annexe A : Paramètres	92

.2	Annexe B : Notions mathématiques	
	préliminaires	93
.2.1	Torsuer cinématique	93
.2.2	Torseur dynamique	95
.3	Annexe C : Résultats de la Modélisation	97
.4	Annexe D : Les plans 2D des pièces utilisées dans la conception	106

Table des figures

1	Segway (Segway Inc, 2009)	13
1.1	Types de ballbot	16
1.2	présentation des robots auto équilibrants	17
1.3	Modèles de ballbot	19
1.4	Structure du régulateur du ballbot de CMU[1]	22
1.5	Structure du régulateur du Rezero[2]	23
2.1	Représentation géométrique du modèle 3D [2]	27
2.2	Description des repères	30
2.3	Positionnement des roues par rapport le repère \mathcal{R}_r	43
2.4	Réponse libre du modèle 3D non linéaire pour un angle initiale de tan- gage de 0.01 rad	47
2.5	Comparaison entre les modèle NL et L	48
3.1	Système en boucle fermée avec un régulateur LQR	52
3.2	Simulation en boucle fermé de la réponse du système à une condition initiale de $\theta = 0.15$ rad avec des valeurs différentes de ρ	54
3.3	Simulation en boucle fermée du couple du système τ_2 à une condition initiale de $\theta = 0.15$ rad avec des valeurs différentes de ρ	54
3.4	Présentation du système complet sur Simulink	55
3.5	Test de balancement avec la commande LQR	56
3.6	Test de positionnement à l'endroit initial	57
3.7	L'évolution de x_k et y_k par rapport une référence	58
3.8	Suivi d'une séquence des points	58
3.9	Surface de glissement	61
3.10	Structure de la commande par mode glissant	63
3.11	Phénomène de broutement	64

3.12	La fonction sigmoïde	65
3.13	Test de balancement avec SMC en injectant une condition initiale	66
3.14	Test de balancement avec SMC en présence des différentes perturbation	67
3.15	Rejet de perturbation pour les deux régulateurs	68
3.16	Couple généré pour le rejet de perturbation par les deux régulateurs	69
4.1	Modèle 3D CAD du prototype du robot	72
4.2	La base en bas	73
4.3	L'étagère supérieur	74
4.4	Etagère usiné	74
4.5	Design 3D CAD de la roue	75
4.6	La roue omnidirectionnelle	75
4.7	Pièces du support	76
4.8	Design 3D CAD du support	76
4.9	Support usiné	77
4.10	Moyau	78
4.11	L'assemblage de la structure du robot	79
4.12	Structure réelle du robot	79
4.13	Carte Arduino Mega 2560	80
4.14	Moteur DCM50205D1000	81
4.15	Servo Drive DCS810	82
4.16	SparkFun 9DoF Razor IMU M0	84
4.17	Environnement VR Sink	85

Liste des abréviations

2D	à deux dimension
3D	à trois dimension
DDL	Degré De Liberté
CMU	Carnegie Mellon University
ETH	Eidgenössische Technische Hochschule Zürich
TGU	Tohoku Gakuin University
UA	University of Adelaide
L	Linéaire
NL	Non Linéaire
IMU	Inertia Measurement Unit
LQR	Linear Quadratic Regulator
MIMO	Multiple Input Multiple Output
PD	Proportionnel Dérivé
PI	Proportionnel Intégral
PID	Proportionnel Intégral Dérivé
CAD	Computer-Aided Design
MEMS	Micro-Electro-Mechanical Systems
FKE	Filtre de Kalman Etendu
SMC	Sliding Mode Controler

Introduction

Le concept d'un "Ballbot" est simple : c'est un robot qui balance sur une balle. La balle, par conséquent, agit comme une roue sphérique, permettant au robot de se déplacer dans n'importe quelle direction. Contrairement aux robots traditionnels, qui reposent sur un centre de masse bas et une base large pour rester en position verticale, le "Ballbot" doit s'équilibrer activement, ce qui permet une stabilité dynamique. Cela permet potentiellement de développer des robots à taille humaine avec les mêmes performances en stabilité que l'homme.

Ce projet vise à concevoir et à construire un robot auto équilibrant sur une sphère. Le développement de ce robot est un projet complet, comprenant le calcul des équations de mouvement généralisées, la conception et la construction du robot, la conception et la mise en œuvre d'un régulateur et d'un logiciel approprié permettant d'équilibrer le système et de le tester.

Motivation

Pour les robots traditionnels à plusieurs roues, seul un léger décalage de la position du centre de gravité peut rendre le robot instable. Cela peut être corrigé en abaissant le centre de gravité du robot afin de réduire le bras de levier. Toutefois, cela se fait généralement au prix d'un poids important ajouté au système. En plus, la vitesse à laquelle le robot peut se déplacer est généralement limitée, de peur que son élan le renverse.

Un autre facteur est le fait que l'entraînement traditionnel oblige le robot à tourner pour changer de direction. En fonction de l'environnement, de la taille et de la conception mécanique du robot, cette tâche peut parfois s'avérer impossible, ce qui peut entraîner le blocage du robot.

Une solution plausible à ce problème, développée dans ce projet, est la stabilité dynamique. C'est-à-dire que la théorie de commande est utilisée pour garantir le positionnement vertical du robot, sans qu'il soit nécessaire de compter sur la stabilité statique. La stabilité dynamique a été utilisée de la manière la plus connue sur l'unité de transport pour une personne, le Segway. Le Segway (Figure 1) possède deux roues alignées axialement et utilise la solution au problème de commande du « pendule inversé ». Cependant, le Segway repose sur la stabilité statique dans sa direction transversale - une contrainte surmontée par le concept de ballbot qui vise à produire un robot qui soit dynamiquement stable dans toutes les directions. En outre, le domaine de la stabilité dynamique constitue une application intéressante de la théorie de commande.



FIGURE 1 – Segway (Segway Inc, 2009)

Commander le ballbot est un défi. Tout d'abord, le robot est sous-actionné. Les systèmes sous-actionnés ont un nombre d'actionneurs inférieur au degré de liberté . Par conséquent, les mêmes actionneurs doivent être utilisés pour plusieurs degrés de liberté. Le sous-actionnement du robot résulte de la propriété selon laquelle le robot doit commander la position de la balle $(x, y) \in \mathbb{R}^2$ et la rotation du corps, soit au total 5 degrés de liberté, en utilisant seulement 2 à 3 actionneurs.

Positionnement du problème

La thèse comprend une partie théorique et une partie pratique. La partie théorique peut être scindée entre la dérivation d'un modèle 3D, la conception d'un régulateur.

Le régulateur conçu doit pouvoir :

1. Stabiliser le robot en position verticale
2. Stabiliser le robot à une certaine position sur le sol

Pour la modélisation du robot, un modèle 3D sera dérivé en utilisant les torseurs (Screw Theory). Il en résulte un modèle exprimé en vitesses angulaires, translations et vitesses linéaires.

La partie pratique de ce travail comprend la construction du robot (conception 3D avec SolidWorks et l'usinage des pièces du modèle) ainsi que la mise en œuvre des algorithmes de commande proposés (Partie électronique).

Structure du rapport

La thèse est organisée comme suit :

Chapitre 1 : Historique du ballbot : Ce chapitre d'introduction passera brièvement en revue les travaux antérieurs sur les robots auto équilibrants "Ballbot".

Chapitre 2 : Modélisation du robot : Ce chapitre décrit la création d'un modèle 3D complet à l'aide de la théorie des torseurs (Screw Theory).

Chapitre 3 : Proposition de lois de commande : Ce chapitre propose des lois de commande pour commander la position du robot et maintenir la stabilité verticale. L'efficacité des régulateurs proposés est montrée dans la simulation.

Chapitre 4 : Conception et implémentation : Ce chapitre décrit la conception mécanique de la structure du robot ainsi que les modules électroniques à appliquer.

Conclusion générale : Cette thèse est conclue par des conclusions et des recommandations.

Chapitre 1

Historique de Ballbot

1.1 Introduction

Ce chapitre couvre les ballbots tels que trouvés dans les articles et les publications. Tout d'abord, un aperçu des robots les plus connus sera donné, suivi par la modélisation et le moyen de commande de ces robots.

1.2 Bref historique de ballbot

Dans cette section sont présentés les robots auto équilibrants les plus connus. On connaît principalement deux types de ces robots en ce qui concerne le mécanisme d'actionnement; le robot est actionné en utilisant soit un entraînement basé sur le fonctionnement d'une souris (Inverted mouse-ball drive) ou trois roues omnidirectionnelles,(Figure 1.1)

Les quatre meilleurs instituts renommés dans le domaine des robots de ce type sont l'Université Carnegie Mellon (CMU) avec 'Ballbot', l'Université Tohoku Gakuin (TGU) avec BallIp, Eidgenössische Technische Hochschule Zürich (ETH) avec Rezero et enfin le National Université Chung Hsing (NCHU) avec deux ballbots sans nom.

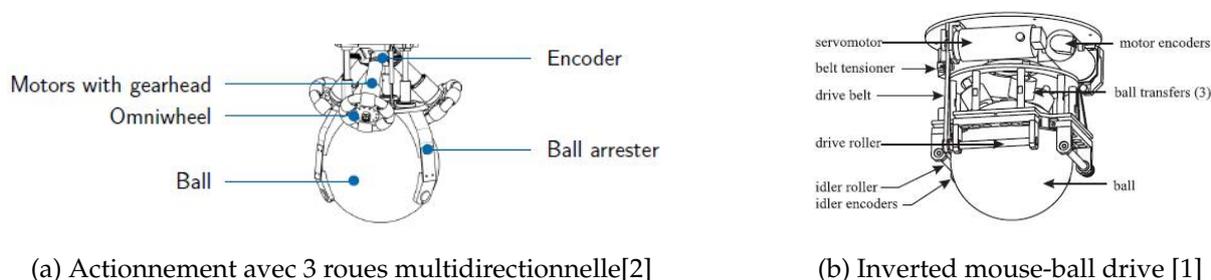
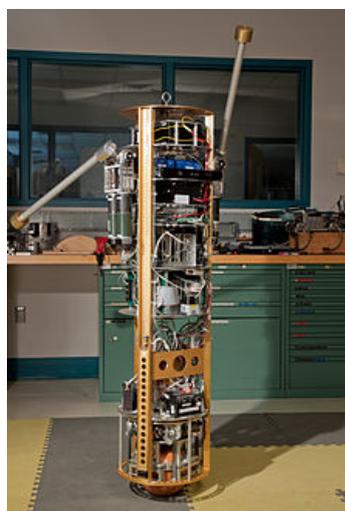


FIGURE 1.1 – Types de ballbot

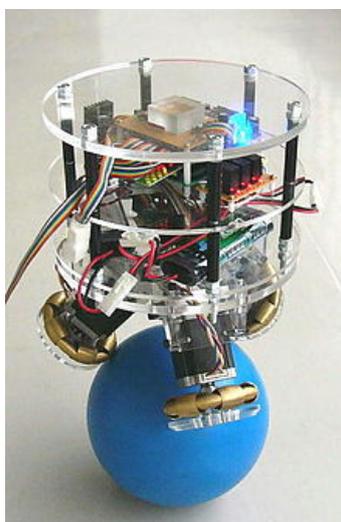
Université Carnegie Mellon L'un des premiers robots auto équilibrants sur une sphère a été développé par l'Université Carnegie Mellon [1], voir la figure 1.2.a. Ce robot utilise le principe de fonctionnement inversé d'une souris pour rester en équilibre , voir Figure 1.1.b. Une version évoluée a ,par la suite, été introduite, qui comprenait un mécanisme d'entraînement en lacet capable de réorienter le corps par rapport au mécanisme d'entraînement[3]. En outre, le robot est également équipé de trois pieds rétractables, ce qui lui permet d'être statistiquement stable. Une IMU fournit des angles et des vitesses corporelles filtrées par filtre de Kalman. Enfin, les quatre moteurs qui entraînent la balle sont équipés d'encodeurs pour mesurer la

rotation de la balle.

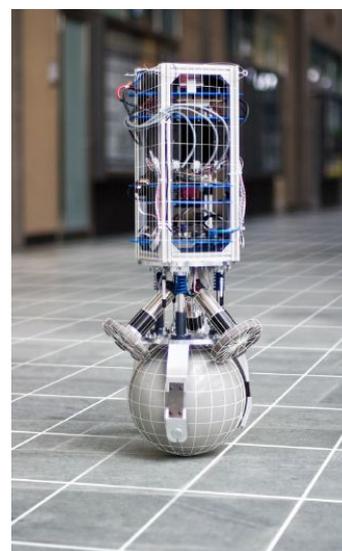
Université Tohoku Gakuin En 2008, l'Université Tohoku Gakuin a présenté BallIP [4], illustré sur la figure 1.2.b. Contrairement au ballbot de CMU, la balle est entraînée par trois roues omnidirectionnelles, chacune actionnée par un moteur pas à pas. En outre, le robot est équipé de deux ensembles de gyroscopes et d'accéléromètres, qui fournissent les angles de tangage en combinant les deux signaux à l'aide d'un filtre numérique du premier ordre. Les vitesses angulaires sont directement obtenues à partir des gyroscopes.



(a) Ballbot de CMU [3]



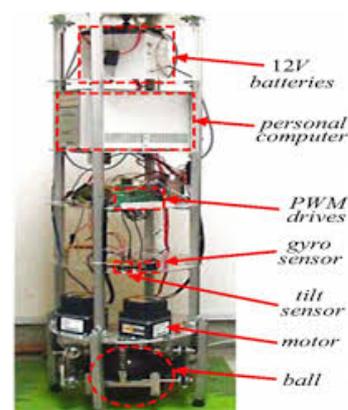
(b) BallIP de TGU [4]



(c) Rezero de ETH [2]



(d) Deuxième Ballbot de NCHU [5]



(e) Ballbot de NCHU [6]

FIGURE 1.2 – présentation des robots auto équilibrants

ETH Zurich Semblable à BallIP, Rezero de ETH Zurich dispose également de trois roues omnidirectionnelles, entraînées par des moteurs sans balai à courant continu avec des engrenages ayant chacun leur propre encodeur, voir Figure 1.2.c. De plus, le Rezero est équipé d'un IMU, qui mesure les trois angles d'Euler et les vitesses angulaires du corps. La balle est une sphère creuse en aluminium recouverte fibres synthétiques à haute résistance[2].

Université nationale Chung Hsing À l'Université nationale Chung Hsing, les deux types des robots ont été développées. Le premier robot possède une commande souris-boule inversée [6, 7, 8], similaire au Ballbot de CMU, comme le montre la figure 1.2.e. Le second robot est actionné par trois roues omnidirectionnelles [5, 9], similaires à Rezero et BallIP, comme illustré sur la figure 1.2.d. Les robots sont équipés d'un capteur d'inclinaison, d'un gyroscope et les moteurs sont équipés d'un encodeur.

1.3 Modélisation

Dans cette section, nous examinons les différentes approches pour modéliser un ballbot expérimenté dans les travaux précédents. Le plan vertical xz est également appelé plan sagittal et le plan vertical yz , plan coronal. L'angle d'inclinaison (pour les plans sagittal et coronal) est défini comme l'angle entre le corps et l'axe z . Les angles d'inclinaison sont souvent appelés angles de tangage et de roulis.

L'angle de lacet est défini comme la rotation autour de l'axe z en coordonnées corporelles.

Les hypothèses communes pour la modélisation du robot sont :

- Le robot est simplifié à un cylindre rigide et le ballon à une sphère rigide.
- Il n'y a pas de glissement entre la balle et le sol et la balle et le mécanisme d'entraînement.
- Les dynamiques dans les plans sagittal et coronal sont identiques.
- La balle ne tourne pas autour de l'axe z .
- La balle ne se déforme pas.
- La dynamique de l'actionneur est rapide, c'est-à-dire que l'entrée n'a pas de retard.
- Le sol est plat.
- Les roues sont toujours en contact avec la balle.

Pour les modèles 2D, on suppose que la dynamique des deux plans est indépendante et que, par conséquent, le système peut être découpé en tant que deux modèles planaires indépendants. CMU : [1, 3], TGU : [4, 10], ETH : [2], NCHU : [6, 8, 5, 9]. Cette hypothèse est basée sur le fait que les termes de couplage entre les plans verticaux contiennent des produits sinus. Par conséquent, si le système est seulement soumis à de petits angles de pas, les termes de couplage sont négligeables et la dynamique plane peut donc être découpée [3].

Outre les deux modèles 2D découplés, des modèles 3D ont également été proposés; CMU : [11], ETH : [2] et NCHU : [7, 12] ainsi que dans [13] et [14]. La visualisation des modèles 2D et 3D pour un ballbot est illustrée dans la figure 1.3 respectivement [13]. Le principal avantage de deux modèles 2D découplés identiques réside dans sa simplicité relative. Cependant, il présente les limitations et inconvénients suivants. Tout d'abord, les termes de couplage sont négligés. Ces effets deviennent particulièrement importants à des vitesses plus élevées. [2, 15]. Deuxièmement, la dynamique naturelle du lacet est complètement négligée [13]. Enfin, l'ajout de bras ou de charges asymétriques rendrait un modèle 2D encore moins précis [13]. Cependant, le coût d'un modèle 3D complet est la complexité ajoutée, et la condition antidérapante entraîne une contrainte non holonomique, par opposition à une contrainte holonomique pour le modèle 2D.

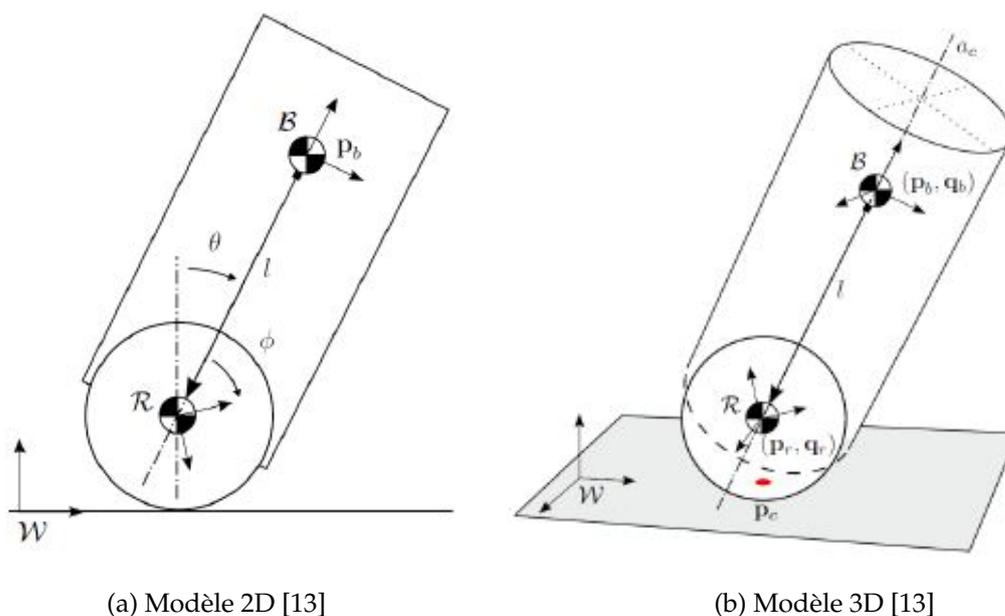


FIGURE 1.3 – Modèles de ballbot

En général, les équations d'Euler-Lagrange sont utilisées pour dériver les équations de mouvement. Cependant, le choix des coordonnées généralisées varie souvent entre les différents robots. Sauf indication contraire, la mécanique lagrangienne est utilisée pour les équations dynamiques.

Le modèle du ballbot de CMU a l'angle d'inclinaison du corps du robot ϕ et le positionnement angulaire de la balle θ comme coordonnées généralisées. Le positionnement angulaire de la balle est choisi de manière à ce que la position horizontale de la balle par rapport à un référentiel inertiel fixe est donné par $x = r(\theta + \phi)$, où r est le rayon de la balle. L'avantage de choisir la coordonnée de positionnement de la balle θ est qu'elle correspond directement aux lectures de l'entraînement de la balle par l'encodeur. Les équations d'Euler-Lagrange sont utilisées pour dériver les équations dynamiques. Le frottement a été initialement modélisé comme un frottement uniquement visqueux, afin d'éviter une dynamique discontinue causée par le frottement de Coulomb [1]. Cependant, dans le travail amélioré, le frottement de Coulomb fut également introduit [3].

Pour la modélisation de Rezero, un modèle 2D planaire et un modèle 3D ont été proposés [2]. Comme hypothèse supplémentaire, le frottement est négligé, car un frottement non continu (de Coulomb) résulterait en des équations plus complexes. Le modèle 3D se compose de la balle, de 3 roues omnidirectionnelles et du corps du robot comprenant les moteurs et les réducteurs. Les coordonnées généralisées sont l'intégrale de la vitesse angulaire de la balle, c'est-à-dire la position angulaire parcourue, et trois angles d'Euler, des angles d'orientation spatiale du corps. Ces coordonnées sont choisies de manière à pouvoir être mesurées directement par les capteurs du système.

La modélisation pour les deux différents ballbots de NCHU est similaire. Cependant, il y a quelques différences. Pour le robot à 3 roues omnidirectionnelles, les coordonnées généralisées sont l'angle d'inclinaison et les positions angulaires des moteurs pour les deux plans verticaux [5, 9]. Pour l'entraînement souris / balle inversé, les coordonnées généralisées sont l'angle d'inclinaison du robot et les angles de rotation de la balle pour les deux plans verticaux [6, 7].

Pour les modèles 3D de [7] et [12], l'orientation du cylindre est décrite avec un angle de tangage et de lacet et sa position planaire.

Exceptionnellement, dans [12], la mécanique newtonienne est utilisée pour dériver un modèle.

Pour tous les travaux de NCHU, sauf [7], le frottement est modélisé comme une somme du frottement de Coulomb et du frottement visqueux entre la balle et le sol.

Inal et tous. [13] ont proposé un modèle 3D complet, estimant qu'un modèle 2D ne prend pas en compte le mouvement de lacet résultant du mouvement de roulement de la roue et des effets inertiels couplés. De plus, il a été constaté que, pour un modèle 2D, la condition de non-glissement entraîne une contrainte holonomique, alors que pour le modèle 3D, cette contrainte est non holonomique.

Trois repères sont utilisés : un repère inertiel, un repère lié au corps du robot et un autre lié à la balle. Au lieu d'angles d'Euler, les quaternions étaient utilisés pour représenter la rotation. On suppose que la rotation relative en lacet du corps par rapport la balle est empêchés à cause de frottement qui n'est pas directement commandé. En outre, il est supposé que les frottements entre la balle et le sol l'empêchent de glisser dans les directions horizontale et en lacet. Contrairement aux autres modèles, les quaternions représentent les rotations de la balle et du robot.

Dans Lotfiani et tous. [14] Un modèle 3D complet a également été proposé. Encore une fois, trois repères sont utilisés : un repère inertiel, un lié au corps et un lié à la sphère. Les coordonnées généralisées sont deux ensembles de trois rotations consécutives, un ensemble pour le corps et la sphère, résultant en 6 coordonnées généralisées au total. Une hypothèse supplémentaire est introduite selon laquelle le moment résultant autour de l'axe longitudinal du cylindre est toujours égal à zéro. Cela donne deux contraintes non holonomiques. De plus, un frottement visqueux est modélisé entre les roues et la sphère. Le système est ensuite modélisé à l'aide des équations d'Euler-Lagrange et des multiplicateurs de Lagrange. Les multiplicateurs de Lagrange sont éliminés en utilisant l'espace nul de la matrice de contraintes A .

1.4 Commande

Dans cette section, les méthodes de commande proposées pour les ballbots sont examinées.

CMU : La commande initialement proposée consistait en un régulateur à deux boucles pour chaque plan vertical, composé d'un régulateur PI en tant que boucle interne et d'un régulateur LQR en tant que boucle externe [1]. Ces régulateurs planaires sont découplés et indépendants les uns des autres. La sortie de la boucle externe est une vitesse désirée de la balle, qui est ensuite suivie par la boucle interne. Le but de la boucle interne est de réduire l'effet du frottement statique et dynamique non modélisé, tandis que la boucle externe est utilisée pour suivre une référence donnée.

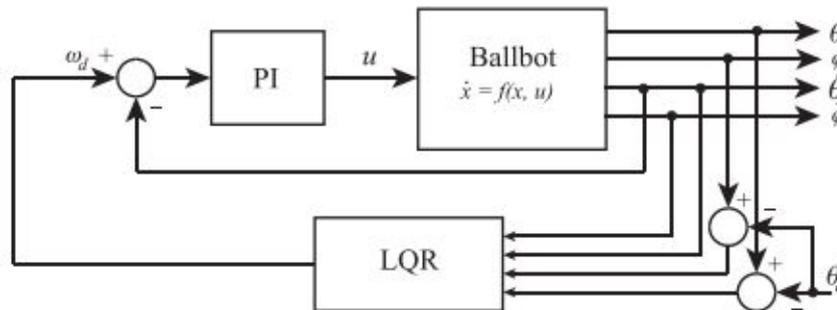


FIGURE 1.4 – Structure du régulateur du ballbot de CMU[1]

Dans [3], un schéma de commande amélioré est présenté, consistant à nouveau en deux régulateurs indépendants à deux boucles pour chaque plan. La boucle interne est utilisée pour l'équilibrage et la boucle externe pour la commande de position. Cependant, l'utilisation de ce régulateur pour suivre la trajectoire entraîne un mouvement saccadé. Cela était dû au fait que les boucles interne et externe se combattent [1], car le système est sous-actionné. Par conséquent, un planificateur de trajectoire hors ligne est proposé [3]. Ce régulateur utilise la dynamique naturelle pour le mouvement souhaité de point à point.

TGU : La commande du robot BallIP consiste en deux régulateurs indépendants pour les deux plans verticaux. Chaque régulateur est formé par la somme de deux PD, l'un pour l'angle de tangage et l'autre pour la position [4, 10].

Rezero : Pour la commande de Rezero, un régulateur linéaire LQR a été utilisé. Pour améliorer les performances de suivi des points de consigne non nuls, un terme de feed-forward a été utilisé [2]. Un régulateur de vitesse non linéaire a été proposé en utilisant la planification de gain (gain scheduling), c'est-à-dire la conception de plusieurs régulateurs linéaires à différents points de fonctionnement.

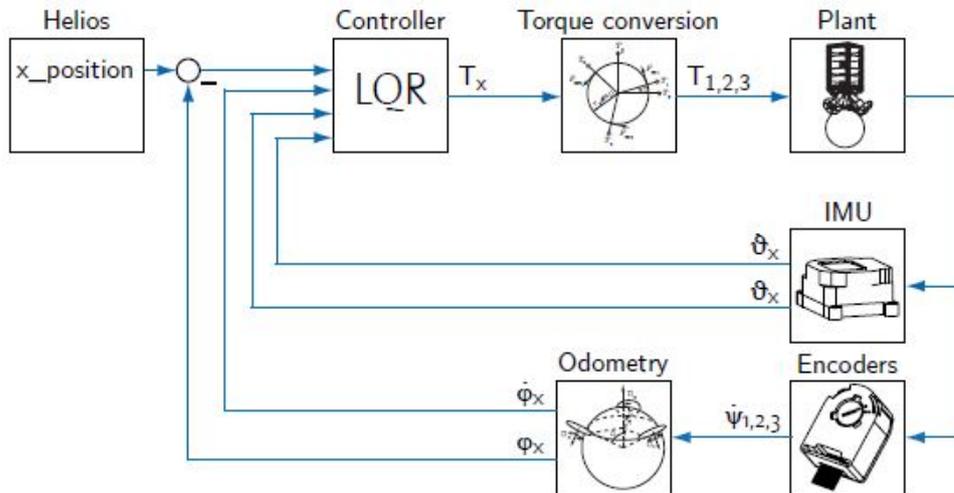


FIGURE 1.5 – Structure du régulateur du Rezero[2]

En raison de la grande dimension du système, de la mémoire limitée et de la puissance de calcul, seules les non-linéarités causées par la vitesse sont prises en compte, c'est-à-dire la linéarisation autour de points de consigne ayant des vitesses différentes. Contrairement au régulateur linéaire, ce régulateur est capable de commander de manière stable le système à des vitesses supérieures (supérieures à 2 m/s), car à ces vitesses également, les conditions de couplage ont une influence plus significative. En conséquence, le régulateur non linéaire est mieux même pour la poursuite des séquences complexes par rapport au régulateur linéaire.

En 2013, un filtre de Kalman étendu (FKE) pour estimer les états du Rezero [16] a été proposé. Il utilise les données des accélérateurs, du gyroscope et des encodeurs pour estimer l'état complet des robots, c'est-à-dire leur positionnement vertical, leur position et leur vitesse. Cela contrairement aux robots BallIP, ballbot de CMU et les robots de l'Université nationale Chung Hsing, dans lesquels le positionnement vertical et la position sont estimés séparément.

Les états du FKE sont la position de l'IMU, sa vitesse et la matrice de rotation du corps, tous exprimés dans un repère lié au corps. Des équations de prédiction discrètes

sont dérivées, qui relient les sorties du capteur aux états du FKE. Pour l'étape de mise à jour du FKE, une mesure virtuelle de la vitesse de la balle est utilisée, ce qui est définie comme la différence entre l'équation reposant sur les états filtrés et celle s'appuyant sur les lectures des capteurs à roues multiples.

Il a été prouvé que la position absolue et la rotation autour de l'axe de gravité sont non observables. Cependant, la déviation de ces états n'influence pas les autres états prédits.

NCHU : Tout au long des travaux à la NCHU, les régulateurs proposés adoptent une commande en mode glissant sans [7] et avec des incertitudes paramétriques [6, 8, 5], un contrôle LQR [9] et une approche de commande par plusieurs boucles [12].

Dans [12], un régulateur double à deux boucles a été proposé pour l'auto-équilibrage et la commande de position, Un à deux boucle commande l'angle d'inclinaison et la position vers l'avant, un autre à deux boucles commande la vitesse de et le mouvement latitudinal. Chaque régulateur possède une boucle interne composée d'un régulateur PI et d'une boucle externe avec un régulateur LQR, similaire à [1]. Un terme "feed-forward" supplémentaire a été utilisé pour surmonter le frottement.

Dans [9], un seul régulateur LQR a été proposé. La commande de maintien de la position et d'équilibrage est combinée dans un problème de régulateur avec comme états l'angle d'inclinaison, sa vitesse, l'erreur entre la position actuelle et la position désirée ainsi que sa dérivée.

Dans [7], il est proposé un régulateur hiérarchique en mode glissant basé sur "backstepping" consistant en une couche interne et une couche externe. La couche interne a une fonction de glissement pour l'angle d'inclinaison et pour la direction du mouvement. La couche externe est définie comme une combinaison convexe des deux surfaces de glissement internes.

Dans [6, 8, 5], le régulateur de mode glissant basé sur "backstepping" a été augmenté par l'ajout d'un réseau de neurones à intervalle récurrent de type 2 (RIT2FNN) afin de pouvoir gérer les incertitudes.

Pour que les erreurs auxiliaires de "backstepping" convergent à zéro, la procédure d'un régulateur hiérarchique en mode glissant est utilisée, comme présenté par [17].

Autres : Dans [18] un régulateur flou a été proposé pour un ballbot. Ce régula-

teur a deux boucles, interne et externe, similaire à celle de [1, 3], et chaque boucle est constitué d'un régulateur flou avec 25 règles chacun.

Dans [14], un régulateur basé sur la méthode "computed torque" et autre basé sur le mode glissant ont été proposés pour la commande d'équilibrage et ont été vérifiés par simulation. Le sous-actionnement du système a été traité à l'aide d'un planificateur de trajectoire floue. Ce régulateur flou utilise les variations angulaires de la sphère et ses dérivés comme entrée et renvoie les angles désirés du cylindre. Ces angles désirés sont utilisés comme entrée de commande pour les 2 types du régulateur mentionnés.

Chapitre 2

Modélisation du robot

2.1 Introduction

La représentation du système réel du robot en tant que modèle planaire implique de multiples inconvénients (implication de conversion). Cependant, l'approche décrite dans ce chapitre prend en compte la géométrie et les paramètres tridimensionnels complets. Avec ce modèle, tous les effets de couplage sont pris en compte et aucune conversion n'est requise. Le modèle 3D du robot sera dérivé à l'aide de la théorie des torseurs (screw theory) et les équations d'Euler-Lagrange. Premièrement, les repères et hypothèses utilisés seront définis. Deuxièmement, la cinématique du robot sera discutée et enfin, les équations dynamiques complètes seront dérivées.

2.2 Description du modèle

2.2.1 Géométrie

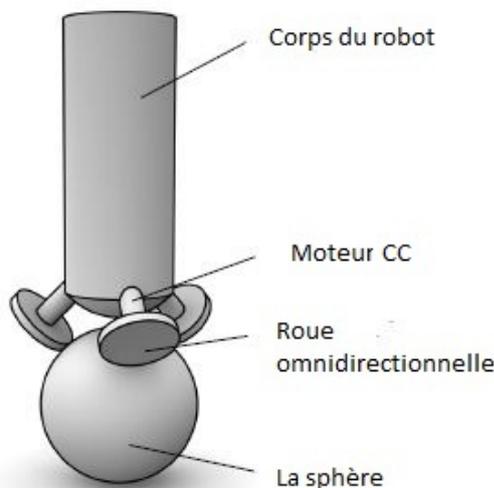


FIGURE 2.1 – Représentation géométrique du modèle 3D [2]

Le modèle 3D proposé, illustré à la figure (2.1), comprend cinq corps solides :

- 1 balle.
- 3 roues omnidirectionnelles.
- 1 corps avec 3 moteurs.

La balle est toujours en contact avec le sol en un point arbitraire. Trois roues omnidirectionnelles relient le système supérieur à la balle en trois points à tout moment.

Chacune de ces roues est fixée sur un moteur. La liaison entre les roues et la balle est holonomique et la roue ne peut appliquer des forces à la balle que dans le sens tangentiel de sa rotation.

2.2.2 Entrées et sorties

Les entrées du système sont les trois couples τ_1, τ_2, τ_3 générés par les moteurs. Les capteurs du système sont choisis de manière à ce que tous les états puissent être mesurés directement. Par conséquent, chaque sortie représente un état séparément. Ce système a plusieurs entrées et sorties, il est donc considéré un système MIMO.

2.3 Coordonnées

Pour dériver les équations du mouvement, les coordonnées doivent être définies. Les coordonnées sont définies en fonction du degré de liberté (DDL) du système.

Ce système peut être décrit avec les cinq degrés de liberté suivants :

- 2 DDL pour la position de la balle,
- 3 DDL pour la rotation du corps.

Dans la première phase, l'étude sera faite en utilisant toutes les coordonnées généralisées données par le vecteur q comme suivant :

$$q = \left(\alpha(t) \quad \beta(t) \quad \gamma(t) \quad \theta(t) \quad \phi(t) \quad \psi(t) \quad x_k(t) \quad y_k(t) \quad 0 \right)^T \quad (2.1)$$

Dans la deuxième phase, la dimension du système sera réduite en maintenant 6 coordonnées généralisées.

$$q_r = \left(\alpha(t) \quad \beta(t) \quad \gamma(t) \quad \theta(t) \quad \phi(t) \quad \psi(t) \right)^T \quad (2.2)$$

La position de la balle peut être calculée à partir de ces 6 coordonnées. Une description détaillée de la procédure du calcul est donnée dans les sections suivantes.

2.4 Hypothèses

Dans cette section, toutes les hypothèses sont listées :

- Zéro glissement entre la sphère et le sol, c'est-à-dire que la vitesse instantanée au point de contact est toujours zéro.

- Il n'y a pas de glissement entre les roues d'actionnement et la balle.

- La sphère est homogène, c'est-à-dire dans les moment d'inertie de la sphère

$$j_s = j_{s,x} = j_{s,y} = j_{s,z}.$$

- La surface sur laquelle la balle roule est plate et horizontale.

- La balle est rigide et ne se déforme donc pas, c'est-à-dire que le rayon de la balle r_s est constant.

- Le corps peut être modélisé comme un pendule qui tourne autour du centre de la balle.

- La distance entre le centre de gravité de la balle et le corps est constante.

2.5 Repères

La Figure (2.2) montrent les différents repères qui seront utilisés. La figure (2.2.a) montre les repère de sphère \mathcal{R}_k et \mathcal{R}_s . Ici, l'origine du repère \mathcal{R}_k coïncide avec le centre de gravité de la balle et n'est translatée que par rapport au référentiel inertiel \mathcal{R}_0 . L'origine du repère \mathcal{R}_s coïncide également avec le centre de gravité de la balle et le repère est lié rigidement au ballon. Ce repère sera également appelé le repère de la sphère.

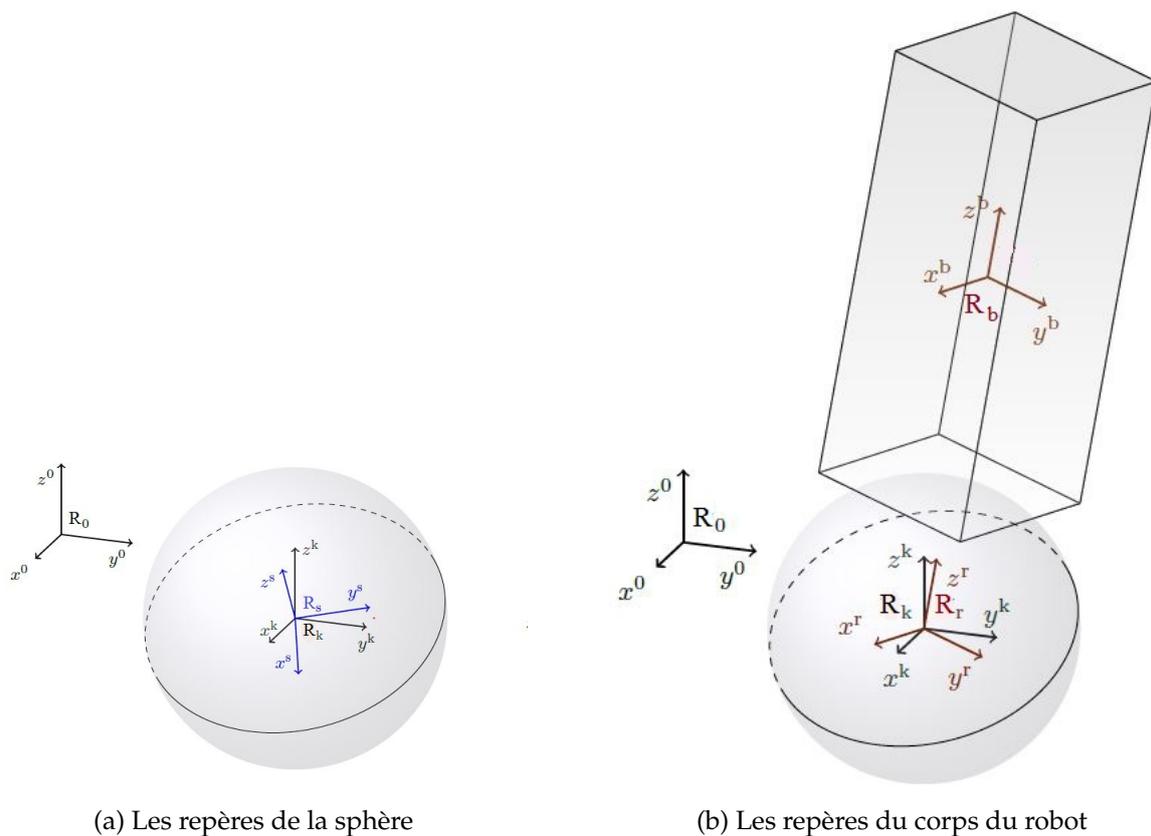


FIGURE 2.2 – Description des repères

La Figure (2.2.b) montre les deux repères relié au corps \mathcal{R}_r et \mathcal{R}_b . Ici, l'origine de \mathcal{R}_r coïncide avec le centre de gravité de la balle et est liée rigidement au corps, c'est-à-dire que le repère a la même rotation que le corps. L'origine de \mathcal{R}_b coïncide avec le centre de masse du corps et est liée rigidement au corps. Ce repère sera également appelé le repère du corps.

2.6 Étude cinématique

Dans cette section, les relations cinématiques du robot sont dérivées. Tout d'abord, les matrices homogènes reliant les différents repères sont définies. Ensuite, les torseurs sont définis et la matrice jacobienne sera construite.

2.6.1 Matrice homogène

La relation entre les repères définis peut être exprimée à l'aide de matrices homogènes. Les matrices homogènes reliant un certain repère au repère inertiel seront utilisées ultérieurement pour trouver les torseurs entre le repère inertiel et un certain repère.

Le repère \mathfrak{R}_k est liée au repère inertiel \mathfrak{R}_0 par une translation pure p_k^0 :

$$T_k^0 = \begin{pmatrix} 1 & 0 & 0 & x_k(t) \\ 0 & 1 & 0 & y_k(t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

Le repère \mathfrak{R}_s est lié au repère \mathfrak{R}_k par une rotation R_s^k et pas de translation.

$$T_s^k = \begin{pmatrix} \cos(\beta(t)) \cos(\gamma(t)) & \sin(\alpha(t)) \sin(\beta(t)) \cos(\gamma(t)) - \sin(\gamma(t)) \cos(\alpha(t)) & \sin(\alpha(t)) \sin(\gamma(t)) + \sin(\beta(t)) \cos(\alpha(t)) \cos(\gamma(t)) & 0 \\ \sin(\gamma(t)) \cos(\beta(t)) & \cos(\alpha(t)) \cos(\gamma(t)) + \sin(\alpha(t)) \sin(\beta(t)) \sin(\gamma(t)) & \sin(\beta(t)) \sin(\gamma(t)) \cos(\alpha(t)) - \sin(\alpha(t)) \cos(\gamma(t)) & 0 \\ -\sin(\beta(t)) & \sin(\alpha(t)) \cos(\beta(t)) & \cos(\alpha(t)) \cos(\beta(t)) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matrice homogène reliant le repère de la sphère \mathfrak{R}_s et le repère inertiel \mathfrak{R}_0 peut être trouvée en utilisant la multiplication des matrices suivantes :

$$\begin{aligned} T_s^0 &= T_k^0 T_s^k \\ &= \begin{pmatrix} I & p_k^0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_s^k & 0 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} R_s^k & p_k^0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

A partir de cette matrice homogène, il est évident que $R_s^0 = R_s^k$. Par conséquent, la matrice homogène reliant le repère de la sphère au repère inertiel est donnée par :

$$T_s^0 = \begin{pmatrix} R_s^0 & p_k^0 \\ 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos(\beta(t)) \cos(\gamma(t)) & \sin(\alpha(t)) \sin(\beta(t)) \cos(\gamma(t)) - \sin(\gamma(t)) \cos(\alpha(t)) & \sin(\alpha(t)) \sin(\gamma(t)) + \sin(\beta(t)) \cos(\alpha(t)) \cos(\gamma(t)) & x_k(t) \\ \sin(\gamma(t)) \cos(\beta(t)) & \cos(\alpha(t)) \cos(\gamma(t)) + \sin(\alpha(t)) \sin(\beta(t)) \sin(\gamma(t)) & \sin(\beta(t)) \sin(\gamma(t)) \cos(\alpha(t)) - \sin(\alpha(t)) \cos(\gamma(t)) & y_k(t) \\ -\sin(\beta(t)) & \sin(\alpha(t)) \cos(\beta(t)) & \cos(\alpha(t)) \cos(\beta(t)) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

De la même manière concernant le repère de la sphère \mathfrak{R}_s , la matrice homogène pour le repère \mathfrak{R}_r est donnée par :

$$T_r^0 = \begin{pmatrix} R_r^0 & p_k^0 \\ 0 & 1 \end{pmatrix} \quad (2.5)$$

Avec $R_r^k = R_r^0$

Enfin, la matrice homogène entre le repère lié au corps \mathfrak{R}_b et le repère inertiel est à nouveau trouvée en utilisant la multiplication des matrices suivantes :

$$\begin{aligned} T_b^0 &= T_r^0 T_b^r \\ &= \begin{pmatrix} R_r^0 & p_k^0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I & p_b^r \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} R_r^0 & R_r^0 * p_b^r + p_k^0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

Avec :

$$p_b^r = [0 \quad 0 \quad l]^T$$

Là encore, il est évident que $R_b^0 = R_r^0$. Par conséquent, la matrice homogène entre le repère du corps et le repère inertiel est donnée par :

$$T_b^0 = \begin{pmatrix} R_b^0 & p_b^0 \\ 0 & 1 \end{pmatrix} \quad (2.6)$$

Avec :

$$R_b^0 = \begin{pmatrix} \cos(\phi(t)) \cos(\psi(t)) & \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t)) - \sin(\psi(t)) \cos(\theta(t)) & \sin(\psi(t)) \sin(\theta(t)) + \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)) \\ \sin(\psi(t)) \cos(\phi(t)) & \cos(\psi(t)) \cos(\theta(t)) + \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)) & \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t)) - \sin(\theta(t)) \cos(\psi(t)) \\ -\sin(\phi(t)) & \sin(\theta(t)) \cos(\phi(t)) & \cos(\phi(t)) \cos(\theta(t)) \end{pmatrix}$$

$$p_b^0 = \begin{pmatrix} x_k(t) + l (\sin(\psi(t)) \sin(\theta(t)) + \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t))) \\ y_k(t) - l (\sin(\theta(t)) \cos(\psi(t)) - \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \\ l \cos(\phi(t)) \cos(\theta(t)) \end{pmatrix}$$

2.6.2 Torseur cinématique

2.6.2.1 Configuration du vecteur des coordonnées généralisées

En utilisant ces transformations homogènes, il est possible de décrire parfaitement le mouvement du système. Comme défini en 2.1, le vecteur contenant les états est donné par q dont sa dérivée :

$$\dot{q} = \left(\frac{\partial}{\partial t} \alpha(t) \quad \frac{\partial}{\partial t} \beta(t) \quad \frac{\partial}{\partial t} \gamma(t) \quad \frac{\partial}{\partial t} \theta(t) \quad \frac{\partial}{\partial t} \phi(t) \quad \frac{\partial}{\partial t} \psi(t) \quad \frac{\partial}{\partial t} x_k(t) \quad \frac{\partial}{\partial t} y_k(t) \quad 0 \right)^T$$

2.6.2.2 Configuration de torseur cinématique

Le torseur cinématique total, de la sphère et du corps, est donné par :

$$\Upsilon^{0,0} = \begin{pmatrix} \Upsilon_s^{0,0} \\ \Upsilon_b^{0,0} \end{pmatrix},$$

Rappelez-vous dans les préliminaires (annexe .2 de la section 2.1.1, la relation entre la forme "tilde" des torseurs $\tilde{\Upsilon}_i^{0,0}$ et les matrices homogènes :

$$\tilde{\Upsilon}_i^{0,0} = \dot{T}_i^0 T_0^i = \begin{pmatrix} \tilde{\omega}_i^{0,0} & v_i^{0,0} \\ 0 & 0 \end{pmatrix}$$

où i désigne un certain repère i .

Pour le repère \mathfrak{R}_k :

$$\tilde{\Upsilon}_k^{0,0} = \dot{T}_k^0 T_0^k = \begin{pmatrix} 0 & \dot{p}_k^0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I & -p_k^0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \tilde{\omega}_k^{0,0} & v_k^{0,0} \\ 0 & 0 \end{pmatrix}$$

Par conséquent, les relations suivantes sont valables :

$$\tilde{\omega}_k^{0,0} = \omega_k^{0,0} = 0, \quad v_k^{0,0} = \begin{pmatrix} \frac{\partial}{\partial t} x_k(t) \\ \frac{\partial}{\partial t} y_k(t) \\ 0 \end{pmatrix} \quad (2.7)$$

Pour le repère \mathfrak{R}_s :

$$\tilde{\Upsilon}_s^{0,0} = \dot{T}_s^0 T_0^s = \begin{pmatrix} \dot{R}_s^0 & \dot{p}_k^0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_0^s & -R_0^s p_k^0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \tilde{\omega}_s^{0,0} & v_s^{0,0} \\ 0 & 0 \end{pmatrix}$$

On obtient :

$$\omega_s^{0,0} = \begin{pmatrix} \cos(\beta(t)) \cos(\gamma(t)) \frac{\partial}{\partial t} \alpha(t) - \sin(\gamma(t)) \frac{\partial}{\partial t} \beta(t) \\ \cos(\gamma(t)) \frac{\partial}{\partial t} \beta(t) + \sin(\gamma(t)) \cos(\beta(t)) \frac{\partial}{\partial t} \alpha(t) \\ \frac{\partial}{\partial t} \gamma(t) - \sin(\beta(t)) \frac{\partial}{\partial t} \alpha(t) \end{pmatrix} \quad (2.8)$$

$$v_s^{0,0} = \begin{pmatrix} \frac{\partial}{\partial t} x_k(t) + y_k(t) \left(\frac{\partial}{\partial t} \gamma(t) - \sin(\beta(t)) \frac{\partial}{\partial t} \alpha(t) \right) \\ \frac{\partial}{\partial t} y_k(t) - x_k(t) \left(\frac{\partial}{\partial t} \gamma(t) - \sin(\beta(t)) \frac{\partial}{\partial t} \alpha(t) \right) \\ x_k(t) \left(\cos(\gamma(t)) \frac{\partial}{\partial t} \beta(t) + \sin(\gamma(t)) \cos(\beta(t)) \frac{\partial}{\partial t} \alpha(t) \right) + y_k(t) \left(\sin(\gamma(t)) \frac{\partial}{\partial t} \beta(t) - \cos(\beta(t)) \cos(\gamma(t)) \frac{\partial}{\partial t} \alpha(t) \right) \end{pmatrix} \quad (2.9)$$

Finalement, le calcul pour \mathfrak{R}_b :

$$\tilde{\Upsilon}_b^{0,0} k = \dot{T}_b^0 T_0^b = \begin{pmatrix} \dot{R}_b^0 & \dot{R}_b^0 p_b^r + p_k^0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} R_0^b & -R_0^b (\dot{R}_b^0 p_b^r + p_k^0) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \tilde{\omega}_b^{0,0} & v_b^{0,0} \\ 0 & 0 \end{pmatrix}$$

Les résultats obtenus :

$$\omega_b^{0,0} = \begin{pmatrix} \cos(\phi(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \theta(t) - \sin(\psi(t)) \frac{\partial}{\partial t} \phi(t) \\ \cos(\psi(t)) \frac{\partial}{\partial t} \phi(t) + \sin(\psi(t)) \cos(\phi(t)) \frac{\partial}{\partial t} \theta(t) \\ \frac{\partial}{\partial t} \psi(t) - \sin(\phi(t)) \frac{\partial}{\partial t} \theta(t) \end{pmatrix} \quad (2.10)$$

$$v_b^{0,0} = \begin{pmatrix} \frac{\partial}{\partial t} x_k(t) + y_k(t) \left(\frac{\partial}{\partial t} \psi(t) - \sin(\phi(t)) \frac{\partial}{\partial t} \theta(t) \right) \\ \frac{\partial}{\partial t} y_k(t) - x_k(t) \left(\frac{\partial}{\partial t} \psi(t) - \sin(\phi(t)) \frac{\partial}{\partial t} \theta(t) \right) \\ x_k(t) \left(\cos(\psi(t)) \frac{\partial}{\partial t} \phi(t) + \sin(\psi(t)) \cos(\phi(t)) \frac{\partial}{\partial t} \theta(t) \right) + y_k(t) \left(\sin(\psi(t)) \frac{\partial}{\partial t} \phi(t) - \cos(\phi(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \theta(t) \right) \end{pmatrix} \quad (2.11)$$

2.6.3 Matrice jacobienne

En utilisant les relations entre les torseurs cinématiques et les vitesses de calculées données par (2.6) à (2.10), la matrice jacobienne peut être construit.

$$\Upsilon^{0,0} = J\dot{q} \quad (2.12)$$

tel que :

$$\Upsilon^{0,0} = \begin{pmatrix} \Upsilon_s^{0,0} \\ \Upsilon_b^{0,0} \end{pmatrix}, \quad \dot{q} = \left(\frac{\partial}{\partial t} \alpha(t) \quad \frac{\partial}{\partial t} \beta(t) \quad \frac{\partial}{\partial t} \gamma(t) \quad \frac{\partial}{\partial t} \theta(t) \quad \frac{\partial}{\partial t} \phi(t) \quad \frac{\partial}{\partial t} \psi(t) \quad \frac{\partial}{\partial t} x_k(t) \quad \frac{\partial}{\partial t} y_k(t) \quad 0 \right)^T$$

2.7 Etude dynamique

Dans cette section, la dynamique du robot est dérivée à l'aide des équations d'Euler-Lagrange. Tout d'abord, l'énergie cinétique et potentielle seront dérivées, ensuite, les équations du mouvement peuvent être calculées. Cette section va inclure la contrainte de roulement dans les équations du mouvement.

2.7.1 Tenseur d'inertie

Avant de pouvoir calculer l'énergie cinétique, il faut définir le tenseur d'inertie. Les repères \mathcal{R}_s et \mathcal{R}_b sont liés rigidement au centre de masse de la balle et du corps du robot respectivement, et correspondent donc aux repères inertiels principaux des corps rigides, c'est-à-dire que le tenseur d'inertie prend la forme :

$$\mathbf{I}^{i,i} = \begin{pmatrix} J_i & 0 \\ 0 & m_i I \end{pmatrix} \quad (2.13)$$

avec :

$$J_i = \begin{pmatrix} j_{i,x} & 0 & 0 \\ 0 & j_{i,y} & 0 \\ 0 & 0 & j_{i,z} \end{pmatrix} \quad (2.14)$$

où $j_{i,j}$ désigne le moment d'inertie de i autour de l'axe j et m_i la masse de i . Les tenseurs d'inertie du repère de la sphère et du repère du corps sont donnés par :

$$\mathbf{I}^{s,s} = \begin{pmatrix} J_s & 0 \\ 0 & m_s I \end{pmatrix} \quad \mathbf{I}^{b,b} = \begin{pmatrix} J_b & 0 \\ 0 & m_b I \end{pmatrix}$$

Pour une balle homogène, on a : $j_s = j_{s,x} = j_{s,y} = j_{s,z}$, $J_s = j_s I$.

Les tenseurs d'inertie sont définis dans les repères liés à chaque partie du robot (sphère et corps). Cependant, la dynamique sera exprimée dans le repère inertiel. Par conséquent, il est nécessaire que les tenseurs d'inertie soient également transformés dans ce repère. La transformation d'un tenseur d'inertie exprimé dans un repère i en repère inertiel est donnée par la relation suivante :

$$\mathbf{I}^{0,i} = (Ad_{T_0^i})^T \mathbf{I}^{i,i} Ad_{T_0^i} \quad (2.15)$$

Avec :

$$Ad_{T_0^i} = \begin{pmatrix} (R_i^0)^T & 0 \\ -(R_i^0)^T \tilde{p}_i^0 & (R_i^0)^T \end{pmatrix} \quad (2.16)$$

2.7.2 Énergie cinétique

L'énergie cinétique du système global est donnée par :

$$E_c(q, \dot{q}) = \frac{1}{2} \begin{pmatrix} \Upsilon_s^{0,0} & \Upsilon_b^{0,0} \end{pmatrix} \begin{pmatrix} \mathbf{I}^{0,s} & 0 \\ 0 & \mathbf{I}^{0,b} \end{pmatrix} \begin{pmatrix} \Upsilon_s^{0,0} \\ \Upsilon_b^{0,0} \end{pmatrix}$$

En utilisant l'équation (2.12), l'énergie cinétique peut être exprimée en utilisant la matrice jacobienne et la dérivée du vecteur des coordonnées \dot{q} :

$$E_c(q, \dot{q}) = \frac{1}{2} \dot{q}^T \underbrace{J^T \begin{pmatrix} \mathbf{I}^{0,s} & 0 \\ 0 & \mathbf{I}^{0,b} \end{pmatrix} J}_{M(q)} \dot{q}$$

Ici, $M(q)$ désigne la matrice d'inertie. L'énergie cinétique résultante est donnée par :

$$E_c = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (2.17)$$

L'équation complète de l'énergie cinétique sera donnée dans l'annexe 3

2.7.3 Énergie potentielle

Maintenant, l'énergie potentielle du robot sera dérivée. Le centre de gravité de la sphère est à une hauteur constante. Par conséquent, l'énergie potentielle de la sphère reste constante et n'aura aucune influence sur la dynamique. De plus, si le repère inertiel est choisi à un niveau avec le repère \mathfrak{R}_k , l'énergie potentielle est nulle et restera nulle. Par conséquent, seule l'énergie potentielle du corps du robot est prise en compte.

Le repère du corps du robot \mathfrak{R}_b est fixé au centre de gravité, le point du centre de gravité exprimé dans ce repère est donc donné par $P^b = (0 \ 0 \ 0 \ 1)^T$. Ce point peut facilement être écrit dans le repère inertiel en utilisant la matrice homogène T_b^0 . L'énergie potentielle peut alors être exprimée en :

$$E_p(q) = m_b g \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} T_b^0 P^b$$

L'expression finale de l'énergie potentielle est donnée par :

$$E_p(q) = g l m_b \cos(\phi(t)) \cos(\theta(t)) \quad (2.18)$$

2.7.4 Équation d'Euler Lagrange

Étant donné l'énergie cinétique et potentielle, c'est possible de calculer le lagrangien

$$L = E_c(q, \dot{q}) - E_p(q)$$

et calculer les équations du mouvement en utilisant les équations d'Euler-Lagrange

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = F_{ext}^T \quad (2.19)$$

les équations d'Euler-Lagrange peuvent être facilement calculées et peuvent être réécrites sous la forme :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = F_{ext}^T \quad (2.20)$$

Où $M(q)$ désigne la matrice d'inertie, $C(q, \dot{q})$ la matrice de Coriolis, $G(q)$ la matrice de gravité et F_{ext} désigne toutes les forces externes.

2.7.5 Contrainte de roulement

Jusqu'à présent, la dynamique du robot sans interaction implicite avec le sol est déduite et donnée dans (2.20). Cependant, en réalité, la sphère est en contact avec le sol et est soumise à un roulement pur, c'est-à-dire à un roulement sans glissement. La partie suivante décrira comment inclure la contrainte de roulement dans les équations du mouvement[19].

La contrainte de roulement pure implique que la vitesse au point de contact est égale à zéro. Considérons un repère supplémentaire \mathfrak{R}_c au point de contact entre la sphère et le sol. Ce repère est lié au repère \mathfrak{R}_k par $p_c^k = r = \begin{pmatrix} 0 & 0 & -r_s \end{pmatrix}^T$ où r_s désigne le rayon de la balle :

$$T_k^c = \begin{pmatrix} I & -r \\ 0 & 1 \end{pmatrix} \quad (2.21)$$

En prenant le repère \mathfrak{R}_c au point de contact, il est connu que la vitesse linéaire de la sphère par rapport au repère inertiel vue depuis le repère \mathfrak{R}_c doit être égale à zéro, c-à-d. $v_s^{c,0} = 0$. Reliant cette vitesse aux torseurs du repère \mathfrak{R}_k résulte en :

$$v_s^{c,0} = -\tilde{r}\omega_s^{k,0} + v_s^{k,0} \quad (2.22)$$

L'étape suivante consiste à exprimer $\omega_s^{k,0}$ et $v_s^{k,0}$. La vitesse angulaire $\omega_s^{k,0}$ peut être exprimée comme suit :

$$\omega_s^{k,0} = R_0^k \omega_s^{0,0} = \omega_s^{0,0} \quad (2.23)$$

La vitesse linéaire $v_s^{k,0}$ est légèrement plus impliquée. Tout d'abord, la relation appliquée pour calculer la vitesse linéaire $v_s^{0,0}$ dans (4-9) est donnée par :

$$v_s^{0,0} = \tilde{p}_k^0 \omega_s^{0,0} + v_k^{0,0}$$

Réécrivons $v_s^{k,0}$ dans le repère inertiel et remplaçons $v_s^{0,0}$

$$\begin{aligned} v_s^{k,0} &= \tilde{p}_0^k \omega_s^{0,0} + v_s^{0,0} \\ &= \tilde{p}_0^k \omega_s^{0,0} + \tilde{p}_k^0 \omega_s^{0,0} + v_k^{0,0} \end{aligned}$$

Donc :

$$v_s^{k,0} = v_k^{0,0} \quad (2.24)$$

La contrainte de roulement exprimée en fonction des vitesses linéaires et angulaire est donnée par :

$$\begin{aligned} -\tilde{r} \omega_s^{0,0} + v_k^{0,0} &= 0 \\ \left(\begin{array}{ccc} -\tilde{r} & 0 & I \end{array} \right) \begin{pmatrix} \omega_s^{0,0} \\ \omega_b^{0,0} \\ v_k^{0,0} \end{pmatrix} &= 0 \end{aligned}$$

On peut écrire les vitesses en fonction des coordonnées généralisées de \dot{q} :

$$\begin{pmatrix} \omega_s^{0,0} \\ \omega_b^{0,0} \\ v_k^{0,0} \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(\beta(t)) \cos(\gamma(t)) & -\sin(\gamma(t)) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sin(\gamma(t)) \cos(\beta(t)) & \cos(\gamma(t)) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sin(\beta(t)) & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\phi(t)) \cos(\psi(t)) & -\sin(\psi(t)) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sin(\psi(t)) \cos(\phi(t)) & \cos(\psi(t)) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\sin(\phi(t)) & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_Q \dot{q}$$

Finalement, la contrainte de roulement est donnée par :

$$A\dot{q} = 0 \quad (2.25)$$

Avec :

$$A = \left(\begin{array}{ccc} -\tilde{r} & 0 & I \end{array} \right) Q = \begin{pmatrix} -r_s \sin(\gamma(t)) \cos(\beta(t)) & -r_s \cos(\gamma(t)) & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ r_s \cos(\beta(t)) \cos(\gamma(t)) & -r_s \sin(\gamma(t)) & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.26)$$

2.7.6 Simplification des équation de mouvement

La contrainte de roulement donnée dans l'équation (4-33) peut être incluse par le biais de multiplicateurs lagrangiens, qui imposent une force virtuelle sur les équations de mouvements :

$$\begin{cases} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = A^T \lambda + F_{ext}^T \\ A\dot{q} = 0 \end{cases} \quad (2.27)$$

Les multiplicateurs lagrangiens peuvent être explicitement résolus. Cependant, pour les matrices relativement grandes du système, cela peut nécessiter des calculs intensifs. Au lieu de cela, l'espace nul de la matrice de contraintes est utilisé pour éliminer les multiplicateurs lagrangiens des équations du mouvement. Le vecteur \dot{q} devrait se situer dans cet espace nul.

Considérons un vecteur des coordonnées généralisées réduit \dot{q}_r , avec :

$$\dot{q} = S\dot{q}_r \quad (2.28)$$

En multipliant par S^T , on obtient les équations de mouvement réduites suivantes :

$$\begin{aligned} S^T M(q)\ddot{q} + S^T C(q, \dot{q})\dot{q} + S^T G(q) &= S^T F_{ext}^T \\ S^T M(q)S\ddot{q}_r + (S^T M(q)\dot{S} + S^T C(q, \dot{q})S)\dot{q}_r + S^T G(q) &= S^T F_{ext}^T \\ \overline{M}\ddot{q}_r + \overline{C}\dot{q}_r + \overline{G} &= S^T F_{ext}^T \end{aligned} \quad (2.29)$$

Avec :

$$\begin{aligned} \overline{M} &= S^T M(q)S \\ \overline{C} &= S^T M(q)\dot{S} + S^T C(q, \dot{q})S \\ \overline{G} &= S^T G(q) \end{aligned}$$

Maintenant, le vecteur réduit contenant les coordonnées généralisées est proposé :

$$\dot{q}_r = \left(\frac{\partial}{\partial t}\alpha(t) \quad \frac{\partial}{\partial t}\beta(t) \quad \frac{\partial}{\partial t}\gamma(t) \quad \frac{\partial}{\partial t}\theta(t) \quad \frac{\partial}{\partial t}\phi(t) \quad \frac{\partial}{\partial t}\psi(t) \right)^T \quad (2.30)$$

Avec :

$$\dot{q} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ r_s \sin(\gamma(t)) \cos(\beta(t)) & r_s \cos(\gamma(t)) & 0 & 0 & 0 & 0 & 0 \\ -r_s \cos(\beta(t)) \cos(\gamma(t)) & r_s \sin(\gamma(t)) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_S \dot{q}_r$$

Les matrices \bar{M} , \bar{C} et \bar{G} seront trouver dans l'annexe .3.

2.8 Forces externes

Dans la section précédente, les équations de mouvement sont dérivées, qui ont été soumises à des forces externes. Dans cette section, ces forces externes sont définies.

2.8.1 Actionnement

Le robot est actionné par un nombre spécifique de roues actionnés par des moteurs, lesquels appliquent un torseur dynamique sur la balle et sur le corps. Pour trouver la relation entre le torseur appliquée $W^{0,i}$ et les forces externes F_{ext} , on utilise la puissance fournie :

$$P = F_{ext} \dot{q} = \begin{pmatrix} W^{0,s} & W^{0,b} \end{pmatrix} \begin{pmatrix} T_s^{0,0} \\ T_b^{0,0} \end{pmatrix} \quad (2.31)$$

Alors que (équation 4.12) :

$$\begin{pmatrix} T_s^{0,0} \\ T_b^{0,0} \end{pmatrix} = J \dot{q}$$

Donc :

$$F_{ext} \dot{q} = \begin{pmatrix} W^{0,s} & W^{0,b} \end{pmatrix} J \dot{q}$$

Il est maintenant possible d'exprimer les forces externes en fonction des torseurs appliqués :

$$F_{ext}^T = J^T \begin{pmatrix} (W^{0,s})^T \\ (W^{0,b})^T \end{pmatrix} \quad (2.32)$$

Dans le reste de cette section, les deux torseurs appliquées à la fois à la sphère et au corps sont dérivées.

2.8.2 Actionnement sur la sphère

D'abord, considérons l'actionnement de la sphère. Deux ensembles de vecteurs sont spécifiés. Le premier ensemble est constitué des vecteurs d'unité de direction des actionneurs \hat{r}_j^r , qui donnent la direction dans laquelle la force appliquée agit. Le deuxième ensemble est constitué des vecteurs d'unité de position \hat{r}_j^r , pointant du centre de la sphère aux points de contact avec les roues. Ici j désigne un actionneur spécifique et en exposant le cadre dans lequel le vecteur est exprimé. Notez qu'il est choisi d'exprimer les vecteurs dans le repère \mathfrak{R}_r , qui est fixé au centre de la sphère et lié rigidement au corps. Comme ce repère est fixe par rapport au corps, ces ensembles de vecteurs sont constants.

Pour la roue j , l'amplitude de la force appliquée au point de contact f_j et l'amplitude du couple d'entrée τ_j sont liées par :

$$f_j = \frac{\tau_j}{r_w} \quad (2.33)$$

Ici, r_w désigne le rayon de la roue. La direction de cette force sur la balle exprimée en repère \mathfrak{R}_r est donnée par le vecteur unitaire \hat{u}_j^r , ainsi la force appliquée à la sphère est donnée par :

$$(f_j^{r,s})^T = f_j \hat{u}_j^r = \frac{1}{r_w} \tau_j \hat{u}_j^r$$

Le couple appliqué à la balle est alors :

$$\begin{aligned} (\tau_j^{r,s})^T &= r_s \hat{r}_j^r \times f_j^{r,s} \\ &= \frac{r_s}{r_w} \tau_j (\hat{r}_j^r \times \hat{u}_j^r) \end{aligned}$$

Maintenant, en utilisant la forme tilde pour exprimer le produit vectoriel et en donnant $W^{r,s} = \begin{pmatrix} \tau^{r,s} & f^{r,s} \end{pmatrix}$, le torseur dynamique appliqué sur la sphère exprimée dans le repère \mathfrak{R}_r en fonction des couples pour n actionneurs est donnée par :

$$(W^{r,s})^T = \begin{pmatrix} \frac{r_s}{r_w} I & 0 \\ 0 & \frac{r_s}{r_w} I \end{pmatrix} \begin{pmatrix} \tilde{\hat{r}}_1^r \hat{u}_1^r & \tilde{\hat{r}}_2^r \hat{u}_2^r & \dots & \tilde{\hat{r}}_n^r \hat{u}_n^r \\ \hat{u}_1^r & \hat{u}_2^r & \dots & \hat{u}_n^r \end{pmatrix} \begin{pmatrix} \tau_1 \\ \tau_2 \\ \cdot \\ \cdot \\ \tau_n \end{pmatrix} \quad (2.34)$$

Dans l'équation (2.34), le torseur dynamique est exprimée dans le repère \mathfrak{R}_r , mais les équations du mouvement de la sphère sont exprimées dans le repère inertiel \mathfrak{R}_0 . Cependant, c'est faisable facilement de l'exprimer dans \mathfrak{R}_0 avec :

$$(W^{0,s})^T = (Ad_{T_0^r})^T (W^{r,s})^T \quad (2.35)$$

Cette matrice adjointe transposée est donnée par :

$$(Ad_{T_0^r})^T = \begin{pmatrix} R_r^0 & -R_r^0 \tilde{p}_0^r \\ 0 & R_r^0 \end{pmatrix}$$

Il conviendra de réécrire cette matrice adjointe en fonction de la matrice de rotation R_b^0 le vecteur de translation p_k^0 . La translation \tilde{p}_0^r peut être réécrite comme :

$$p_0^r = -R_0^r p_k^0$$

En utilisant deux propriétés : $(Rp)^{\sim} = R\tilde{p}R^T$ et $\tilde{a}^T = -\tilde{a}$, on obtient $-R_r^0 \tilde{p}_0^r = \tilde{p}_k^0 T_r^0$. En connaissant que $R_r^0 = R_b^0$, la matrice adjointe devient :

$$(Ad_{T_0^r})^T = \begin{pmatrix} R_b^0 & -\tilde{p}_k^0 R_b^0 \\ 0 & R_b^0 \end{pmatrix}$$

Par conséquent, le torseur dynamique appliquée sur la sphère dans le repère inertiel \mathfrak{R}_0 est donnée par :

$$(W^{0,s})^T = \begin{pmatrix} R_b^0 & -\tilde{p}_k^0 R_b^0 \\ 0 & R_b^0 \end{pmatrix} \begin{pmatrix} \frac{r_s}{r_w} I & 0 \\ 0 & \frac{r_s}{r_w} I \end{pmatrix} \begin{pmatrix} \tilde{r}_1^r \hat{u}_1^r & \tilde{r}_2^r \hat{u}_2^r & \dots & \tilde{r}_n^r \hat{u}_n^r \\ \hat{u}_1^r & \hat{u}_2^r & \dots & \hat{u}_n^r \end{pmatrix} \begin{pmatrix} \tau_1 \\ \tau_2 \\ \cdot \\ \cdot \\ \tau_n \end{pmatrix} \quad (2.36)$$

2.8.3 Configuration des roues omnidirectionnelles

La structure choisie de notre ballbot est actionné par trois moteurs à roues omnidirectionnelles. La Figure (2.3) montre une image schématique de la position des trois roues. Dans le plan xy , l'angle entre les actionneurs est de $\frac{2}{3}\pi$ radians (120 degrés).

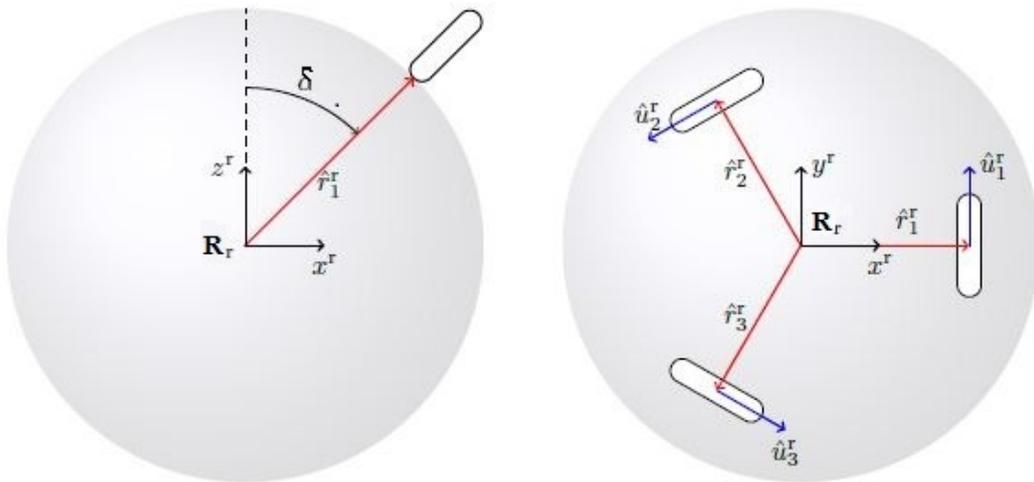
Pour cette configuration, les vecteurs unité de direction de l'actionneur et vecteurs unité de position sont donnés par :

$$\hat{r}_1^r = \begin{pmatrix} \sin(\delta) \\ 0 \\ \cos(\delta) \end{pmatrix}, \quad \hat{r}_2^r = \begin{pmatrix} -\frac{1}{2}\sin(\delta) \\ \frac{1}{2}\sqrt{3}\sin(\delta) \\ \cos(\delta) \end{pmatrix}, \quad \hat{r}_3^r = \begin{pmatrix} -\frac{1}{2}\sin(\delta) \\ -\frac{1}{2}\sqrt{3}\sin(\delta) \\ \cos(\delta) \end{pmatrix}$$

$$\hat{u}_1^r = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \hat{u}_2^r = \begin{pmatrix} -\frac{1}{2}\sqrt{3} \\ -\frac{1}{2} \\ 0 \end{pmatrix}, \quad \hat{u}_3^r = \begin{pmatrix} \frac{1}{2}\sqrt{3} \\ -\frac{1}{2} \\ 0 \end{pmatrix}$$

En utilisant ces vecteurs et ayant $\delta = 45^\circ = \frac{\pi}{4}rad$, le torseur appliquée à la sphère dans le repère \mathfrak{R}_k est donné par :

$$(W^{0,s})^T = \begin{pmatrix} R_b^0 & -\tilde{p}_k^0 R_b^0 \\ 0 & R_b^0 \end{pmatrix} \begin{pmatrix} \frac{r_s}{r_w} I & 0 \\ 0 & \frac{r_s}{r_w} I \end{pmatrix} \begin{pmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ 0 & -\frac{1}{2}\sqrt{3} & \frac{1}{2}\sqrt{3} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} \quad (2.37)$$



.jpg

FIGURE 2.3 – Positionnement des roues par rapport le repère \mathfrak{R}_r

2.8.4 Actionnement sur le corps

Les roues et les moteurs sont considérés comme faisant partie du corps du robot. Comme le dit la troisième loi du mouvement de Newton, les actions de deux corps l'un sur l'autre sont toujours égales et de sens contraires. Cela implique que le torseur dynamique appliquée sur le corps (dans le cadre \mathfrak{R}_r) est opposée au torseur obtenue précédemment et agissant sur la sphère :

$$W^{0,b} = -W^{0,s} \quad (2.38)$$

2.8.5 Forces externes totales

Il est maintenant possible d'exprimer les forces externes en fonction des couples des moteurs (couple), en combinant les équations (2.32) avec (2.37) et (2.38) :

$$\begin{aligned} F_{ext}^T &= J^T \begin{pmatrix} (W^{0,s})^T \\ (W^{0,b})^T \end{pmatrix} \\ &= B\tau^T \end{aligned}$$

De la même manière, on introduit la matrice S pour rendre réduite la matrice B . l'entrée sur la dynamique du système est donnée par :

$$S^T F_{ext}^T = S^T J^T \begin{pmatrix} (W^{0,s})^T \\ (W^{0,b})^T \end{pmatrix} = \bar{B}\tau^T$$

Il est possible de simplifier le calcul. On obtient :

$$\bar{B} = \frac{r_s}{r_w} \begin{pmatrix} I \\ -I \end{pmatrix} R_b^0 \begin{pmatrix} -\frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2} \\ 0 & -\frac{1}{4}\sqrt{6} & \frac{1}{4}\sqrt{6} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} & \frac{1}{4}\sqrt{2} \end{pmatrix} \quad (2.39)$$

La matrice complète de \bar{B} sera trouver dans l'annexe .3.

2.9 Linéarisation

Pour équilibrer le ballbot, les équations de mouvement ne sont nécessaires que autour de la position où il se tient debout. Par conséquent, un modèle linéaire, qui est linéarisé autour de la position verticale du robot, est utilisé pour modéliser son dynamique. Dans la section suivante, il sera vérifié si le modèle linéarisé est une approximation valide du modèle non linéaire. Premièrement, les états et les entrées doivent être définis.

L'état x est défini comme suivant

$$x = \left(\theta(t) \quad \phi(t) \quad \psi(t) \quad x_k(t) \quad y_k(t) \quad \dot{\theta}(t) \quad \dot{\phi}(t) \quad \dot{\psi}(t) \quad \dot{x}_k(t) \quad \dot{y}_k(t) \right)^T \quad (2.40)$$

L'entrée est définie par le vecteur U :

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} \quad (2.41)$$

On remarque que le vecteur d'état x ne contient pas les états en relation avec le roulement de la balle (α, β, γ) . Ça change rien puisque c'est déjà démontré dans la section 2.7.5 qu'il y a une relation entre la translation et le roulement de la sphère. Par conséquent, ils sont remplacés par x_k et y_k afin que nous puissions les utiliser dans le calcul de la commande permettant le maintien de position ainsi que le poursuit d'un trajectoire.

Le système sera linéarisé autour le point d'équilibre instable, où toutes les variables d'état et entrées sont nulles :

$$x_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \quad (2.42)$$

$$u_0 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$$

La linéarisation aboutit à la représentation linéaire de l'espace à états, définie comme suit :

$$\dot{x} = A_l \cdot x + B_l \cdot u \quad (2.43)$$

$$y = C_l \cdot x$$

En appliquant les valeurs de paramètre selon le tableau dans l'annexe .1 à ce modèle d'états, on peut remarquer que les variables d'état x_k, y_k et ψ n'ont pas d'effet sur la dynamique du système (colonnes nuls). Par conséquent, le modèle d'états pourrait en fait être réduit en annulant ces variables d'état. Cependant, étant donné que x_k, y_k, ψ sont des variables de la commande nécessaire (par exemple, pour la commande de la position, de lacet ou de la trajectoire du robot), on va les garder.

2.10 Simulation en boucle ouverte

Dans les paragraphes précédents, un modèle dynamique 3D complet a été dérivé et été linéarisé autour l'origine qui représente le point d'équilibre instable. Dans ce paragraphe, des simulations avec ce modèle 3D seront effectuées pour le valider. De plus, des simulations seront faite sur le modèle linéairisé pour étudier sa précision par rapport le modèle NL.

Tout d'abord, la réponse libre (c'est-à-dire des conditions initiales non nulles et aucun couple externe) du modèle non linéaire pour le plan yz est simulée pendant 20 secondes avec un angle de tangage initial de 0.01 rad. Les résultats sont montrés dans la figure (2.4).

La figure (2.4) montre que le robot va tomber et va commencer à osciller autour du centre de la balle.

Comme le sol n'est pas défini, le robot passera par le plan $z = 0$ et les amplitudes des oscillations ne diminueront pas car aucune perte d'énergie ni aucun frottement dynamique ne sont modélisés. Alors que les oscillations répétitives, on note que la balle se déplace dans la direction de l'axe y dans les deux sens, également de manière répétitive. D'autre part, cette condition initiale n'influence pas les autres états $\phi = \psi = x_k = 0$.

En analysant la figure (2.5), il est clair que la réponse libre (en appliquant la même condition initiale 0,01 rad) du modèle L et NL sont presque identique au voisinage du zéro. Si le modèle NL atteint un angle de tangage θ de $0.5 \text{ rad} \simeq 28 \text{ deg}$, il ya une erreur de 2.59% pour la réponse du modèle L. Concernant, l'évolution de la valeur de position y_k , lorsque la réponse NL égale à 50 cm, on constate une erreur de 0.8% dans le modèle L.

Ainsi, à partir des résultats présentés dans les figure (2.4) et (2.5), premièrement, le modèle 3D non linéaire est valide. Deuxièmement, on peut en conclure que le modèle linéaire est fiable en ce qui concerne l'orientation du corps du ballbot lorsque l'angle d'inclinaison est au plus égal à 0.5 rad. Le modèle linéaire semble également fiable en ce qui concerne la position de la balle.

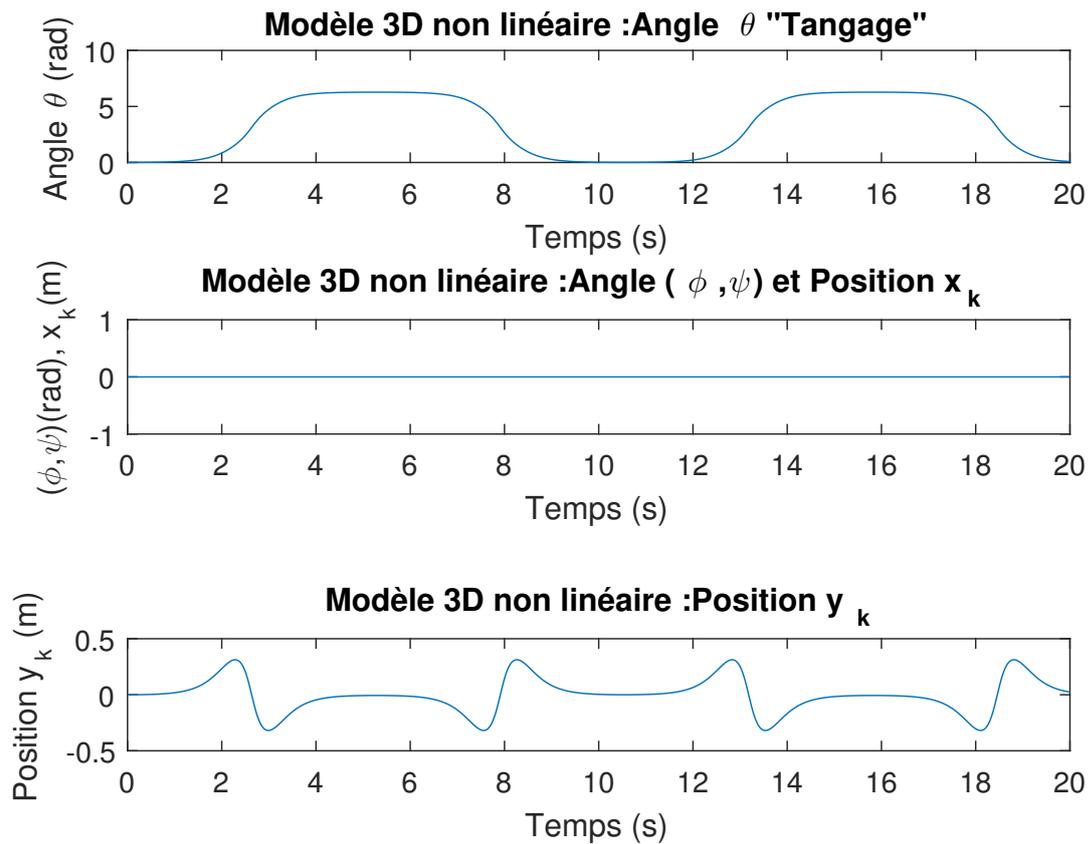


FIGURE 2.4 – Réponse libre du modèle 3D non linéaire pour un angle initiale de tangage de 0.01 rad

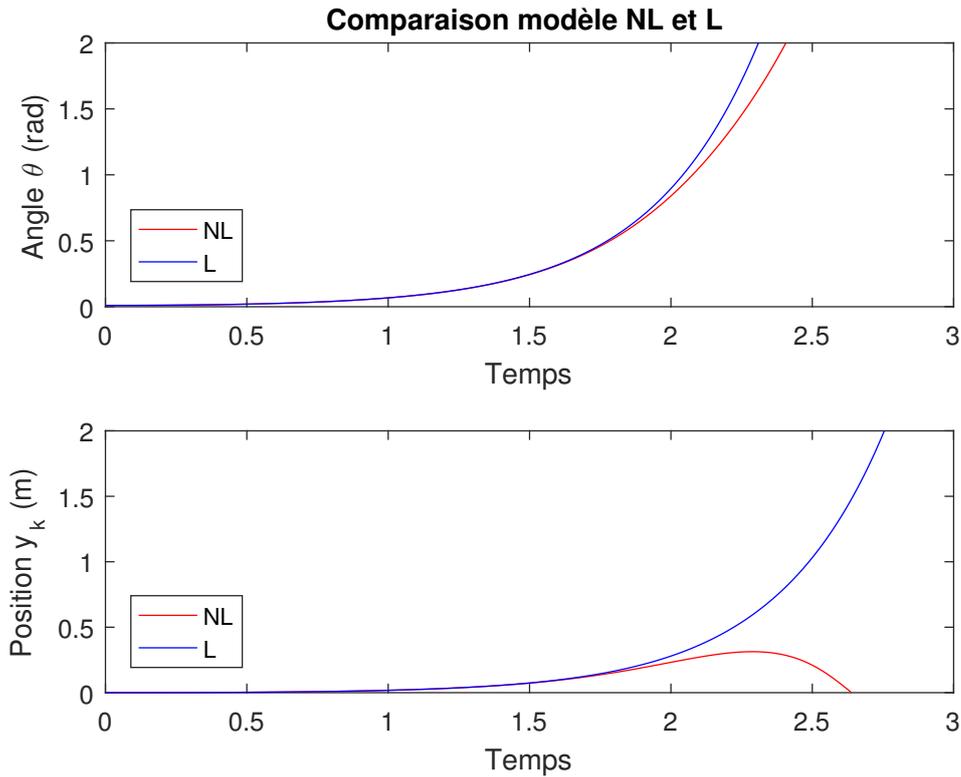


FIGURE 2.5 – Comparaison entre les modèle NL et L

2.11 Conclusion

Dans cette section, les équations finales du mouvement sont brièvement discutées. Tout d'abord, il a été déterminé que coordonnées généralisées était données par le vecteur q ayant 9 états qui décrivent le système complet. L'espace de configuration de la dynamique a été réduit en éliminant 3 coordonnées, donc il consiste en 6 degrés de liberté. Il n'y a que 3 actionneurs et le système est désormais sous-actionné.

Cependant, en considérant le système comme deux sous-systèmes, c'est-à-dire le corps et la sphère, on peut considérer que la sphère ou le corps est complètement actionné. Ceci sera utilisé pour le l'étude de la commande.

Il est toujours possible de choisir des repères différentes de celles définies dans ce chapitre. L'équation du mouvement peut être liée à un autre choix du repère en utilisant les matrices homogènes entre les repères choisies, les nouvelles repères et les matrices adjointes correspondantes. Les équations de mouvement sont exprimées dans le repère inertiel. Cependant, il est possible de les exprimer par rapport à n'importe quel autre repère en utilisant la matrice adjointe et sa dérivée.

Chapitre 3

Proposition de lois de commande

3.1 Introduction

Ce chapitre est consacré à la synthèse de lois de commande. On va développer deux lois de commande LQR et commande par mode glissant pour deux objectifs principaux : le premier c'est la stabilisation du robot balanceur à l'origine. Le deuxième objectif est le maintien de la position initiale et la poursuite d'une séquence des points.

3.2 Proposition de lois de commande et simulation

Dans cette section, deux types de commande seront discutés. Dans un premier temps, nous étudierons la capacité des régulateurs linéaires à obtenir des résultats intéressants lorsqu'ils sont appliqués à ce type de robots (ballbot) nécessitant des performances élevées et une grande robustesse pour rester en position verticale. Dans la deuxième partie, un régulateur non linéaire sera proposé et discuté. Notre objectif est de pouvoir comparer les deux méthodes choisies et analyser leurs avantages et inconvénients.

Dans l'énoncé du problème de la section 1.2, les objectifs de commande désirés ont été énoncés. Dans cette section, ces objectifs de commande seront spécifiés plus en détail. Notez qu'il n'est pas nécessaire qu'un régulateur spécifique remplisse tous les objectifs de commande ci-dessus. Il est donc fort possible qu'il existe plusieurs régulateurs, chacun pour un objectif de commande spécifique.

Balancement du robot

Notre premier objectif est d'établir une loi de commande qui maintient le robot en position verticale sans prendre en considération la position désirée du système dans le plan xy . Cette partie est considérée la partie la plus importante puisque toutes les travaux à venir en dépendent.

Maintien de la position horizontale

La fonctionnalité du régulateur de position consiste à commander le robot dans une certaine position tout en maintenant la stabilité dynamique. Ce régulateur peut être utilisé pour le maintien de poste, c'est-à-dire le maintien de l'équilibre tout en maintenant une position fixe, ou la commande point à point, c'est-à-dire la commande du robot dans une certaine position.

Tout d'abord, l'étude de la commandabilité est une nécessité pour pouvoir appliquer nos régulateurs sur le modèle, donc la prochaine section sera dédiée à cet objectif.

3.2.1 Commandabilité

On dit qu'un système est commandable à l'instant t_0 s'il est possible de conduire le système à partir de tout état initial $x(t_0)$ vers tout autre état dans un intervalle de temps fini au moyen d'un vecteur de commande u [20].

Selon le théorème de Kalman, étant donné un système donné par :

$$\dot{x} = Ax + Bu$$

où $A \in R^{n \times n}$ et $B \in R^{n \times m}$. Ce système est considéré commandable si et seulement si la matrice de commandabilité C est de rang n (l'ordre de système) :

$$C = \begin{pmatrix} B & AB & \dots & A^{n-1}B \end{pmatrix} \quad (3.1)$$

En utilisant Matlab, c'est possible de calculer la matrice de commandabilité C ainsi que le rang. On trouve que $\text{rang}(C)=10=n$, donc le modèle L est complètement commandable et par conséquent le modèle NL est localement commandable.

3.2.2 Observabilité

On dit qu'un système est observable à l'instant t_0 si, avec le système dans l'état $x(t_0)$, il est possible de déterminer cet état à partir de l'observation de la sortie sur un intervalle de temps fini [20].

$$O = \begin{pmatrix} C & CA & \dots & CA^{n-1} \end{pmatrix}^T \quad (3.2)$$

L'observabilité est garantie si le rang de la matrice d'observabilité O est égal à n .

Également pour la matrice d'observabilité, matlab nous permet de la calculer et on trouve que le modèle L est complètement observable ($\text{rang}(O)=10=n$). Ça implique l'observabilité locale du modèle NL.

3.3 Commande linéaire LQR

Les simulations en Sec. 2.10 ont montré que le modèle linéarisé est une approximation fiable du modèle non linéaire en ce qui concerne les angles d'orientation du corps, à condition que l'angle d'inclinaison soit d'au plus 0.5 rad (30 deg), ainsi que la position de la balle ne dépasse pas 0.6 m, ce qui est largement suffisant. De plus, la revue de travaux antérieurs dans le chapitre 1 a montré que pour de nombreux ballbot,

les angles d'orientation du corps et la position de la balle pouvaient être commandés à l'aide de régulateurs linéaires. Sur la base de ces arguments, il est décidé de concevoir un régulateur linéaire pour commander à la fois les angles d'orientation du système ainsi que la position de la balle.

La théorie de commande LQR est une commande qui conçoit un régulateur linéaire optimal en minimisant une fonction quadratique soumise à la dynamique du système. Il présente l'avantage de pouvoir être appliqué aux systèmes MIMO. Un inconvénient de cette théorie LQR est que les paramètres de LQR ne sont souvent pas directement liés aux exigences de commande, ce qui rend plus difficile la recherche du régulateur optimal qui répond à toutes ces exigences.

3.3.1 Conception du régulateur

Dans cette section, un régulateur LQR sera conçu pour commander l'orientation du corps et la position du ballbot simultanément.

Le régulateur LQR consiste en une matrice de gain K , qui sera implémentée sous la forme $u = -Ke$ dans la représentation d'espace d'état du système, où e est l'erreur définie comme $e = x_{ref} - x$, avec x_{ref} désignant les états de référence. Il en résulte un système en boucle fermée avec les états de référence en entrée, illustrée à la Fig. 3.1, pour laquelle la représentation d'états est donnée par :

$$\begin{aligned} \dot{x} &= (A - BK)x + BKx_{ref} \\ y &= Cx \end{aligned} \quad (3.3)$$

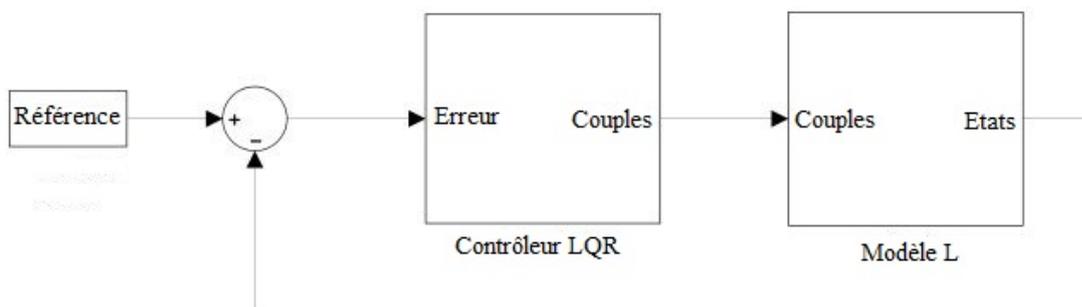


FIGURE 3.1 – Système en boucle fermée avec un régulateur LQR

La matrice de gain K est calculée de sorte que $u = -Ke$ minimise la fonction :

$$J(u) = \int_0^{\infty} (e^T Q e + u^T \rho R u) dt \quad (3.4)$$

3.3.2 Choix de Q et R

Il n'existe pas de méthode standard pour trouver Q , R et ρ . La règle de Bryson [21] est utilisée pour une valeur initiale des matrices Q et R . Avec la règle de Bryson, Q et R sont choisis diagonale avec :

$$\begin{aligned} Q_{ii} &= \frac{1}{e_{i,max}^2} \\ R_{jj} &= \frac{1}{u_{j,max}^2} \end{aligned} \quad (3.5)$$

Avec $e_{i,max}$ est la valeur absolue maximum acceptable de l'erreur entre les états et les références. $u_{j,max}$ est la valeur absolue maximum acceptable en entrée. Dans notre cas, on pose l'erreur de translation 0.5m, l'erreur d'inclinaison égale à 0.15 rad et le couple maximum en entrée égale à 1.59 Nm. En fait, roulis et tangage sont les seuls états dans lesquels leur erreur doit être limitée car ils affectent directement la dynamique du système. D'autre part, le lacet n'a pas d'influence, donc, son erreur peut donc être choisie pour être l'erreur maximale possible π . x_k et y_k n'affectent pas le système, mais il est nécessaire de limiter leur erreur en raison de leur importance dans le maintien de position. Les erreurs maximales pour les dérivées des coordonnées minimales sont prises trois fois plus que les erreurs maximales des coordonnées minimales.

Le calcul sera fait en utilisant Matlab, on peut résoudre l'équation de Riccati et calculer le gain K par différentes valeurs de Q et R les matrices de pondération. On essaye d'optimiser en changeant le facteur ρ . Les matrices de pondération Q et R choisies pour initialiser la simulation sont celles basées sur la règle de Bryson, puis nous changeons ρ et observons comment cela peut influencer la dynamique. Les valeurs choisies de ρ sont : 0.01, 0.1, 1, 10, 100. Nous supposons que toutes les conditions initiales égale à zéro sauf la valeur de l'angle de tangage $\theta = 0.15$ rad. Le résultat obtenu est le suivant :

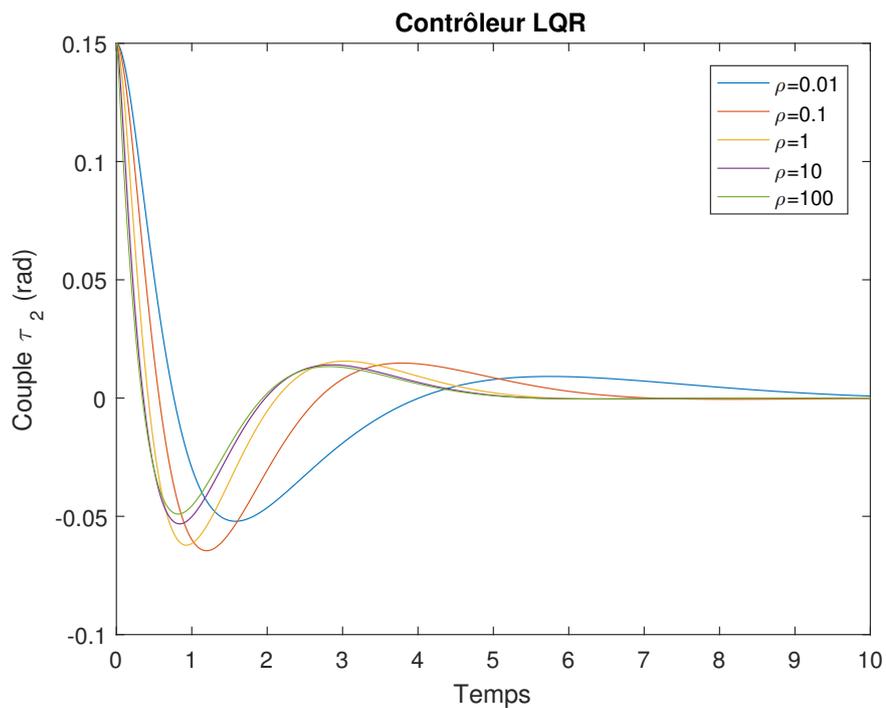


FIGURE 3.2 – Simulation en boucle fermée de la réponse du système à une condition initiale de $\theta = 0.15$ rad avec des valeurs différentes de ρ

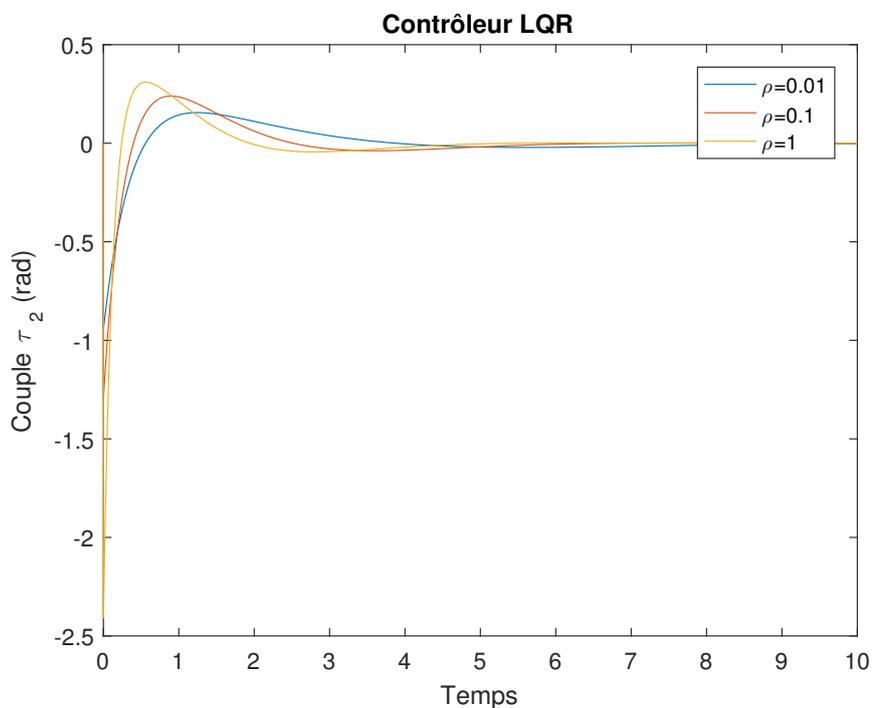


FIGURE 3.3 – Simulation en boucle fermée du couple du système τ_2 à une condition initiale de $\theta = 0.15$ rad avec des valeurs différentes de ρ

Comme le montrent les résultats, les plus petites valeurs de ρ correspondent à un coût bas pour la commande et à une réponse plus lente, tandis que les plus grandes valeurs de ρ correspondent à un coût plus élevé pour la commande et à une réponse relativement rapide. En fait, le choix de ρ dépend fortement de la capacité du micro-contrôleur à utiliser et du couple max des moteurs actionnant le système. Dans notre cas, on pose pour l'instant $\rho = 0.1$ qui permet au système de converger en 7 secondes vers zéro dans le cas de la condition initiale insérée avec un couple absolue maximal atteint égal presque à 1.25 Nm.

La matrice des gain K correspondante à cette valeur de ρ est donnée par :

$$K = \begin{pmatrix} 17.2744 & -0.0000 & -0.0924 & -0.0000 & -0.8211 & 7.2031 & -0.0000 & -0.0818 & -0.0000 & -1.7385 \\ -8.6372 & 15.4599 & -0.0924 & 0.7111 & 0.4105 & -3.6015 & 6.4560 & -0.0818 & 1.5914 & 0.8693 \\ -8.6372 & -15.4599 & -0.0924 & -0.7111 & 0.4105 & -3.6015 & -6.4560 & -0.0818 & -1.5914 & 0.8693 \end{pmatrix} \quad (3.6)$$

3.3.3 Test et simulation

Après avoir établi les matrices de pondération R et Q ainsi que la valeur de ρ , nous allons maintenant lancer des simulations pour observer la robustesse de ce régulateur. La simulation effectuée dans la section précédente a été appliquée sur le modèle linéaire, nous allons donc maintenant implémenter le régulateur avec le modèle non linéaire et étudier les différents cas. La figure 3.4 montre les différentes parties de notre système sur Simulink

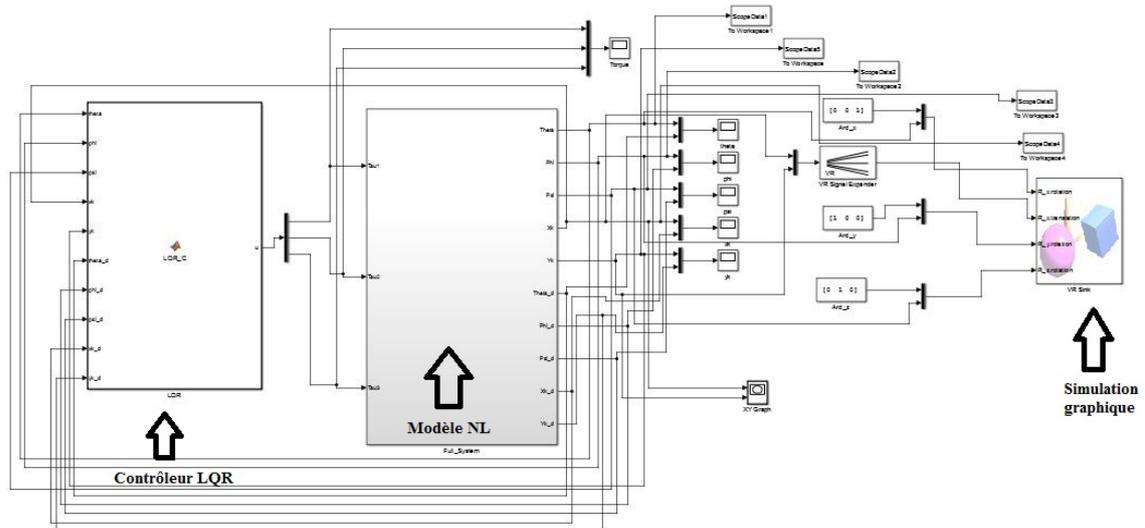


FIGURE 3.4 – Présentation du système complet sur Simulink

3.3.3.1 Balancement du robot

Le premier test à effectuer vérifie la capacité du système à s'équilibrer et maintenir la position verticale en présence de perturbation. Nous exécutons la simulation avec une condition initiale de l'angle de tangage θ égale à 0,2 rad, puis après 4 secondes, nous appliquons une perturbation constante de 0,2 rad suivie d'une autre perturbation constante égale à -0,3 rad. Les résultats sont présentés à la fig 3.5.

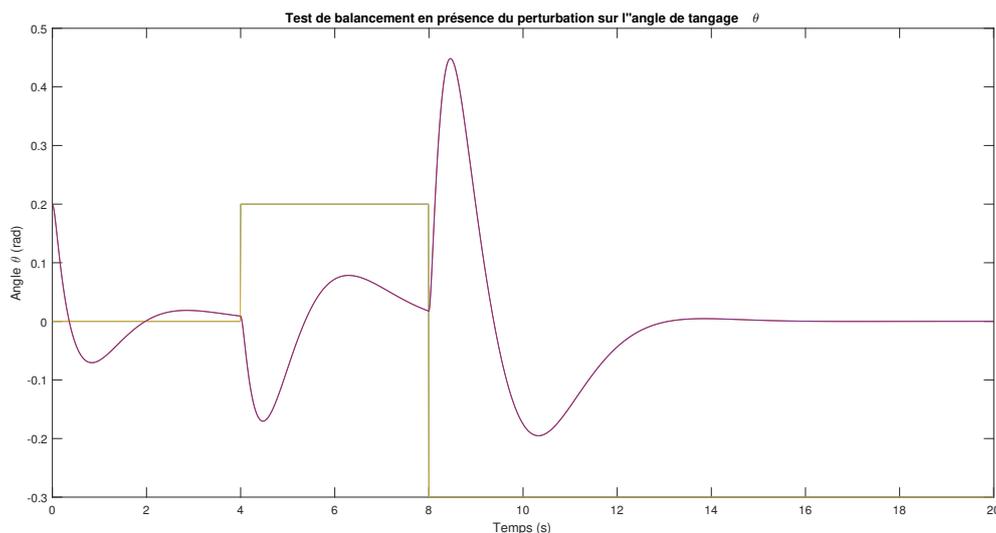


FIGURE 3.5 – Test de balancement avec la commande LQR

Tout d'abord, le régulateur linéaire balance le robot et le maintient en position verticale en présence de différentes perturbations. On peut observer que le système se stabilise après 6s de la dernière perturbation ($\theta_{prt} = -0.3$ rad). En plus, il ya un dépassement de l'origine durant l'évolution de la dynamique du système mais le système revient à sa première position après certain temps . Le robot se déplace toujours dans la direction opposée à la perturbation afin de compenser son influence.

3.3.3.2 Maintien de position

Dans cette partie, les tests étudieront si le système est capable de conserver sa position initiale même en cas de perturbation. Nous appliquons les conditions initiales suivantes : $\theta_0 = 0, 2$ rad , $\phi = 0, 3$ rad, $\psi = 0, 5$ rad. Nous obtenons ces résultats

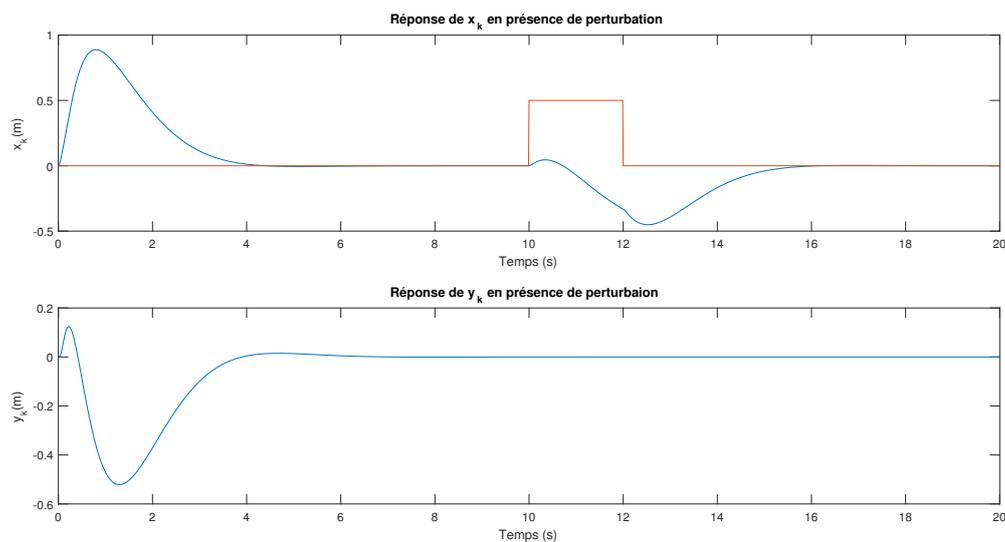


FIGURE 3.6 – Test de positionnement à l'endroit initial

Nous remarquons que le système rejette les perturbations initiales et revient à sa première position dans une durée de 4s. Nous appliquons une perturbation additive sur l'état x_k à $t=10$ s pendant 2s. De même, l'état retourne dans une durée de (4 s) à l'emplacement initial.

3.3.3.3 Commande point à point

Cette sous-section consiste à analyser la capacité des états de translation d'atteindre les points désirés dans le plan xy . Le test proposé pour vérifier ceci est un enchaînement de quatre points de références constituant un chemin d'une forme carré. Les deux signaux à appliquer sont montrés dans la fig 3.7 comme référence avec les réponses des états de x_k et y_k . Les conditions initiales sont nulles. les résultats seront affichés dans la fig 3.8 :

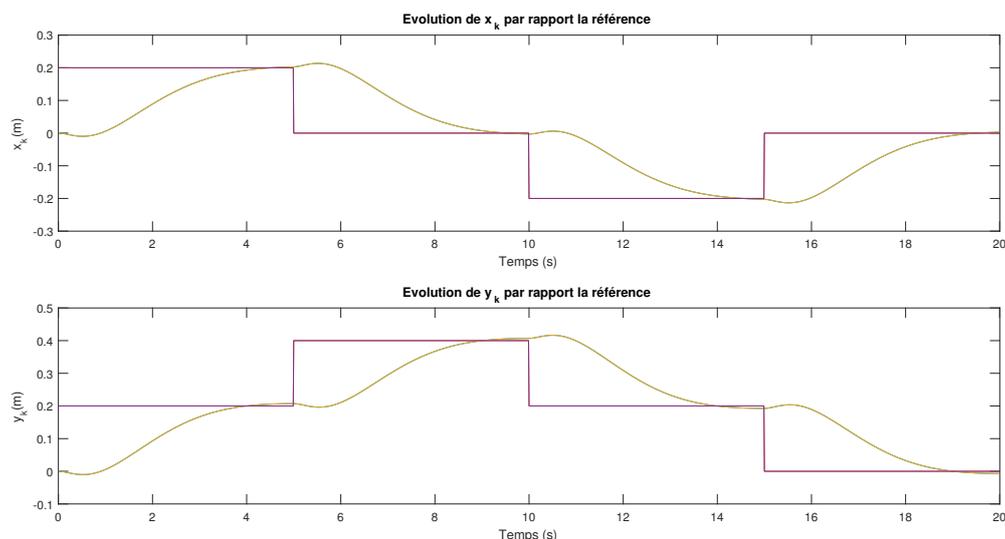


FIGURE 3.7 – L'évolution de x_k et y_k par rapport une référence

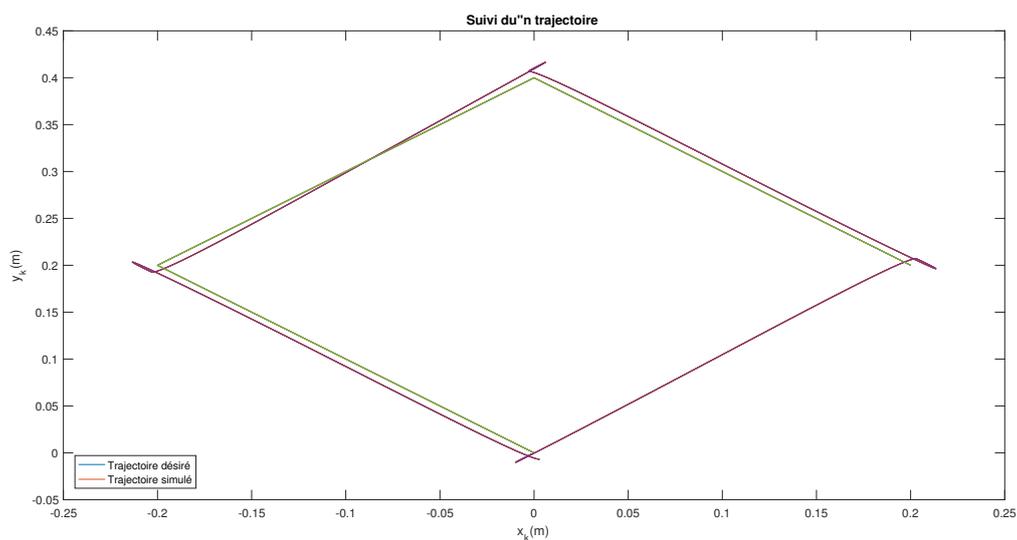


FIGURE 3.8 – Suivi d'une séquence des points

Nous remarquons que ce régulateur linéaire permet au système de suivre les points de référence désirés. Les états x_k et y_k suivent le changement des signaux de référence avec des réponses en 5s et le système reste stable. Nous avons essayé d'augmenter les valeurs de référence mais la simulation indique qu'il ya une possibilité d'existence des singularités (les états divergent brusquement) lorsque la référence dépasse 0,6 m, ce qui est logique puisque le régulateur est basé sur une méthode linéaire LQR qui fonctionne au voisinage de 0.

3.4 Commande non linéaire : mode glissant

Dans cette section, une approche non linéaire du commande du ballbot est proposée. Comme indiqué au chapitre 1, une approche linéaire a souvent été utilisée pour commander le robot. Cependant, les régulateurs linéaires sont synthétisés pour le modèle linéarisé, dans lequel la dynamique non linéaire des systèmes est ignorée. De ce fait, le régulateur n'est valable qu'au voisinage du point de linéarisation où la dynamique est approximativement linéaire, tandis que dans d'autres régions, la stabilité n'est pas assurée. Une méthode de commande non linéaire a également été proposée, c'est la commande par mode glissant. Notez que la synthèse de la loi de commande ne se fera que pour commander les angle d'inclinaison dans cette partie et non pas la commande de la position.

3.4.1 Le partitionnement du système

Nous notons que notre système est sous-actionné avec six états à commander et seulement trois entrées. C'est la raison pour laquelle nous devons le diviser en deux sous-systèmes actionnés. Pour cela, la dynamique du système est d'abord réécrite sous une forme différente. Considérons la forme suivante des équations de mouvement du système :

$$\ddot{q}_r = -\bar{M}^{-1}(\bar{C}\dot{q}_r + \bar{G}) + \bar{M}^{-1}\bar{B}\tau^T$$

Avec

$$\ddot{q}_r = \left(\frac{\partial}{\partial t}\alpha(t) \quad \frac{\partial}{\partial t}\beta(t) \quad \frac{\partial}{\partial t}\gamma(t) \quad \frac{\partial}{\partial t}\theta(t) \quad \frac{\partial}{\partial t}\phi(t) \quad \frac{\partial}{\partial t}\psi(t) \right)^T$$

Laissez-nous vous présenter le partitionnement suivant :

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = -\bar{M}^{-1}(\bar{C}\dot{q}_r + \bar{G})$$

$$\begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = \bar{M}^{-1}\bar{B}$$

et donc :

$$\ddot{q}_r = \begin{pmatrix} \ddot{q}_s \\ \ddot{q}_c \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} + \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \tau^T \quad (3.7)$$

tel que, \ddot{q}_s et \ddot{q}_c sont les accélérations angulaires de la balle et du corps respectivement, et par suite, l'accélération angulaire du corps du robot est donnée par :

$$\ddot{q}_c = f_2 + g_2 \tau^T \quad (3.8)$$

Comme il ressort de calculs antérieurs, la matrice d'inertie a une dimension considérable et son inverse peut difficilement être calculé avec des calculatrices ordinaires. Nous allons donc le décomposer en blocs et calculer chaque inverse. Pour cela, nous appliquons cette formule :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} + A^{-1}BS_p^{-1}CA^{-1} & -A^{-1}BS_p^{-1} \\ -S_p^{-1}CA^{-1} & S_p^{-1} \end{pmatrix}$$

$$\text{Avec : } S_p = D - CA^{-1}B$$

3.4.2 Conception du régulateur de balancement

La commande par mode glissant a connu un essor considérable durant les dernières décennies. Ceci est dû principalement à la propriété de convergence rapide et en temps fini des erreurs, ainsi, que la grande robustesse par rapport aux erreurs de modélisation et certains types de perturbations extérieures [22].

La commande par mode glissant est une commande à structure variable pouvant changer de structure et commutant entre deux valeurs suivant une logique de commutation bien spécifique $s(x)$. Le principe de la commande par mode glissant est de contraindre le système à atteindre une surface donnée appelée surface de glissement et d'y demeurer jusqu'à l'équilibre. Cette commande se fait en deux étapes : la convergence vers la surface et ensuite le glissement le long de celle-ci figure 4.9[22, 23].

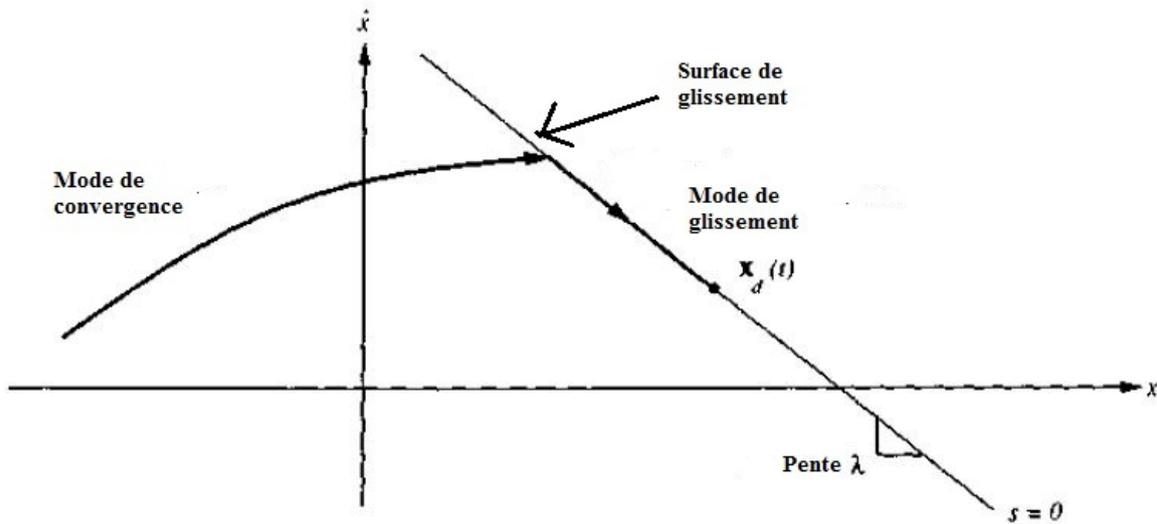


FIGURE 3.9 – Surface de glissement

3.4.2.1 Synthèse de la loi de commande

La synthèse de la commande par mode glissant se fait en trois étapes :

- choix de la surface de glissement
- Etablir la condition de convergence
- déterminer la loi de commande qui permet d'atteindre la surface et d'y demeurer.

3.4.2.2 Choix de la surface de glissement

Le système est considéré comme du type MIMO. Il y a trois entrées pour appliquer la commande (couples) et trois états à équilibrer (converger vers l'origine) qui sont les angles d'inclinaison θ , ϕ and ψ .

Le sous-système est donné dans la section précédente par l'équation 3.7 et peut être mis sous la forme :

$$\dot{x} = f(x, t) + g(x, t)u \quad (3.9)$$

Avec $x = (x_1 \quad \dots \quad x_n)^T$ les états de système, f est un vecteur de n dimensions, g est une matrice de dimension de $(n \times p)$ et u est le vecteur de commande de p dimensions

Dans notre cas : $n = m = 3$ et $p = 3$.

Soit x_{ref} le référence et $e(x)$ l'erreur de poursuite définie par : $e = x - x_{ref}$. Puisque

le régulateur consiste seulement à balancer le système, x_{ref} est considéré nul. On considère la surface de glissement [24] :

$$\begin{aligned} s &= \begin{pmatrix} s_1 & \dots & s_p \end{pmatrix}^T \\ &= \begin{pmatrix} s_1 & s_2 & s_3 \end{pmatrix}^T \end{aligned} \quad (3.10)$$

Une surface de Slotine sera donnée à chaque s_i [23].

$$s_i = \left(\frac{\partial}{\partial t} + \lambda_i \right)^{r_i-1} e(x) \quad (3.11)$$

Avec r est le degré relatif de l'erreur e_i et λ_i est une constante positive. En appliquant cette définition, on aura les surfaces suivantes :

$$\begin{aligned} s_1 &= \lambda_1 \theta + \dot{\theta} \\ s_2 &= \lambda_2 \phi + \dot{\phi} \\ s_3 &= \lambda_3 \psi + \dot{\psi} \end{aligned} \quad (3.12)$$

3.4.2.3 Condition d'existence du mode de glissement

Le choix de la fonction de glissement étant fait, la deuxième étape consiste à concevoir une loi de commande qui puisse amener le vecteur d'état à converger vers la surface et y demeurer ($s=0$). Pour cela, il faut que la loi de commande soit conçue de telle manière à ce que s soit attractive.

Une condition nécessaire et suffisante, appelée condition d'attractivité, pour qu'une variable de glissement $s(x, t)$ tende vers zéro est que la dérivée temporelle de S soit définie négative :

$$s\dot{s} < 0 \quad (3.13)$$

3.4.2.4 Calcul de la commande

En dérivant les surfaces, on trouve :

$$\begin{aligned} \dot{s}_1 &= \lambda_1 \dot{\theta} + \ddot{\theta} \\ \dot{s}_2 &= \lambda_2 \dot{\phi} + \ddot{\phi} \\ \dot{s}_3 &= \lambda_3 \dot{\psi} + \ddot{\psi} \end{aligned} \quad (3.14)$$

Les expressions de deuxième dérivée des états sont données à partir de l'équation 3.7.

La dérivée de la surface de glissement sera écrite sous la forme suivante :

$$\dot{s}_i = h_i + b_{i1}u_1 + \dots + b_{ip}u_p = h_i + \sum_{k=1}^p b_{ik}u_k \quad (3.15)$$

Ensuite, nous pouvons écrire la dérivée du vecteur de surface comme :

$$\dot{s} = h + bu \tag{3.16}$$

Avec :

$$h = \begin{pmatrix} h_1 \\ \cdot \\ \cdot \\ h_p \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} b_{11} & \cdot & \cdot & b_{1p} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ b_{p1} & \cdot & \cdot & b_{pp} \end{pmatrix}$$

Comme mentionné ci-dessus, $p = 3$ alors h est un vecteur de 3 éléments et b est une matrice de dimension 3×3 .

les résultats finaux du calcul de h et b sont donnés par :

$$h = \begin{pmatrix} \lambda_1 \dot{\theta} + f_2(1) \\ \lambda_2 \dot{\phi} + f_2(2) \\ \lambda_3 \dot{\psi} + f_2(3) \end{pmatrix}, \quad b = g_2 \tag{3.17}$$

La loi de commande pour le mode glissant du premier ordre (FOSMC) du système 3.8, de sorte que les surfaces de glissement deviennent nulles en un temps fini est définie par :

$$u = -b^{-1} \left(h + \begin{pmatrix} k_1 \text{sign}(s_1) \\ k_2 \text{sign}(s_2) \\ k_3 \text{sign}(s_3) \end{pmatrix} \right) \tag{3.18}$$

Avec : k_i est une constante positive et b est une matrice inversible

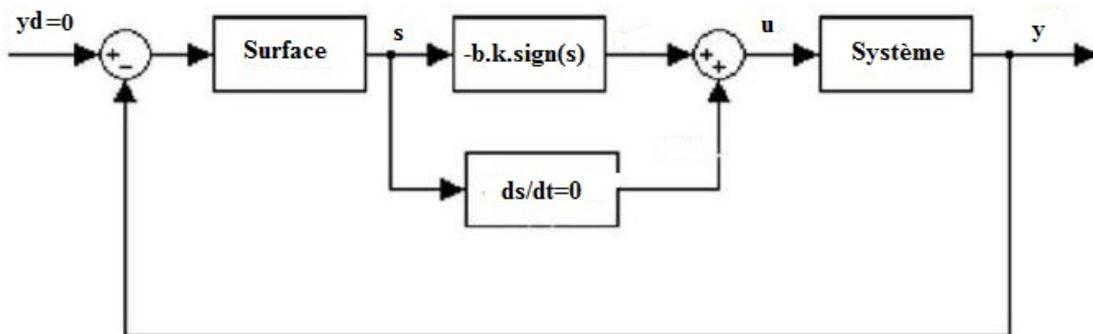


FIGURE 3.10 – Structure de la commande par mode glissant

Selon [24], si $V = \frac{1}{s^T s}$, il est prouvé que $\dot{V} < 0$, alors, la fonction de Lyapunov V tend vers 0, ce qui implique la convergence de toutes les surfaces s_i vers zéro, d'où

l'existence du mode glissant du premier ordre. Il est prouvé également la convergence en temps fini.

3.4.2.5 Le broutement (chattering)

Un régime glissant idéal requiert une commande pouvant commuter à une fréquence infinie. Ainsi, durant le régime glissant, les discontinuités appliquées à la commande peuvent entraîner un phénomène de broutement, appelé "chattering" en anglais. Celui-ci se caractérise par de fortes oscillations des trajectoires du système autour de la surface de glissement figure (3.9). Les principales raisons à l'origine de ce phénomène sont les limitations des actionneurs ou les retards de commutation au niveau de la commande. Ces commutations détériorent la précision de la commande et peuvent s'avérer néfastes pour l'organe de commande en provoquant une détérioration prématurée des systèmes mécaniques et une élévation de température dans les systèmes électriques (perte d'énergie non négligeable).

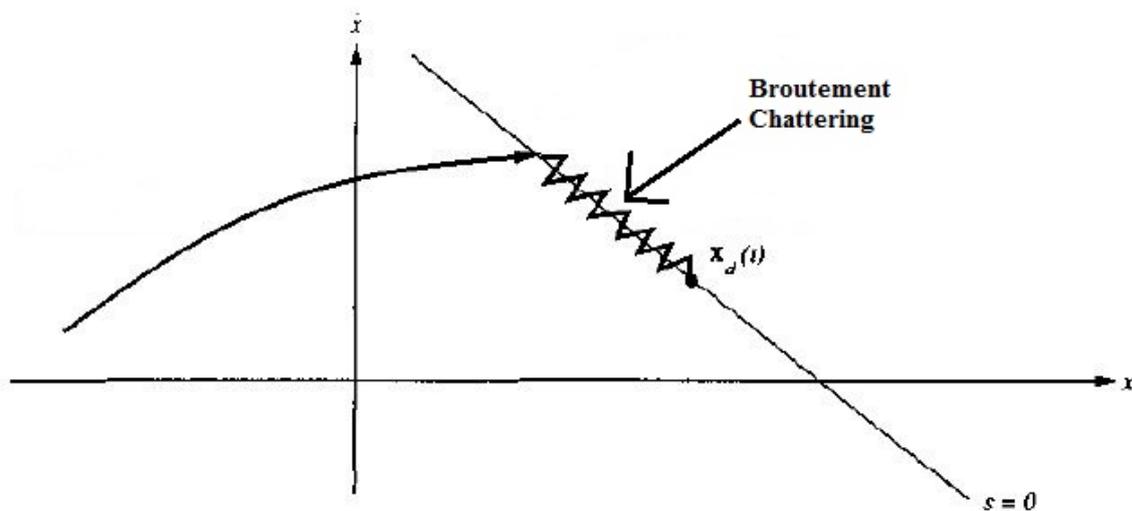


FIGURE 3.11 – Phénomène de broutement

3.4.2.6 Solution pour atténuer le phénomène de "chattering"

Dans le but de réduire ou d'éliminer ce phénomène, de nombreuses solutions ont été proposées, comme la solution de couche limite, fuzzy sliding mode, mode glissant d'ordre supérieur, approach law, etc...

La solution à proposer consiste à remplacer la fonction signe par une approxi-

mation continue, par exemple, la fonction sigmoïde [25].

$$f_{\text{sigmoïde}} = \frac{\exp^{b_c s} - 1}{\exp^{b_c s} + 1} \quad (3.19)$$

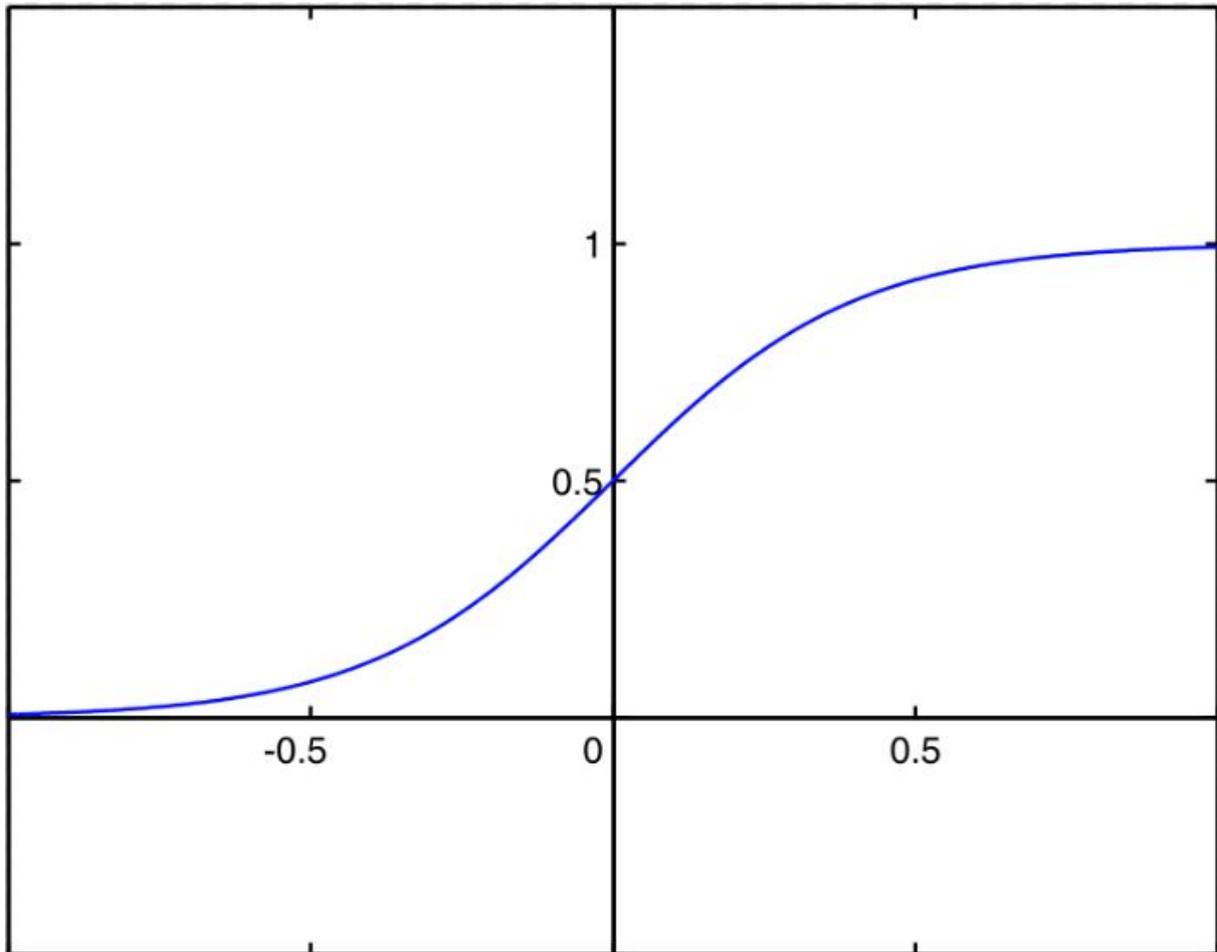


FIGURE 3.12 – La fonction sigmoïde

Le système est dit en régime pseudo-glissant. Bien que cela permette d'atténuer le phénomène de réticence, la précision par rapport à l'objectif fixé, la robustesse de la commande et le temps de réponse s'en trouvent dépréciés.

Cette méthode est paramétrée par une constante positive b_c réglée pour avoir un bon compromis entre réduction du chattering et conservation de la robustesse. Dans la méthodes présentées ici, plus b_c est petit, plus l'approximation tend vers la fonction signe, et donc meilleure est la robustesse, au détriment de la réduction du chattering.

3.4.3 Test et simulation

Dans cette section, l'efficacité des régulateurs proposés sera montrée dans la simulation.

L'étude sera axée sur la robustesse du contrôleur, suivie d'une comparaison entre cette commande non linéaire et la méthode LQR. Nous allons donc diviser cette partie en deux points.

Pour commencer, nous définissons ces valeurs $b_c = 30$, $\lambda_1 = \lambda_2 = \lambda_3 = 10$, $k_1 = k_2 = k_3 = 10$;

Comme cela a été mentionné précédemment, un régulateur de balancement basé sur le mode glissant sera le seul cas d'étude et non de maintien de position. Pour vérifier la fonctionnalité du contrôle, nous allons appliquer des conditions initiales $\theta_0 = 0.2$ rad et $\phi_0 = 0.3$ rad.

Les résultats suivants sont trouvés (figure 3.12) :

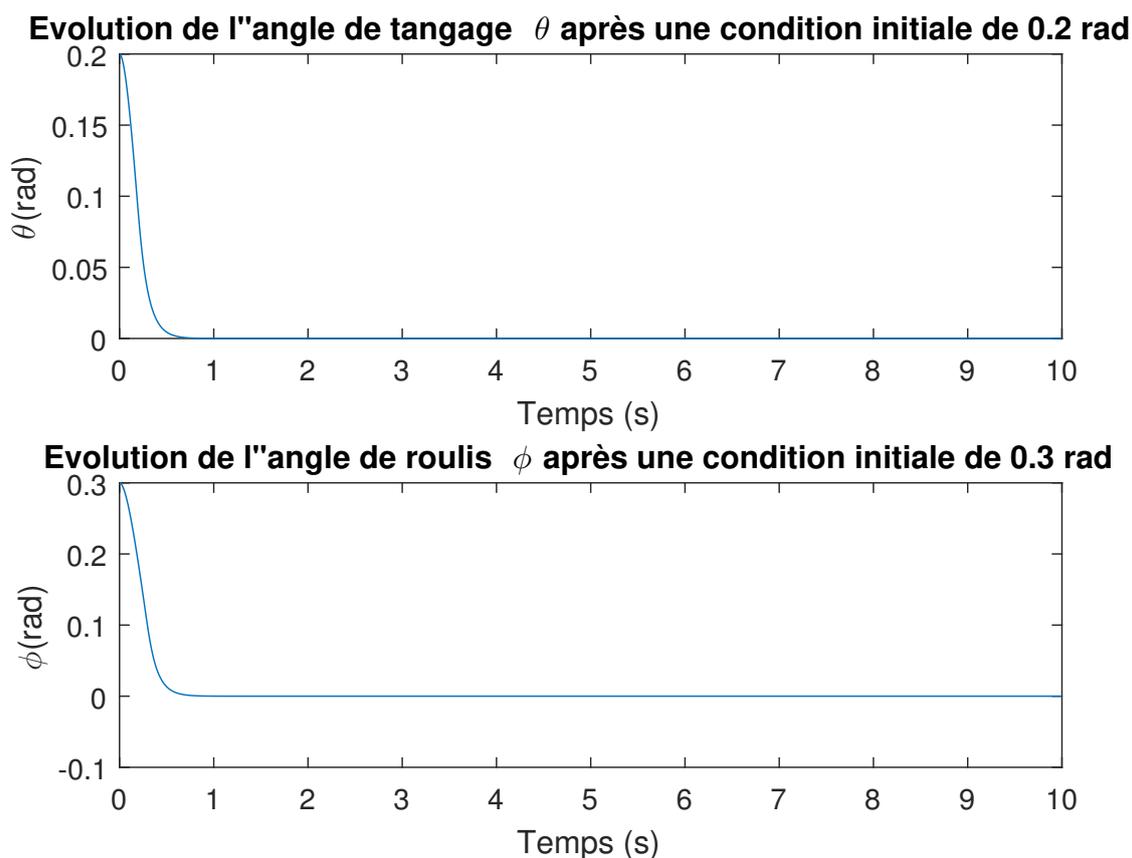


FIGURE 3.13 – Test de balancement avec SMC en injectant une condition initiale

Maintenant, le test va expérimenter le rejet de perturbations par ce régulateur, pour cela nous allons lancer la simulation avec $\theta_0 = 0.2$ rad suivi d'une perturbation

constante de 0.2 rad à $t = 6s$. Une autre perturbation de -0,2 rad sera appliquée à $t = 11s$. Les resultats sont obtenus comme suivant (figure 3.13) :

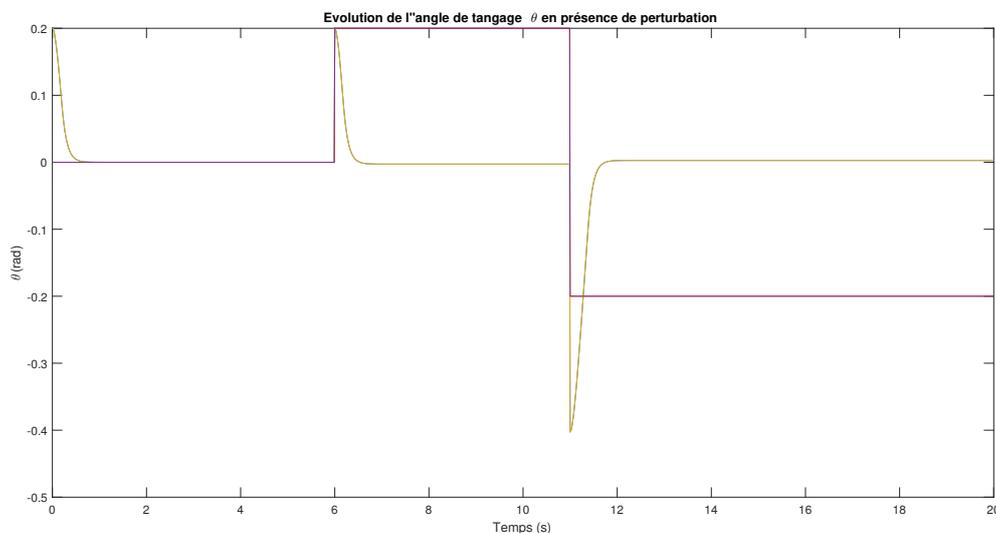


FIGURE 3.14 – Test de balancement avec SMC en présence des différentes perturbation

Nous remarquons que SMC peut rejeter les perturbations dans une durée inférieure de 1s surtout en présence d'une perturbation constante de 0,4 rad. Il n'y a pas des oscillations dans la dynamique du système et le rejet est très rapide par rapport au LQR qui nécessite des secondes pour atteindre le point désiré.

3.5 Etude comparative de deux régulateurs

Cette section est dédiée à la comparaison des deux régulateurs sur la base de plusieurs critères listé ci-après : temps de réponse, dépassement, consommation d'énergie (couple). Cette comparaison est faite en présence des mêmes conditions. On obtient :

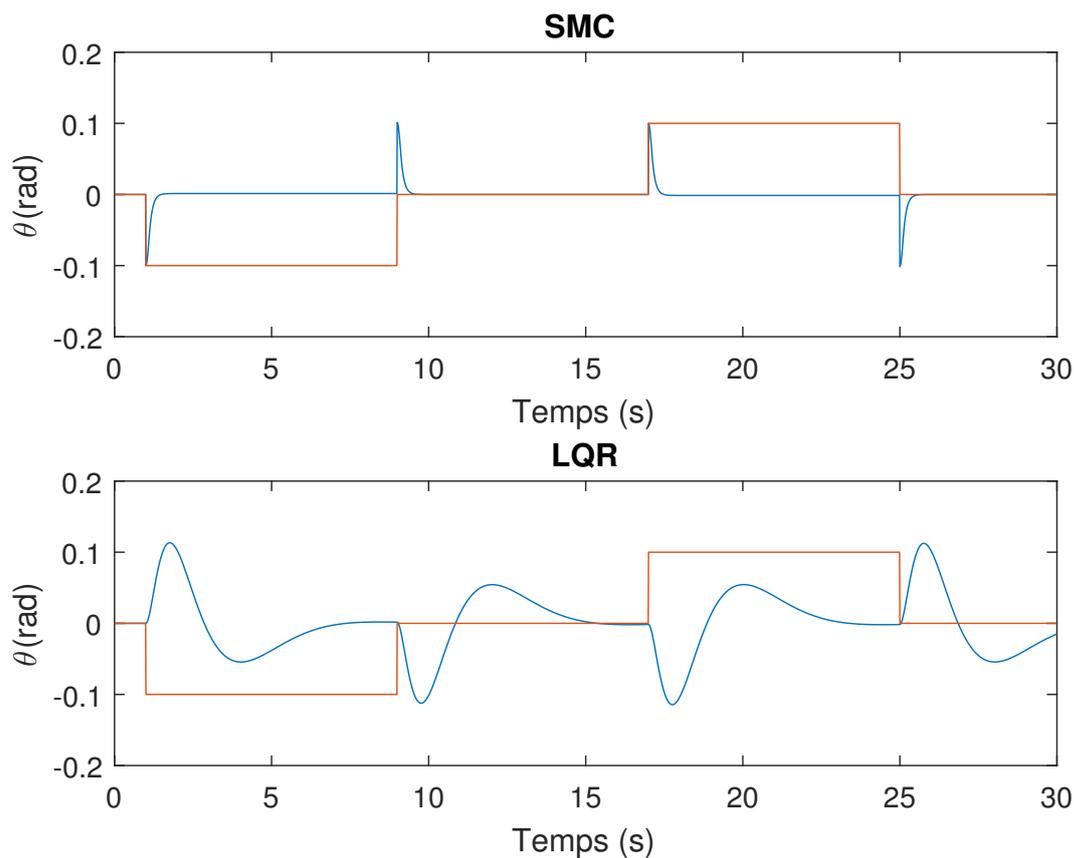


FIGURE 3.15 – Rejet de perturbation pour les deux régulateurs

En observant les graphes, les deux régulateurs remplissent leur objectif consistant à maintenir le robot en position verticale. En ce qui concerne les performances, il est clair que SMC est bien plus performant que LQR. le temps de réponse de SMC ne dépasse pas $1s$ contrairement à celui de LQR qui prend $8s$ pour stabiliser l'état. En outre, bien qu'il n'y ait pas de dépassement dans la réponse lors de l'application de SMC, chaque réponse du système basé sur LQR oscille un peu (2 dépassement) avant la convergence. Maintenant, le critère de l'énergie à consommer (les couples). La figure (3.15) montre les résultats :

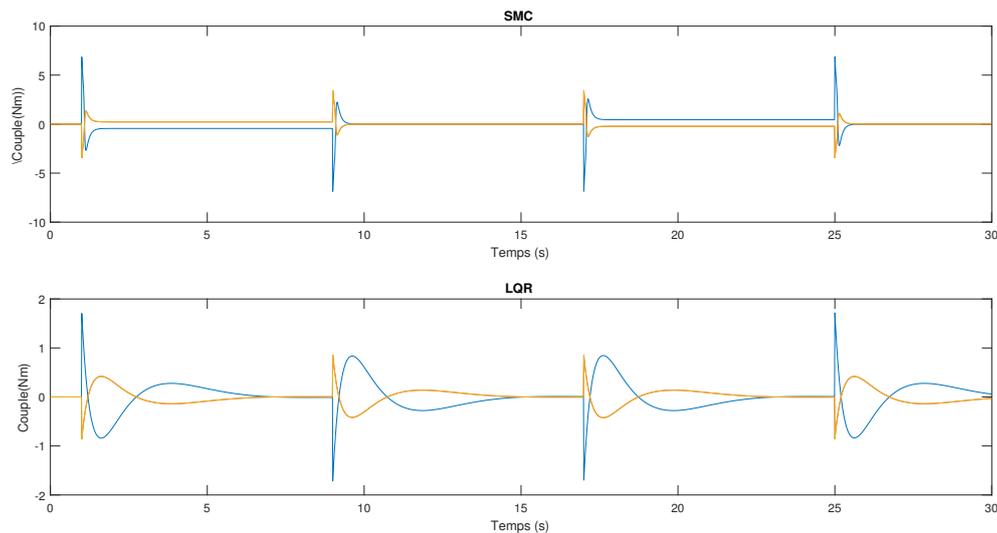


FIGURE 3.16 – Couple généré pour le rejet de perturbation par les deux régulateurs

SMC pousse le système à utiliser un couple élevé pour atteindre des performances et une robustesse uniques. Les valeurs de couple atteignent près de 7 Nm et apparaissent sous forme de pics (haute fréquence). Ceci explique le petit temps de réponse et l'absence de dépassement. D'autre part, LQR utilise des valeurs de couple plus faibles par rapport à SMC. Aussi, Les variations des signaux de commande(couples) sont moins brusque dans le cas LQR, au contraire au SMC.

3.6 Singularités

Dans cette section, nous aborderons un problème survenu plusieurs fois au cours de la simulation ci-dessus. Nous avons remarqué que, dans de nombreux cas, lorsque nous simulons notre modèle en présence ou en absence du régulateur (SMC ou LQR), si les valeurs des conditions initiales dépassent une certaine limite ou si nous injectons des signaux de référence avec des valeurs relativement grand, la simulation s'arrête et nous recevons des messages semblables à :

Warning : Matrix is singular, close to singular or badly scaled. Results may be inaccurate

Cela signifie qu'il ya des domaines dans lesquels notre modèle ne répond pas normalement. Ce problème apparaît de manière répétitive lorsque nous traitons deux états à la fois, par exemple, en posant deux conditions initiales pour θ et ϕ .

Apparemment, les modèles basés sur les angles d'Euler présentent cet inconvénient appelé "Blocage du cardan"(gimbal lock en anglais).C'est la perte d'un degré de

liberté qui survient quand les axes de deux des trois cardans nécessaires pour appliquer ou compenser les rotations dans l'espace à trois dimensions sont portés par la même direction. La seule explication possible serait celle-ci puisque tous les calculs ont été effectués correctement. La seule solution pour ce problème est de changer le modèle basé sur les angles d'Euler et essayer de le simplifier ou simplement, changer complètement la méthode de modélisation.

3.7 Conclusion

Dans ce chapitre, nous avons développé deux méthodes de commande, l'une linéaire, LQR, et la seconde, une méthode non linéaire appelée SMC. Des simulations ont été effectuées sur le modèle du robot et nous concluons que : les deux régulateurs atteignent leurs objectifs désirés, tels que l'opération de balancement. Mais il existe une différence énorme en comparant les performance entre les deux lois. Nous constatons que SMC donne des résultats étonnants mais consomme énormément l'énergie en retour alors que LQR donne des résultats acceptables, mais consomme moins d'énergie pour atteindre cet objectif. Clairement, selon ces simulations et ces résultats, SMC semble être le meilleur choix pour commander ce robot, mais d'un autre point de vue, le choix dépendra fortement du matériel disponible, allant des limites de couple des moteurs à la capacité de calcul des microcontrôleurs.

Dans le chapitre suivant, la conception mécanique sera discutée et montrée, ainsi que la partie électronique.

Chapitre 4

Conception et implémentation

4.1 Introduction

Ce chapitre sera consacré à expliquer la procédure de conception du prototype du robot. Dans un premier temps, nous aborderons le 3D design CAD (SolidWorks), puis nous discuterons la réalisation mécanique. Finalement, on va présenter la partie électronique nécessaires pour l'actionnement.

4.2 Conception mécanique du robot

4.2.1 Forme du Robot

La forme générale du robot est cylindrique. Il y a 3 étagères en parallèles, celle en bas tient 3 moteurs inclinés par rapport le plan horizontal avec des roues spéciales reliées à la partie terminale. Le lien entre le robot et le sol est une sphère qui joue le rôle du moyen du déplacement. La figure (3.1) montre le design 3D CAD du robot.

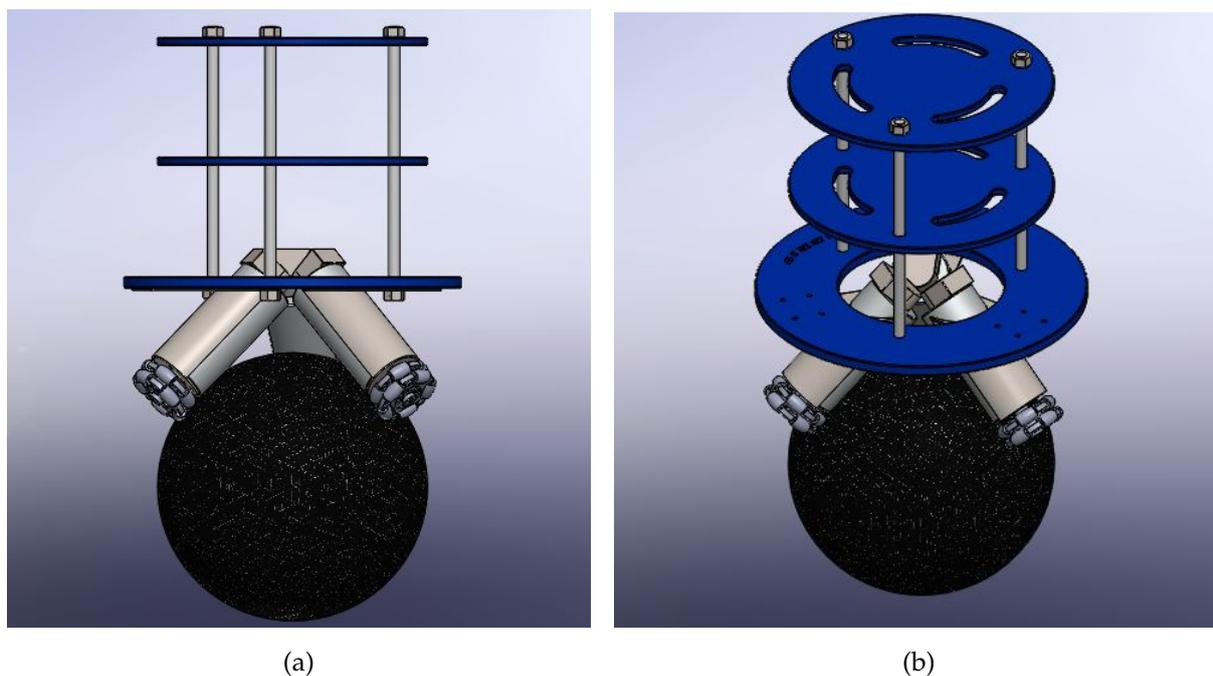


FIGURE 4.1 – Modèle 3D CAD du prototype du robot

Comme il apparaît, il y a 3 tiges en fer pour garder les étagères en position. Dans ma conception, j'ai choisi de mettre 3 étagères qui sont destinées à supporter les pièces électroniques. La base en bas est censée contenir les pilotes des moteurs. L'étagère intermédiaire va porter le microcontrôleur et à l'étagère supérieur, l'IMU est placé ainsi que la batterie. Il est possible de monter plus de 3 bases, tout dépend le nombre des

composants électroniques à insérer, tels qu'une mini caméra pour l'analyse des trajectoires et des modules sans fil permettant de commander à distance le système, sans oublier la possibilité de placer un affichage comme outil de communication avec le robot.

4.2.2 Les étagères

Comme indiqué dans la figure (4.1), les étagères ont été usiné d'une forme de disque. Le deux étagères supérieurs (Figure 4.3) sont identiques et la base en bas (Figure 4.2) est un peu plus large. La matière utilisée est l'alucobond. Les dimensions sont montés sur les figures. La figure 4.4 montre les étagères réelles.

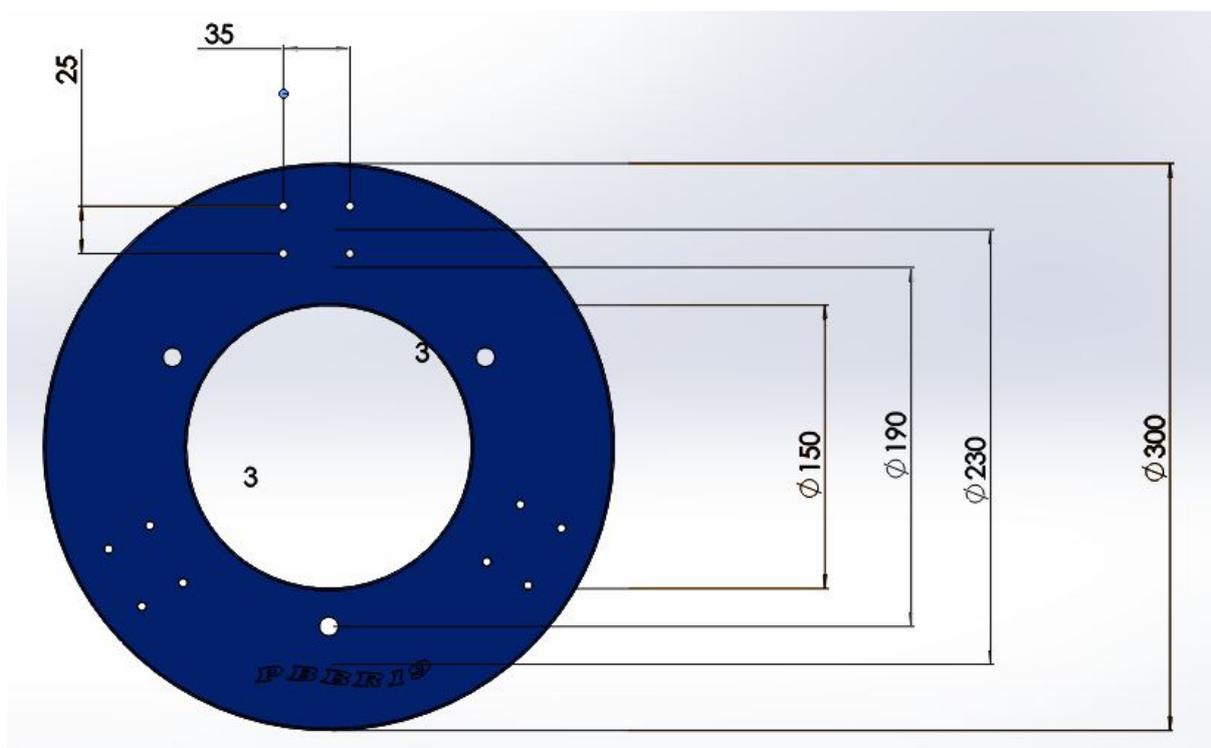


FIGURE 4.2 – La base en bas

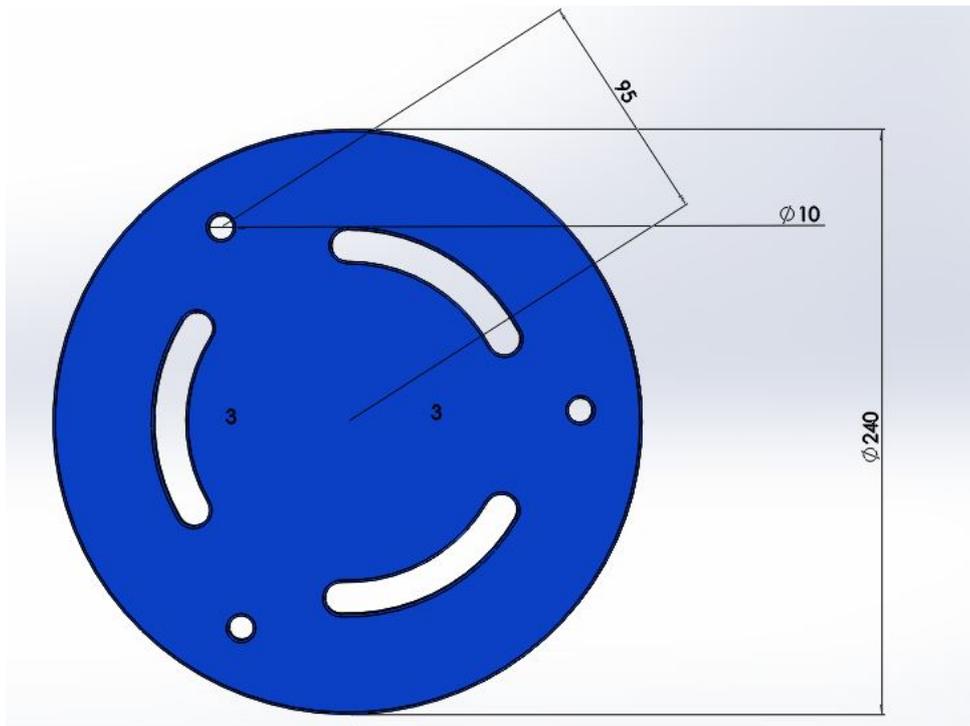


FIGURE 4.3 – L'étagère supérieur



FIGURE 4.4 – Etagière usiné

4.2.3 Les roues omnidirectionnelles

Une roue omnidirectionnelle ou bien holonome est un mécanisme constitué d'un moyeu disposant d'un chapelet de galets répartis sur sa périphérie. Utilisée pour les

plateformes robotiques, elle confère au robot des possibilités de déplacements dans un espace plus compact en actionnant la sphère selon une direction et permettant le mouvement dans la direction perpendiculaire (Figure 4.5 et 4.6).

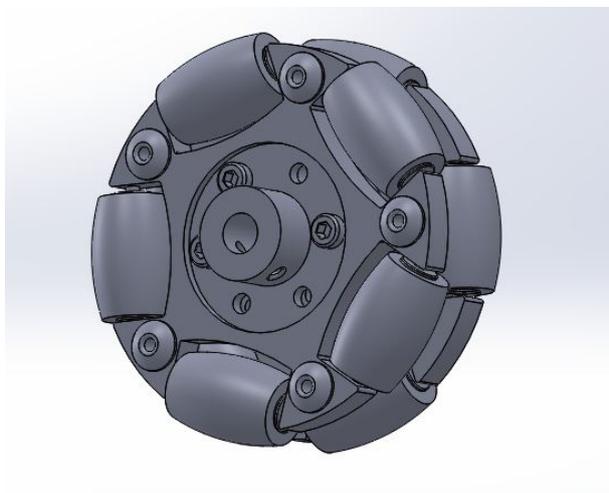


FIGURE 4.5 – Design 3D CAD de la roue FIGURE 4.6 – La roue omnidirectionnelle

Puisque la structure choisie du robot dépend de 3 moteurs d’actionnement, 3 roues holonome seront utilisées.

4.2.4 Les supports des moteurs

Ces supports constituent la partie la plus importante de la structure du robot car ils maintiennent les moteurs dans une position inclinée de 45 degrés par rapport à l’horizontale, il est donc nécessaire d’être usiné parfaitement. La figure 4.7 montr le design 3D des différentes pièces incluses afin de fabriquer le support final (Figure 4.8). La matière utilisée est l’acier vue que les moteurs sont un peu lourd ainsi qu’on peut minimiser les vibrations lors de mouvement.

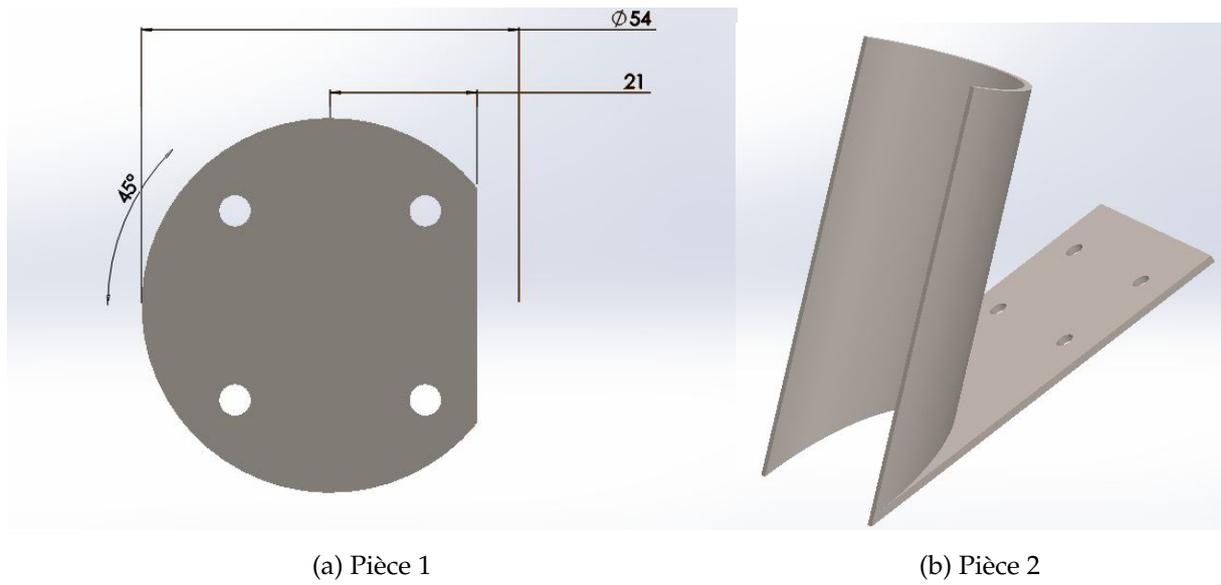


FIGURE 4.7 – Pièces du support

Les deux pièces seront assemblées pour avoir le support complet (Figure 4.8) :

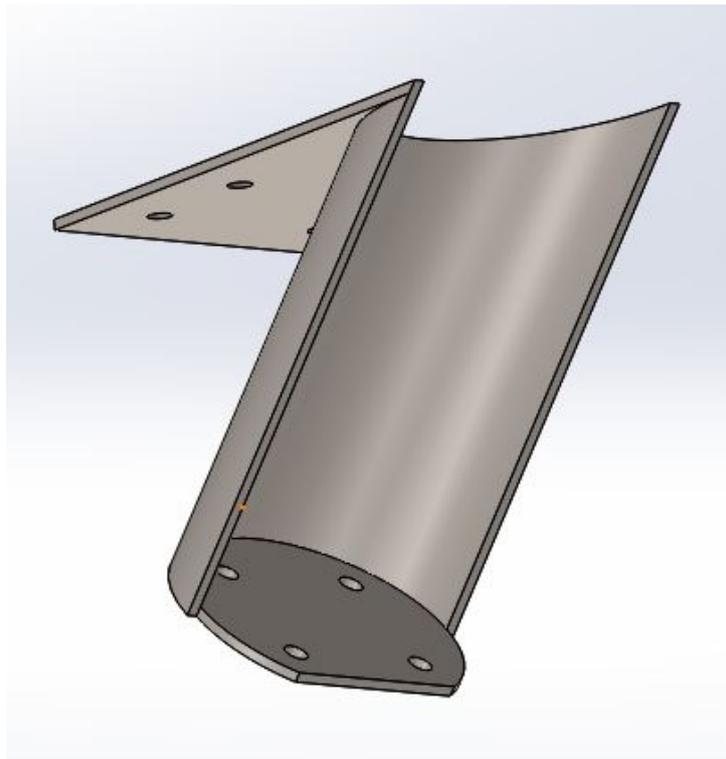


FIGURE 4.8 – Design 3D CAD du support

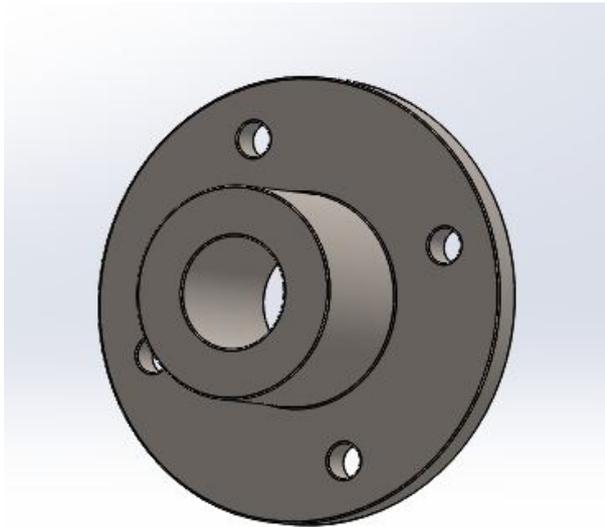
La pièce assemblé et usiné est montée dans la figure 4.9 :



FIGURE 4.9 – Support usiné

4.2.5 Les moyaux des roues

Pour que les roues soient reliées aux moteurs, il est nécessaire d'usiner ce qu'on appelle les moyeux, assurant la roue tourne avec l'axe du moteur (Figure 4.10). La matière utilisé pour l'usinage est l'aluminium.



(a) Design 3D CAD du moyau



(b) Moyau usiné

FIGURE 4.10 – Moyau

4.2.6 Assemblage du robot

On a conçu le robot dans une forme qui garantit la facilité de l'assemblage, le changement et l'adjonction des pièces. Les images suivantes montrent comment on a assemblé les pièces du robot. On a utilisé les images de design 3D parce que elles sont plus claires que les images réelles.

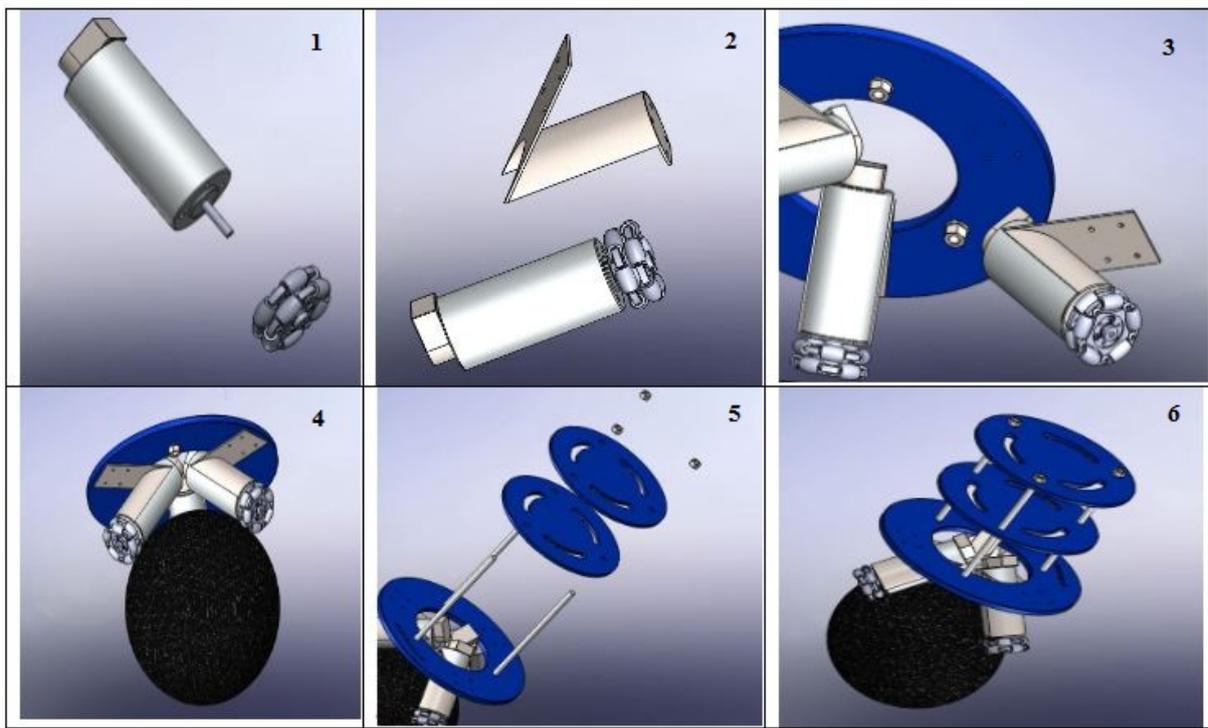


FIGURE 4.11 – L'assemblage de la structure du robot

La structure complète du robot est montrée dans la figure 4.11 :



(a)



(b)

FIGURE 4.12 – Structure réelle du robot

4.3 Modules électronique

Dans cette partie on va décrire les composants électriques qui sont utilisés dans le robot.

4.3.1 Le microcontrôleur Arduino 2560

L'Arduino Mega est une carte microcontrôleur basé sur l'ATmega1280. Il dispose de 54 broches numériques d'entrée / sortie (dont 14 peuvent être utilisées comme sorties PWM), 16 entrées analogiques, 4 UART (ports série matériels), un oscillateur en cristal de 16 MHz, d'une connexion USB, une prise d'alimentation, d'une embase ICSP et un bouton de réinitialisation. Il contient tout le nécessaire pour soutenir le microcontrôleur, il suffit de le connecter à un ordinateur avec un câble USB, avec un adaptateur AC-DC ou batterie pour commencer.[26]

La programmation sur Arduino se fait par l'interface Arduino IDE avec langage de programmation C ou C++.



FIGURE 4.13 – Carte Arduino Mega 2560

4.3.2 Moteurs

Chaque roue est motorisée indépendamment de l'autre, offrant le contrôle du robot par différentiation. le moteur utilisé est le DCM50205D1000[27]. C'est un servo-



FIGURE 4.15 – Servo Drive DCS810

4.3.4 Capteurs

4.3.4.1 Gyroscope

Il est nécessaire pour un système de balancement d'avoir la possibilité de déterminer l'angle d'inclinaison. Ceci peut être obtenu par l'utilisation d'un gyroscope.

Le gyroscope est un appareil qui exploite le principe de la conservation du moment angulaire en physique (ou encore stabilité gyroscopique ou effet gyroscopique). Il donne la position angulaire (selon un, deux ou trois axes). Il ne donne pas directement un angle d'orientation. Cet angle s'obtient par intégration dans le temps de la vitesse angulaire, en faisant attention au cumul des erreurs de dérivée. Le gyroscope permet de déterminer la rotation ressentie par l'objet auquel il est attaché, il calcule la vitesse angulaire en degrés par seconde. Dans notre cas il va servir à atténuer les bruits et à déterminer l'angle lorsqu'il y a des mouvements brusques ce qui est difficile à mesurer par l'accéléromètre.

4.3.4.2 Accéléromètre

Le gyroscope seul ne peut pas fournir la valeur exacte de l'angle d'inclinaison. On doit donc ajouter un autre capteur pour estimer cette dernière.

L'accéléromètre est un élément important dans la conception d'un système d'acquisition de données, car il permet de mesurer l'accélération linéaire d'un objet mobile selon 3 axes orthogonaux(X, Y et Z). À partir de cette mesure, l'utilisateur peut déduire le déplacement et la vitesse de l'objet mobile. Mais en raison de forces brusques et sa sensibilité aux vibrations qui pourraient induire des erreurs, on utilise un gyroscope combiné avec l'accéléromètre pour pallier à ce problème.

4.3.4.3 Magnétomètre

Un magnétomètre est un appareil qui sert à mesurer selon les cas l'intensité ou la direction d'un champ magnétique, ou l'aimantation d'un échantillon. Il repose sur un élément sensible au champ magnétique : le capteur de champ magnétique qui, lorsqu'il est associé à un dispositif électronique, permet d'extraire la mesure du champ magnétique. Il en existe différents types, basés sur des principes physiques différents .

4.3.4.4 SparkFun 9DoF Razor IMU M0

L'introduction précédente a été faite puisque le fonctionnement de la centrale à inertie est basé sur ces trois concepts pour mesurer avec précision l'inclinaison du robot. Le choix de ce module est le module SparkFun 9DoF Razor IMU M0.

Le SparkFun 9DoF Razor IMU M0 [29] associe un microprocesseur SAMD21 à un capteur MPU-9250 9DDL pour créer une unité IMU ré-programmable et multifonctionnelle. Il peut être programmé pour surveiller et enregistrer les mouvements, transmettre les angles d'Euler sur un port série.

Le MPU-9250 de 9DoF Razor est doté de trois capteurs à 3 axes (accéléromètre, gyroscope et magnétomètre) qui lui permettent de détecter l'accélération linéaire, la vitesse de rotation angulaire et les vecteurs de champ magnétique. Le microprocesseur intégré, le SAMD21 d'Atmel, est un microcontrôleur ARM 32 bits ARM Cortex-M0 + compatible avec Arduino.

Outre une paire de circuits intégrés, le module 9DoF Razor IMU comprend un chargeur de batterie LiPo, un commutateur de contrôle de l'alimentation et une multitude de commutations d'E / S.

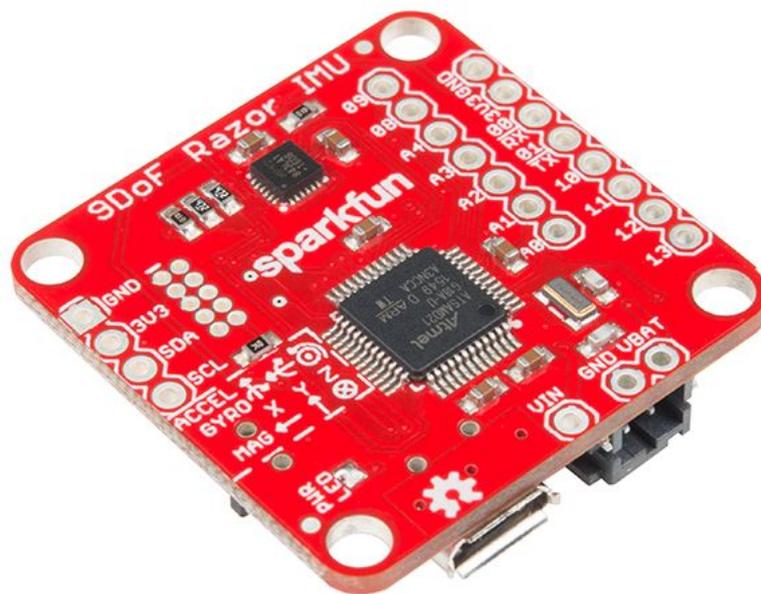


FIGURE 4.16 – SparkFun 9DoF Razor IMU M0

4.4 Simulation graphique

La partie électronique n'a pas pu être réalisée. La principale raison qui empêche notre projet d'être achevé est l'absence de certains éléments électroniques nécessaires pour configurer l'environnement et établir la connexion entre les moteurs et les pilotes (câble RS232-RJ11) . De plus, la sphère qui est supposée être la roue du robot, qui est un ballon de basket (le plus gros ballon du marché pouvant jouer ce rôle), est un peu petite, donc le contact entre le ballon et les trois roues n'est pas assurée. Tous ces problèmes ne peuvent pas être résolus en raison de la non disponibilité des composants nécessaires sur le marché algérien, en plus la contrainte de temps. Cette solution alternative est proposée. La simulation 3D jouera le rôle du vrai robot et convertira toute la simulation et la réponse en animation 3D. VR sink est l'application utilisée pour créer le robot virtuel.

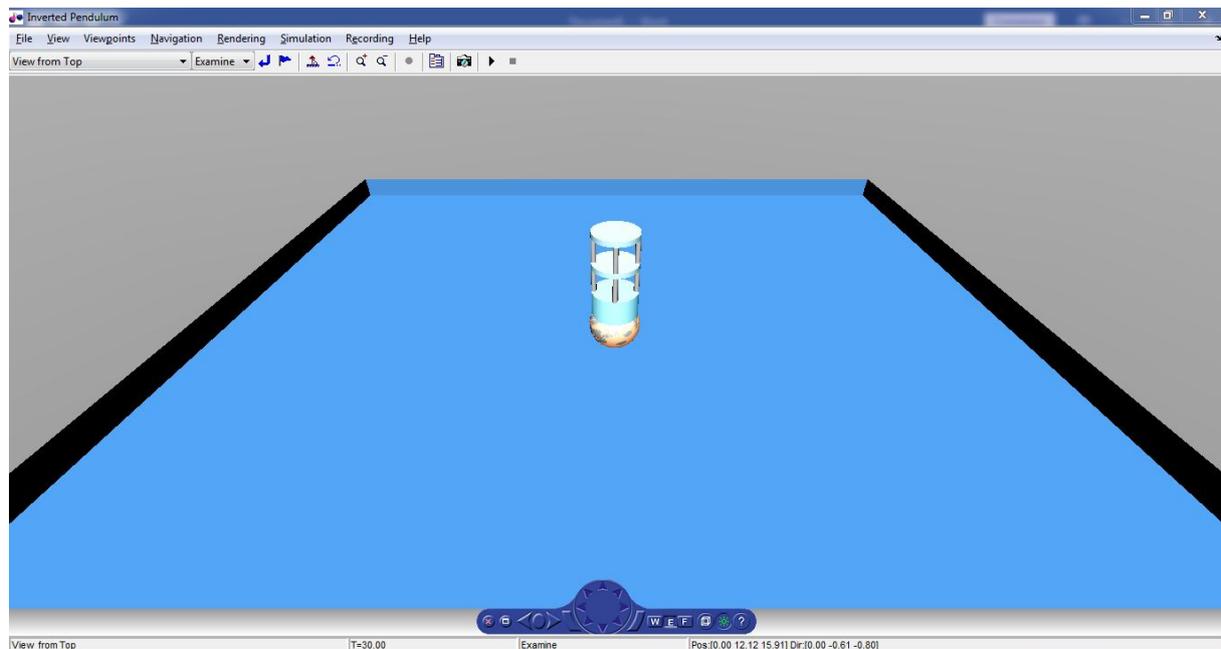


FIGURE 4.17 – Environnement VR Sink

4.5 Conclusion

La conception mécanique a été faite. La structure est prête mais, en raison de la contrainte de temps, nous n'avons pas pu terminer avec la partie électronique. Nous n'avons donc listé que le matériel électronique nécessaire pour établir l'environnement (Microcontrôleur+capteurs+driver) et introduire les lois de commande dans le système réel.

Conclusion générale

Dans le cadre du projet de fin d'études, nous avons étudié un type spécial des robots capable de s'équilibrer sur une sphère et de rester en position verticale. Il utilise aussi cette sphère comme roue pour se déplacer et suivre une trajectoire. Ce projet nous a permis d'aborder les aspects et les techniques qu'on a étudiées durant notre formation théorique.

Le premier chapitre était une compilation détaillée sur les projets les plus réussis dans l'étude du robot auto équilibrant à une roue. Nous avons également abordé les différentes méthodes de la modélisation ainsi que les lois de commande appliquée.

Le deuxième chapitre est consacré à la modélisation du robot. Pour y parvenir, nous avons utilisé la théorie des torseurs basée sur les transformation homogène et torseurs cinématique et dynamique. Nous avons étudié à la fois la cinématique et la dynamique du système et expliqué tous les calculs mentionnés. Finalement, nous construisons un modèle non linéaire 3D complet basé sur les angles d'Euler.

Le troisième chapitre était consacré à l'étude des lois de commande appliquées pour assurer la stabilité dynamique. Nous avons proposé deux régulateurs, LQR et SMC. Nous construisons les algorithmes associés sur la base du modèle 3D et simulons les réponses correspondant à diverses conditions. Nous finissons par la comparaison entre les deux régulateurs en fonction des résultats de la simulation. Alors que LQR était simple à mettre en œuvre, il est limité car il est linéaire. SMC était difficile à implémenter car le système est sous-actionné, nous avons donc dû le diviser en deux sous-systèmes. Cependant, sa performance est supérieure à celle de LQR.

Dans le quatrième chapitre, on a présenté la conception du robot et une descrip-

tion détaillée sur toutes les pièces utilisées pour réaliser le robot. La réalisation du robot contient deux parties, la partie mécanique qui concerne la construction du robot, et la partie électrique qui discute l'environnement idéal pour implémenter les lois de commande.

A cause d'une contrainte de temps, nous ne sommes pas arrivés à compléter la partie électronique et par suite l'implémentation de la commande. Notre premier objectif à court terme est donc de mettre en œuvre les lois de commande développées dans ce projet. Dès que cette étape sera terminée et que le système complet sera réalisé, l'objectif à long terme est de travailler sur des lois de commande plus avancée pour améliorer les performances du robot.

En outre, afin de résoudre les problèmes de singularité, je propose comme un travail futur, que la modélisation du système sera modifiée en appliquant de méthodes plus efficaces telles que la modélisation basé sur quaternion, ou l'application des lois de commande qui ne nécessitent pas un modèle, par exemple "Time Delay Control -TDC".

Bibliographie

- [1] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2884–2889, IEEE, 2006.
- [2] P. Fankhauser and C. Gwerder, "Modeling and control of a ballbot," B.S. thesis, Eidgenössische Technische Hochschule Zürich, 2010.
- [3] U. Nagarajan, G. Kantor, and R. Hollis, "The ballbot : An omnidirectional balancing mobile robot," *The International Journal of Robotics Research*, vol. 33, no. 6, pp. 917–930, 2014.
- [4] M. Kumagai and T. Ochiai, "Development of a robot balancing on a ball," in *2008 International Conference on Control, Automation and Systems*, pp. 433–438, IEEE, 2008.
- [5] C.-K. Chan and C.-C. Tsai, "Intelligent backstepping sliding-mode control using recurrent interval type 2 fuzzy neural networks for a ball-riding robot," in *2012 International conference on Fuzzy Theory and Its Applications (iFUZZY2012)*, pp. 169–174, IEEE, 2012.
- [6] C.-K. Chan and C.-C. Tsai, "Direct adaptive recurrent interval type 2 fuzzy neural networks control using for a ball robot with a four-motor inverse-mouse ball drive," in *2013 International Conference on Advanced Robotics and Intelligent Systems*, pp. 5–10, IEEE, 2013.
- [7] C.-W. Liao, C.-C. Tsai, Y. Y. Li, and C.-K. Chan, "Dynamic modeling and sliding-mode control of a ball robot with inverse mouse-ball drive," in *2008 SICE Annual Conference*, pp. 2951–2955, IEEE, 2008.
- [8] C.-K. Chan and C.-C. Tsai, "Intelligent backstepping sliding-mode control using recurrent interval type 2 fuzzy neural networks for a ball robot with a four-motor

- inverse-mouse ball drive,” in *2012 Proceedings of SICE Annual Conference (SICE)*, pp. 1281–1286, IEEE, 2012.
- [9] C.-C. Tsai, C.-K. Chan, and L.-C. Kuo, “Lqr motion control of a ball-riding robot,” in *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 861–866, IEEE, 2012.
- [10] M. Kumaga and T. Ochiai, “Development of a robot balanced on a ball—application of passive motion to transport—,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 4106–4111, IEEE, 2009.
- [11] U. Nagarajan and R. Hollis, “Shape space planner for shape-accelerated balancing mobile robots,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1323–1341, 2013.
- [12] C.-C. Tsai, M.-H. Juang, C.-K. Chan, C.-W. Liao, and S.-J. Chan, “Self-balancing and position control using multi-loop approach for ball robots,” in *2010 International Conference on System Science and Engineering*, pp. 251–256, IEEE, 2010.
- [13] A. N. Inal, Ö. Morgül, and U. Saranlı, “A 3d dynamic model of a spherical wheeled self-balancing robot,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5381–5386, IEEE, 2012.
- [14] A. Lotfiani, M. Keshmiri, and M. Danesh, “Dynamic analysis and control synthesis of a spherical wheeled robot (ballbot),” in *2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pp. 481–486, IEEE, 2013.
- [15] J. Blonk, “Modeling and control of a ball-balancing robot,” Master’s thesis, University of Twente, 2014.
- [16] L. Hertig, D. Schindler, M. Bloesch, C. D. Remy, and R. Siegwart, “Unified state estimation for a ballbot,” in *2013 IEEE International Conference on Robotics and Automation*, pp. 2471–2476, IEEE, 2013.
- [17] W. Wang, J. Yi, D. Zhao, and D. Liu, “Design of a stable sliding-mode controller for a class of second-order underactuated systems,” *IEE Proceedings-Control Theory and Applications*, vol. 151, no. 6, pp. 683–690, 2004.
- [18] Y.-F. Peng, C.-H. Chiu, W.-R. Tsai, and M.-H. Chou, “Design of an omni-directional spherical robot : Using fuzzy control,” in *Proceedings of the international multiconference of engineers and computer scientists*, vol. 1, pp. 18–20, Citeseer, 2009.

- [19] E. Pellegrini, K. J. Diepold, R. Dessort, and H. Panzer, "3d-modeling of a robot balancing on a ball," tech. rep., Lehrstuhl für Regelungstechnik, 2011.
- [20] M. Boulouiha, "Techniques de commande avancée," *Université de Rélizane*, 2015.
- [21] J. Komatsu, *Characterizing the Potential of Active Aircraft Diversion Technologies for Remotely Redirecting an Aircraft*. PhD thesis, Tufts University, 2011.
- [22] I. Deghboudj, "Commade des systèmes non linéaires par mode glissant d'ordre superieur," Université Constantine 1, Novembre 2013.
- [23] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.
- [24] A. Benamor, L. Chrifi-Alaoui, H. Messaoud, and M. Chaabane, "Sliding mode control, with integrator, for a class of mimo nonlinear systems," *Engineering*, vol. 3, no. 05, p. 435, 2011.
- [25] M. Van, S. S. Ge, and H. Ren, "Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control," *IEEE transactions on cybernetics*, vol. 47, no. 7, pp. 1681–1693, 2016.
- [26] <https://store.arduino.cc/mega-2560-r3>.
- [27] <http://www.leadshine.com/productdetail.aspx?type=products&category=servo-products&producttype=brushed-servo-motors&series=DCM&model=DCM50205>.
- [28] <http://www.leadshine.com/productdetail.aspx?type=products&category=servo-products&producttype=brushed-servo-drives&series=DCS&model=DCS810>.
- [29] <https://www.sparkfun.com/products/14001>.
- [30] R. M. Murray, Z. Li, S. S. Sastry, and S. Sastry, "A mathematical introduction to robotic manipulation, 1994," *Boca Raton, FL : CRC*.

Annexes

.1 Annexe A : Paramètres

Symbole	Définition
$\alpha(t)$	Position angulaire de la sphère par rapport l'axe x
$\beta(t)$	Position angulaire de la sphère par rapport l'axe y
$\gamma(t)$	Position angulaire de la sphère par rapport l'axe z
$\theta(t)$	Angle de tangage
$\phi(t)$	Angle de roulis
$\psi(t)$	Angle de lacet
$x_k(t)$	Position de la sphère selon l'axe x
$y_k(t)$	Position de sphère selon l'axe y
m_s	La masse de la sphère
m_b	La masse du corps du robot
r_s	Le rayon de la sphère
r_w	Le rayon des roues omnidirectionnelles
g	Constante de gravité
j_s	Moment d'inertie de la sphère par rapport les trois axes(x,y,z)
I	Tenseur d'inertie
T	Matrice homogène
v	Vitesse linéaire
ω	Vitesse angulaire
Υ	Torseur cinématique
W	Torseur dynamique
δ	Angle d'inclinaison des moteurs par rapport l'horizontale
τ	Couple
f	Force
J	Matrice jacobienne
M	Matrice d'inertie
C	Matrice de Coriolis
G	Matrice de gravité
s	Surface de glissement

.2 Annexe B : Notions mathématiques préliminaires

Dans cet annexe, les notions mathématiques préliminaires seront traités.

.2.1 Torsuer cinématique

Dans cette section, les concepts de base de la théorie de torsuer cinématique et leur notation utilisés dans ce travail sont introduits (Repères, Matrice de Rotation et Matrice Homogène)[30].

Pour exprimer l'orientation et la position d'un objet dans l'espace, différents repères de coordonnées seront utilisés. Un repère est noté \mathfrak{R}_i et consiste en trois vecteurs orthonormaux $(x, y, z) \in \mathbb{R}^3$. Les repères sont liés les uns aux autres par rotation et translation.

La rotation entre les repères \mathfrak{R}_a et \mathfrak{R}_b est désignée par la matrice de rotation R_a^b . La matrice de rotation appartient au groupe orthogonal spécial $SO(3)$.

$$SO(3) := \{R \in \mathbb{R}^{3 \times 3}; RR^T = I, \det R = +1\} \quad (1)$$

Notez que l'inverse de la matrice de rotation est égal à sa transposée :

$$(R_a^b)^{-1} = (R_a^b)^T = R_b^a$$

Une matrice de rotation R_1^n peut être trouvée en multipliant les matrices de rotation des repères intermédiaires :

$$R_1^n = R_{n-1}^n \dots R_2^3 R_1^2 \quad (2)$$

Cette multiplication des matrices donne la rotation du repère 1 à 2, suivie par la rotation du repère 2 à 3, etc. L'algèbre de Lie de la matrice de rotation $R \in SO(3)$ est donnée par $\dot{R}R^T$ et $R^T \dot{R}$, ce qui donne une matrices asymétriques appartenant à $so(3)$.

$$so(3) := \{\tilde{\omega} \in \mathbb{R}^{3 \times 3}; \tilde{\omega}^T = -\tilde{\omega}\} \quad (3)$$

Ici ω est la vitesse angulaire instantanée. La vitesse angulaire $\omega_a^{c,b} \in \mathbb{R}^3$ représente la vitesse angulaire de \mathfrak{R}_a par rapport à \mathfrak{R}_b exprimée dans le repère \mathfrak{R}_c et est définie par :

$$\tilde{\omega}_b^{b,a} = R_a^b \dot{R}_b^a \quad (4)$$

$$\tilde{\omega}_b^{a,a} = R_b^a \dot{R}_a^b \quad (5)$$

Remarque :

Soit un vecteur $a \in \mathbb{R}^3$ tel que $a = [a_x \ a_y \ a_z]^T$ donc la forme "tilde" de ce vecteur est donnée par :

$$\tilde{a} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}$$

Le produit vectoriel de deux vecteurs $(a, b) \in \mathbb{R}^3$ peut également être exprimé en utilisant cette forme comme suivant :

$$a \times b = \tilde{a}b$$

La vitesse angulaire peut être réécrite dans un repère différent en utilisant la relation :

$$\omega_j^{b,k} = R_a^b \omega_j^{a,k} \quad (6)$$

La translation entre les repère \mathfrak{R}_a et \mathfrak{R}_b est donnée par le vecteur de translation $p_a^b \in \mathbb{R}^3$. En utilisant les concepts de rotation et de translation, il est possible de définir une matrice homogène, utilisée pour la transformation entre deux repères. La matrice homogène est notée T_a^b et dénote la transformation du repère \mathfrak{R}_a à \mathfrak{R}_b et est donnée par :

$$T_a^b = \begin{pmatrix} R_a^b & p_a^b \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{4 \times 4} \quad (7)$$

Et l'inverse de cette matrice :

$$(T_a^b)^{-1} = T_b^a = \begin{pmatrix} (R_a^b)^T & -(R_a^b)^T p_a^b \\ 0 & 1 \end{pmatrix} \quad (8)$$

Les matrices homogènes font partie du groupe euclidien spécial $SE(3)$:

$$SE(3) := \left\{ \begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} ; R \in SO(3), p \in \mathbb{R}^3 \right\} \quad (9)$$

Une matrice homogène T_1^n peut être trouvée en multipliant les matrices homogènes des repères intermédiaires :

$$T_1^n = T_{n-1}^n \dots T_2^3 T_1^2 \quad (10)$$

Cette multiplication des matrices donne la transformation du repère 1 à 2, suivie par la transformation du repère 2 à 3, etc.

Un point $p^a \in \mathbb{R}^3$ exprimé dans \mathfrak{R}^a peut être exprimé en repère \mathfrak{R}^b en utilisant la relation suivante :

$$\begin{pmatrix} p^b \\ 1 \end{pmatrix} = T_a^b \begin{pmatrix} p^a \\ 1 \end{pmatrix} \quad (11)$$

Notez que les points p^i , avec $i = \{a, b\}$, sont augmentés avec 1. Dans la suite de ce travail, ce vecteur augmenté sera noté P^i , où l'indice supérieur désigne le repère. La matrice homogène constitue la matrice de transformation entre les repères et peut être considérée comme les relations cinématiques entre différents repères.

L'algèbre de Lie de $T \in SE(3)$ est donnée par $\dot{T}T^{-1}$ et $T^{-1}\dot{T}$ appartient à $se(3)$

$$se(3) := \left\{ \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} ; \tilde{\omega} \in se(3), v \in \mathbb{R}^3 \right\} \quad (12)$$

Ici $v_a^{c,b}$ est la vitesse linéaire entre la vitesse de \mathfrak{R}_a par rapport à \mathfrak{R}_b exprimée dans le repère \mathfrak{R}_c . L'algèbre de Lie est définie comme un torseur, qui est la généralisation des vitesses pour un corps rigide. La torseur $\Upsilon_a^{c,b}$ exprime la vitesse linéaire et angulaire de \mathfrak{R}_a par rapport à \mathfrak{R}_b exprimée dans le repère \mathfrak{R}_c et est définie par :

$$\tilde{\Upsilon}_b^{b,a} = T_a^b \dot{T}_b^a \quad (13)$$

$$\tilde{\Upsilon}_b^{a,a} = \dot{T}_b^a T_a^b \quad (14)$$

et a la structure suivante :

$$\Upsilon = \begin{pmatrix} \omega \\ v \end{pmatrix} \in \mathbb{R}^6, \quad \tilde{\Upsilon} = \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{4 \times 4}$$

Les torseurs peuvent être réécrits dans différents repères en utilisant la matrice adjointe définie par :

$$Ad_{T_a^b} = \begin{pmatrix} R_a^b & 0 \\ \tilde{p}_a^b R_a^b & R_a^b \end{pmatrix} \quad (15)$$

Avec :

$$\Upsilon_j^{b,k} = Ad_{T_a^b} \Upsilon_j^{a,k} \quad (16)$$

.2.2 Torseur dynamique

Auparavant, les torseurs cinématique étaient définies, qui sont les généralisations des vitesses, linéaire et angulaire. Nous allons maintenant définir les torseurs dynamique, ce qui peut être vu comme une généralisation des forces et des moments. Le

torseur dynamique $W^{b,a} \in \mathbb{R}^6$ désigne un torseur agissant sur le repère \mathfrak{R}_a exprimée dans le repère \mathfrak{R}_b et est donnée par :

$$W^{b,a} = \begin{pmatrix} \tau^{b,a} & f^{b,a} \end{pmatrix} \quad (17)$$

où $\tau^{b,a} \in \mathbb{R}^3$ désigne le couple et $f^{b,a} \in \mathbb{R}^3$ la force, les deux agissant sur le repère \mathfrak{R}_a exprimé dans \mathfrak{R}_b . Notez que les torseurs dynamiques sont des covecteurs, elles sont donc représentées à l'aide de vecteurs de lignes. Le torseur dynamique peut être exprimé dans un repère différent en utilisant la relation :

$$(W^{b,j})^T = Ad_{H_b}^T (W^{a,j})^T \quad (18)$$

La matrice adjointe transposée est donnée par :

$$Ad_{H_b}^T = \begin{pmatrix} R_a^b & -R_a^b \tilde{D}_b^a \\ 0 & R_a^b \end{pmatrix} \quad (19)$$

3 Annexe C : Résultats de la Modélisation

Expression de l'Énergie Cinétique :

$$\begin{aligned}
E_c(q, \dot{q}) = & \frac{j_s \frac{\partial \alpha(t)^2}{\partial t^2}}{2} + \frac{j_s \frac{\partial \beta(t)^2}{\partial t^2}}{2} + \frac{j_s \frac{\partial \gamma(t)^2}{\partial t^2}}{2} + \frac{\frac{\partial \psi(t)^2}{\partial t^2} j_{b,xy}}{2} + \frac{\frac{\partial \theta(t)^2}{\partial t^2} j_{b,xy}}{2} + \frac{\frac{\partial \phi(t)^2}{\partial t^2} j_{b,z}}{2} + \frac{m_b \frac{\partial x_k(t)^2}{\partial t^2}}{2} + \frac{m_s \frac{\partial x_k(t)^2}{\partial t^2}}{2} + \\
& \frac{m_b \frac{\partial y_k(t)^2}{\partial t^2}}{2} + \frac{m_s \frac{\partial y_k(t)^2}{\partial t^2}}{2} + \frac{\cos(\theta(t))^2 \frac{\partial \phi(t)^2}{\partial t^2} j_{b,xy}}{2} - \frac{\cos(\theta(t))^2 \frac{\partial \phi(t)^2}{\partial t^2} j_{b,z}}{2} + \frac{l^2 m_b \frac{\partial \psi(t)^2}{\partial t^2}}{2} + \frac{l^2 m_b \frac{\partial \theta(t)^2}{\partial t^2}}{2} - \frac{\cos(\phi(t))^2 \cos(\theta(t))^2 \frac{\partial \psi(t)^2}{\partial t^2} j_{b,xy}}{2} + \\
& \frac{\cos(\phi(t))^2 \cos(\theta(t))^2 \frac{\partial \psi(t)^2}{\partial t^2} j_{b,z}}{2} - j_s \sin(\beta(t)) \frac{\partial \gamma(t)}{\partial t} \frac{\partial \alpha(t)}{\partial t} + \frac{l^2 m_b \cos(\theta(t))^2 \frac{\partial \phi(t)^2}{\partial t^2}}{2} - \sin(\phi(t)) \frac{\partial \theta(t)}{\partial t} j_{b,xy} \frac{\partial \psi(t)}{\partial t} - \\
& \frac{l^2 m_b \cos(\phi(t))^2 \cos(\theta(t))^2 \frac{\partial \psi(t)^2}{\partial t^2}}{2} - l^2 m_b \sin(\phi(t)) \frac{\partial \psi(t)}{\partial t} \frac{\partial \theta(t)}{\partial t} - l m_b \cos(\psi(t)) \cos(\theta(t)) \frac{\partial \theta(t)}{\partial t} \frac{\partial y_k(t)}{\partial t} + \\
& l m_b \sin(\theta(t)) \cos(\psi(t)) \frac{\partial \psi(t)}{\partial t} \frac{\partial x_k(t)}{\partial t} + l m_b \sin(\psi(t)) \cos(\theta(t)) \frac{\partial \theta(t)}{\partial t} \frac{\partial x_k(t)}{\partial t} + \\
& l m_b \sin(\psi(t)) \sin(\theta(t)) \frac{\partial \psi(t)}{\partial t} \frac{\partial y_k(t)}{\partial t} + \sin(\theta(t)) \cos(\phi(t)) \cos(\theta(t)) j_{b,xy} \frac{\partial \phi(t)}{\partial t} \frac{\partial \psi(t)}{\partial t} - \\
& \sin(\theta(t)) \cos(\phi(t)) \cos(\theta(t)) \frac{\partial \psi(t)}{\partial t} j_{b,z} \frac{\partial \phi(t)}{\partial t} + l m_b \cos(\phi(t)) \cos(\psi(t)) \cos(\theta(t)) \frac{\partial \phi(t)}{\partial t} \frac{\partial x_k(t)}{\partial t} + \\
& l m_b \sin(\psi(t)) \cos(\phi(t)) \cos(\theta(t)) \frac{\partial \phi(t)}{\partial t} \frac{\partial y_k(t)}{\partial t} + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)) \frac{\partial \psi(t)}{\partial t} \frac{\partial y_k(t)}{\partial t} + \\
& l^2 m_b \sin(\theta(t)) \cos(\phi(t)) \cos(\theta(t)) \frac{\partial \phi(t)}{\partial t} \frac{\partial \psi(t)}{\partial t} - l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t)) \frac{\partial x_k(t)}{\partial t} \frac{\partial \psi(t)}{\partial t} - \\
& l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t)) \frac{\partial \theta(t)}{\partial t} \frac{\partial x_k(t)}{\partial t} - l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)) \frac{\partial \theta(t)}{\partial t} \frac{\partial y_k(t)}{\partial t}
\end{aligned}$$

Les matrices du modèle non linéaire

a matrice d'inertie :

$$\bar{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} & M_{36} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ M_{61} & M_{62} & M_{63} & M_{64} & M_{65} & M_{66} \end{pmatrix}$$

$$M_{11} = j_s + m_b r_s^2 + m_s r_s^2 - m_b r_s^2 \sin(\beta(t))^2 - m_s r_s^2 \sin(\beta(t))^2$$

$$M_{13} = -j_s \sin(\beta(t))$$

$$\begin{aligned}
M_{14} = & r_s \cos(\beta(t)) \cos(\gamma(t)) (l m_b \cos(\psi(t)) \cos(\theta(t)) + l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t))) \\
& + r_s \sin(\gamma(t)) \cos(\beta(t)) (l m_b \sin(\psi(t)) \cos(\theta(t)) - l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t)))
\end{aligned}$$

$$\begin{aligned}
M_{16} = & r_s \sin(\gamma(t)) \cos(\beta(t)) (l m_b \sin(\theta(t)) \cos(\psi(t)) - l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \\
& - r_s \cos(\beta(t)) \cos(\gamma(t)) (l m_b \sin(\psi(t)) \sin(\theta(t)) + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)))
\end{aligned}$$

$$M_{22} = j_s + m_b r_s^2 + m_s r_s^2$$

$$\begin{aligned}
M_{24} = & r_s \cos(\gamma(t)) (l m_b \sin(\psi(t)) \cos(\theta(t)) - l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t))) \\
& - r_s \sin(\gamma(t)) (l m_b \cos(\psi(t)) \cos(\theta(t)) + l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)))
\end{aligned}$$

$$M_{25} = l m_b r_s \cos(\gamma(t) - \psi(t)) \cos(\phi(t)) \cos(\theta(t))$$

$$\begin{aligned}
M_{26} = & r_s \cos(\gamma(t)) (l m_b \sin(\theta(t)) \cos(\psi(t)) - l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \\
& + r_s \sin(\gamma(t)) (l m_b \sin(\psi(t)) \sin(\theta(t)) + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)))
\end{aligned}$$

$$M_{31} = -j_s \sin(\beta(t))$$

$$M_{33} = j_s$$

$$M_{41} = r_s \cos(\beta(t)) \cos(\gamma(t)) (l m_b \cos(\psi(t)) \cos(\theta(t)) + l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t))) \\ + r_s \sin(\gamma(t)) \cos(\beta(t)) (l m_b \sin(\psi(t)) \cos(\theta(t)) - l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t)))$$

$$M_{42} = r_s \cos(\gamma(t)) (l m_b \sin(\psi(t)) \cos(\theta(t)) - l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t))) \\ - r_s \sin(\gamma(t)) (l m_b \cos(\psi(t)) \cos(\theta(t)) + l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)))$$

$$M_{44} = m_b l^2 + j_{xy}$$

$$M_{46} = -\sin(\phi(t)) (m_b l^2 + j_{xy})$$

$$M_{51} = l m_b r_s \sin(\gamma(t) - \psi(t)) \cos(\beta(t)) \cos(\phi(t)) \cos(\theta(t))$$

$$M_{52} = l m_b r_s \sin(\gamma(t) - \psi(t)) \cos(\beta(t)) \cos(\phi(t)) \cos(\theta(t))$$

$$M_{55} = j_z - j_z \cos(\theta(t))^2 + \cos(\theta(t))^2 j_{xy} + l^2 m_b \cos(\theta(t))^2$$

$$M_{56} = \sin(\theta(t)) \cos(\phi(t)) \cos(\theta(t)) (m_b l^2 - j_z + j_{xy})$$

$$M_{61} = r_s \sin(\gamma(t)) \cos(\beta(t)) (l m_b \sin(\theta(t)) \cos(\psi(t)) - l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \\ - r_s \cos(\beta(t)) \cos(\gamma(t)) (l m_b \sin(\psi(t)) \sin(\theta(t)) + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)))$$

$$M_{62} = r_s \cos(\gamma(t)) (l m_b \sin(\theta(t)) \cos(\psi(t)) - l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \\ + r_s \sin(\gamma(t)) (l m_b \sin(\psi(t)) \sin(\theta(t)) + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)))$$

$$M_{64} = -\sin(\phi(t)) (m_b l^2 + j_{xy})$$

$$M_{65} = \sin(\theta(t)) \cos(\phi(t)) \cos(\theta(t)) (m_b l^2 - j_z + j_{xy})$$

$$M_{66} = j_{xy} + l^2 m_b - \cos(\phi(t))^2 \cos(\theta(t))^2 j_{xy} + j_z \cos(\phi(t))^2 \cos(\theta(t))^2 - l^2 m_b \cos(\phi(t))^2 \cos(\theta(t))^2$$

Le reste des valeurs est nulles.

La matrice de Coriolis :

$$\bar{C} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{pmatrix}$$

Avec :

$$C_{11} = -\frac{r_s^2 \sin(2\beta(t)) (m_b + m_s) \frac{\partial}{\partial t} \beta(t)}{2}$$

$$C_{12} = -\cos(\beta(t)) \frac{\partial}{\partial t} \gamma(t) (j_s + m_b r_s^2 + m_s r_s^2)$$

$$\begin{aligned} C_{14} = & r_s \cos(\beta(t)) \cos(\gamma(t)) (2l m_b \sin(\psi(t)) \sin(\theta(t)) \cos(\phi(t)) \frac{\partial}{\partial t} \phi(t) \\ & - l m_b \sin(\theta(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \theta(t) - 2l m_b \sin(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \psi(t) \\ & + 2l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \psi(t) + l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \theta(t) \\ & - r_s \sin(\gamma(t)) \cos(\beta(t)) (l m_b \sin(\psi(t)) \sin(\theta(t)) \frac{\partial}{\partial t} \theta(t) - 2l m_b \cos(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \psi(t) \\ & + 2l m_b \sin(\theta(t)) \cos(\phi(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \phi(t) + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \theta(t) \\ & - 2l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)) \frac{\partial}{\partial t} \psi(t)) \end{aligned}$$

$$\begin{aligned} C_{15} = & -l m_b r_s \cos(\beta(t)) \cos(\gamma(t)) \cos(\theta(t)) (2 \cos(\phi(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \psi(t) \\ & - \sin(\phi(t)) \sin(\psi(t)) \frac{\partial}{\partial t} \phi(t) - l m_b r_s \sin(\gamma(t)) \cos(\beta(t)) \cos(\theta(t)) \\ & \left(\sin(\phi(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \phi(t) + 2 \sin(\psi(t)) \cos(\phi(t)) \frac{\partial}{\partial t} \psi(t) \right) \end{aligned}$$

$$\begin{aligned} C_{16} = & -l m_b r_s \cos(\beta(t)) \cos(\gamma(t)) (\sin(\theta(t)) \cos(\psi(t)) - \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \frac{\partial}{\partial t} \psi(t) \\ & - l m_b r_s \sin(\gamma(t)) \cos(\beta(t)) (\sin(\psi(t)) \sin(\theta(t)) + \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t))) \frac{\partial}{\partial t} \psi(t) \end{aligned}$$

$$C_{21} = \cos(\beta(t)) \frac{\partial}{\partial t} \gamma(t) (j_s + m_b r_s^2 + m_s r_s^2)$$

$$\begin{aligned} C_{24} = & -r_s \cos(\gamma(t)) (l m_b \sin(\psi(t)) \sin(\theta(t)) \frac{\partial}{\partial t} \theta(t) - 2l m_b \cos(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \psi(t) \\ & + 2l m_b \sin(\theta(t)) \cos(\phi(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \phi(t) + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \theta(t) \\ & - 2l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)) \frac{\partial}{\partial t} \psi(t) - r_s \sin(\gamma(t)) (2l m_b \sin(\psi(t)) \sin(\theta(t)) \cos(\phi(t)) \frac{\partial}{\partial t} \phi(t) \\ & - l m_b \sin(\theta(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \theta(t) - 2l m_b \sin(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \psi(t) + 2l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t)) \\ & + l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \theta(t)) \end{aligned}$$

$$\begin{aligned} C_{25} = & l m_b r_s \sin(\gamma(t)) \cos(\theta(t)) \left(2 \cos(\phi(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \psi(t) - \sin(\phi(t)) \sin(\psi(t)) \frac{\partial}{\partial t} \phi(t) \right) \\ & - l m_b r_s \cos(\gamma(t)) \cos(\theta(t)) \left(\sin(\phi(t)) \cos(\psi(t)) \frac{\partial}{\partial t} \phi(t) + 2 \sin(\psi(t)) \cos(\phi(t)) \frac{\partial}{\partial t} \psi(t) \right) \end{aligned}$$

$$\begin{aligned} C_{26} = & l m_b r_s \sin(\gamma(t)) (\sin(\theta(t)) \cos(\psi(t)) - \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \frac{\partial}{\partial t} \psi(t) \\ & - l m_b r_s \cos(\gamma(t)) (\sin(\psi(t)) \sin(\theta(t)) + \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t))) \frac{\partial}{\partial t} \psi(t) \end{aligned}$$

$$C_{31} = -j_s \cos(\beta(t)) \frac{\partial}{\partial t} \beta(t)$$

$$\begin{aligned}
C_{41} &= (l m_b \sin(\psi(t)) \cos(\theta(t)) - l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t))) (r_s \cos(\beta(t)) \cos(\gamma(t)) \frac{\partial}{\partial t} \gamma(t) \\
&\quad - r_s \sin(\beta(t)) \sin(\gamma(t)) \frac{\partial}{\partial t} \beta(t)) - (l m_b \cos(\psi(t)) \cos(\theta(t)) + l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t))) \\
&\quad \left(r_s \sin(\beta(t)) \cos(\gamma(t)) \frac{\partial}{\partial t} \beta(t) + r_s \sin(\gamma(t)) \cos(\beta(t)) \frac{\partial}{\partial t} \gamma(t) \right) \\
C_{42} &= -r_s \cos(\gamma(t)) (l m_b \cos(\psi(t)) \cos(\theta(t)) + l m_b \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t))) \frac{\partial}{\partial t} \gamma(t) \\
&\quad - r_s \sin(\gamma(t)) (l m_b \sin(\psi(t)) \cos(\theta(t)) - l m_b \sin(\phi(t)) \sin(\theta(t)) \cos(\psi(t))) \frac{\partial}{\partial t} \gamma(t) \\
C_{44} &= -l m_b (\sin(\psi(t)) \sin(\theta(t)) \frac{\partial}{\partial t} x_k(t) - \sin(\theta(t)) \cos(\psi(t)) \frac{\partial}{\partial t} y_k(t) \\
&\quad + \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} x_k(t) + \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} y_k(t)) \\
C_{45} &= \sin(2\theta(t)) \left(\frac{m_b l^2}{2} - \frac{j_z}{2} + \frac{j_{xy}}{2} \right) \frac{\partial}{\partial t} \phi(t) - j_z \cos(\phi(t)) \frac{\partial}{\partial t} \psi(t) - 2 \cos(\phi(t)) \cos(\theta(t))^2 j_{xy} \frac{\partial}{\partial t} \psi(t) \\
&\quad + 2 j_z \cos(\phi(t)) \cos(\theta(t))^2 \frac{\partial}{\partial t} \psi(t) - 2 l^2 m_b \cos(\phi(t)) \cos(\theta(t))^2 \frac{\partial}{\partial t} \psi(t) \\
C_{46} &= -\sin(\theta(t)) \cos(\phi(t))^2 \cos(\theta(t)) \frac{\partial}{\partial t} \psi(t) (m_b l^2 - j_z + j_{xy}) \\
C_{51} &= m_b \cos(\phi(t)) \cos(\psi(t)) \cos(\theta(t)) \left(r_s \cos(\beta(t)) \cos(\gamma(t)) \frac{\partial}{\partial t} \gamma(t) - r_s \sin(\beta(t)) \sin(\gamma(t)) \frac{\partial}{\partial t} \beta(t) \right) \\
&\quad + l m_b \sin(\psi(t)) \cos(\phi(t)) \cos(\theta(t)) (r_s \sin(\beta(t)) \cos(\gamma(t)) \frac{\partial}{\partial t} \beta(t) + r_s \sin(\gamma(t)) \cos(\beta(t)) \frac{\partial}{\partial t} \gamma(t)) \\
C_{52} &= -l m_b r_s \sin(\gamma(t) - \psi(t)) \cos(\phi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \gamma(t) \\
C_{54} &= j_z \cos(\phi(t)) \frac{\partial}{\partial t} \psi(t) - \sin(2\theta(t)) j_{xy} \frac{\partial}{\partial t} \phi(t) + j_z \sin(2\theta(t)) \frac{\partial}{\partial t} \phi(t) + 2 \cos(\phi(t)) \cos(\theta(t))^2 j_{xy} \frac{\partial}{\partial t} \psi(t) \\
&\quad - l^2 m_b \sin(2\theta(t)) \frac{\partial}{\partial t} \phi(t) - 2 j_z \cos(\phi(t)) \cos(\theta(t))^2 \frac{\partial}{\partial t} \psi(t) + 2 l^2 m_b \cos(\phi(t)) \cos(\theta(t))^2 \frac{\partial}{\partial t} \psi(t) \\
C_{55} &= -\sin(\phi(t)) \cos(\theta(t)) (\sin(\theta(t)) j_{xy} \frac{\partial}{\partial t} \psi(t) - j_z \sin(\theta(t)) \frac{\partial}{\partial t} \psi(t) \\
&\quad + l m_b \sin(\psi(t)) \frac{\partial}{\partial t} y_k(t) + l^2 m_b \sin(\theta(t)) \frac{\partial}{\partial t} \psi(t) + l m_b \cos(\psi(t)) \frac{\partial}{\partial t} x_k(t)) \\
C_{56} &= -\sin(\phi(t)) \cos(\phi(t)) \cos(\theta(t))^2 \frac{\partial}{\partial t} \psi(t) (m_b l^2 - j_z + j_{xy}) \\
C_{61} &= (l m_b \sin(\psi(t)) \sin(\theta(t)) + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t))) (r_s \sin(\beta(t)) \cos(\gamma(t)) \frac{\partial}{\partial t} \beta(t) \\
&\quad + r_s \sin(\gamma(t)) \cos(\beta(t)) \frac{\partial}{\partial t} \gamma(t)) + (l m_b \sin(\theta(t)) \cos(\psi(t)) - l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \\
&\quad \left(r_s \cos(\beta(t)) \cos(\gamma(t)) \frac{\partial}{\partial t} \gamma(t) - r_s \sin(\beta(t)) \sin(\gamma(t)) \frac{\partial}{\partial t} \beta(t) \right) \\
C_{62} &= r_s \cos(\gamma(t)) (l m_b \sin(\psi(t)) \sin(\theta(t)) + l m_b \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t))) \frac{\partial}{\partial t} \gamma(t) \\
&\quad - r_s \sin(\gamma(t)) (l m_b \sin(\theta(t)) \cos(\psi(t)) - l m_b \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t))) \frac{\partial}{\partial t} \gamma(t)
\end{aligned}$$

$$\begin{aligned}
C_{64} = & -\cos(\phi(t)) \left(2j_{xy} \frac{\partial}{\partial t} \phi(t) - j_z \frac{\partial}{\partial t} \phi(t) + 2l^2 m_b \frac{\partial}{\partial t} \phi(t) - 2\cos(\theta(t))^2 j_{xy} \frac{\partial}{\partial t} \phi(t) \right. \\
& + 2j_z \cos(\theta(t))^2 \frac{\partial}{\partial t} \phi(t) - 2l^2 m_b \cos(\theta(t))^2 \frac{\partial}{\partial t} \phi(t) - 2\sin(\theta(t)) \cos(\phi(t)) \cos(\theta(t)) j_{xy} \frac{\partial}{\partial t} \psi(t) \\
& \left. + 2j_z \sin(\theta(t)) \cos(\phi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \psi(t) \right)
\end{aligned}$$

$$C_{65} = -\sin(\phi(t)) \cos(\theta(t)) \left(\sin(\theta(t)) \frac{\partial}{\partial t} \phi(t) - 2\cos(\phi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} \psi(t) \right) (m_b l^2 - j_z + j_{xy})$$

$$\begin{aligned}
C_{66} = & -l m_b \left(\sin(\psi(t)) \sin(\theta(t)) \frac{\partial}{\partial t} x_k(t) - \sin(\theta(t)) \cos(\psi(t)) \frac{\partial}{\partial t} y_k(t) + \sin(\phi(t)) \cos(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} x_k(t) \right. \\
& \left. + \sin(\phi(t)) \sin(\psi(t)) \cos(\theta(t)) \frac{\partial}{\partial t} y_k(t) - 2l \sin(\theta(t)) \cos(\phi(t))^2 \cos(\theta(t)) \frac{\partial}{\partial t} \theta(t) \right)
\end{aligned}$$

Le reste des valeurs sont nulles.

La matrice de gravité :

$$\bar{G} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -g l m_b \sin(\theta(t)) \cos(\phi(t)) \\ -g l m_b \sin(\phi(t)) \cos(\theta(t)) \\ 0 \end{pmatrix}$$

La matrice \bar{B} :

$$\bar{B} = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \\ B_{41} & B_{42} & B_{43} \\ B_{51} & B_{52} & B_{53} \\ B_{61} & B_{62} & B_{63} \end{pmatrix}$$

$$\begin{aligned}
B_{11} = & - (r_s \sin(\beta(t)) \sin(\phi(t)) \cos(\delta) + r_s \sin(\beta(t)) \sin(\delta) \cos(\phi(t)) \cos(\theta(t)) \\
& + r_s \cos(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\phi(t)) \cos(\psi(t)) + r_s \sin(\gamma(t)) \sin(\psi(t)) \cos(\delta) \cos(\beta(t)) \cos(\phi(t)) \\
& + r_s \sin(\gamma(t)) \sin(\theta(t)) \sin(\delta) \cos(\beta(t)) \cos(\psi(t)) - r_s \sin(\psi(t)) \sin(\theta(t)) \sin(\delta) \cos(\beta(t)) \cos(\gamma(t)) \\
& - r_s \sin(\phi(t)) \sin(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\psi(t)) \cos(\theta(t)) \\
& - r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\psi(t)) \sin(\delta) \cos(\beta(t)) \cos(\theta(t)) / (r_w)
\end{aligned}$$

$$\begin{aligned}
B_{12} = & (r_s \sin(\beta(t)) \sin(\phi(t)) \cos(\delta) - 2r_s \sin(\beta(t)) \sin(\delta) \cos(\phi(t)) \cos(\theta(t)) \\
& + r_s \cos(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\phi(t)) \cos(\psi(t)) + r_s \sin(\gamma(t)) \sin(\psi(t)) \cos(\delta) \cos(\beta(t)) \cos(\phi(t)) \\
& - 2r_s \sin(\gamma(t)) \sin(\theta(t)) \sin(\delta) \cos(\beta(t)) \cos(\psi(t)) + 2r_s \sin(\psi(t)) \sin(\theta(t)) \sin(\delta) \cos(\beta(t)) \cos(\gamma(t)) \\
& + \sqrt{3}r_s \sin(\beta(t)) \sin(\theta(t)) \cos(\delta) \cos(\phi(t)) + 2r_s \sin(\phi(t)) \sin(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\psi(t)) \cos(\theta(t)) \\
& + 2r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\psi(t)) \sin(\delta) \cos(\beta(t)) \cos(\theta(t)) \\
& - \sqrt{3}r_s \sin(\gamma(t)) \cos(\delta) \cos(\beta(t)) \cos(\psi(t)) \cos(\theta(t)) + \sqrt{3}r_s \sin(\psi(t)) \cos(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\theta(t)) \\
& - \sqrt{3}r_s \sin(\phi(t)) \sin(\theta(t)) \cos(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\psi(t)) \\
& - \sqrt{3}r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)) \cos(\delta) \cos(\beta(t)))/(2r_w)
\end{aligned}$$

$$\begin{aligned}
B_{13} = & (r_s \sin(\beta(t)) \sin(\phi(t)) \cos(\delta) - 2r_s \sin(\beta(t)) \sin(\delta) \cos(\phi(t)) \cos(\theta(t)) \\
& + r_s \cos(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\phi(t)) \cos(\psi(t)) + r_s \sin(\gamma(t)) \sin(\psi(t)) \cos(\delta) \cos(\beta(t)) \cos(\phi(t)) \\
& - 2r_s \sin(\gamma(t)) \sin(\theta(t)) \sin(\delta) \cos(\beta(t)) \cos(\psi(t)) + 2r_s \sin(\psi(t)) \sin(\theta(t)) \sin(\delta) \cos(\beta(t)) \cos(\gamma(t)) \\
& - \sqrt{3}r_s \sin(\beta(t)) \sin(\theta(t)) \cos(\delta) \cos(\phi(t)) + 2r_s \sin(\phi(t)) \sin(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\psi(t)) \cos(\theta(t)) \\
& + 2r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\psi(t)) \sin(\delta) \cos(\beta(t)) \cos(\theta(t)) \\
& + \sqrt{3}r_s \sin(\gamma(t)) \cos(\delta) \cos(\beta(t)) \cos(\psi(t)) \cos(\theta(t)) - \sqrt{3}r_s \sin(\psi(t)) \cos(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\theta(t)) \\
& + \sqrt{3}r_s \sin(\phi(t)) \sin(\theta(t)) \cos(\delta) \cos(\beta(t)) \cos(\gamma(t)) \cos(\psi(t)) \\
& + \sqrt{3}r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)) \cos(\delta) \cos(\beta(t)))/(2r_w)
\end{aligned}$$

$$\begin{aligned}
B_{21} = & - (r_s \sin(\psi(t)) \cos(\delta) \cos(\gamma(t)) \cos(\phi(t)) - r_s \sin(\gamma(t)) \cos(\delta) \cos(\phi(t)) \cos(\psi(t)) \\
& + r_s \sin(\theta(t)) \sin(\delta) \cos(\gamma(t)) \cos(\psi(t)) + r_s \sin(\gamma(t)) \sin(\psi(t)) \sin(\theta(t)) \sin(\delta) \\
& + r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\delta) \cos(\psi(t)) \cos(\theta(t)) - r_s \sin(\phi(t)) \sin(\psi(t)) \sin(\delta) \cos(\gamma(t)) \cos(\theta(t)))/(r_w)
\end{aligned}$$

$$\begin{aligned}
B_{22} = & - (r_s \sin(\gamma(t)) \cos(\delta) \cos(\phi(t)) \cos(\psi(t)) - r_s \sin(\psi(t)) \cos(\delta) \cos(\gamma(t)) \cos(\phi(t)) \\
& + 2r_s \sin(\theta(t)) \sin(\delta) \cos(\gamma(t)) \cos(\psi(t)) + 2r_s \sin(\gamma(t)) \sin(\psi(t)) \sin(\theta(t)) \sin(\delta) \\
& + 2r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\delta) \cos(\psi(t)) \cos(\theta(t)) - 2r_s \sin(\phi(t)) \sin(\psi(t)) \sin(\delta) \cos(\gamma(t)) \cos(\theta(t)) \\
& + \sqrt{3}r_s \cos(\delta) \cos(\gamma(t)) \cos(\psi(t)) \cos(\theta(t)) + \sqrt{3}r_s \sin(\gamma(t)) \sin(\psi(t)) \cos(\delta) \cos(\theta(t)) \\
& - \sqrt{3}r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\theta(t)) \cos(\delta) \cos(\psi(t)) \\
& + \sqrt{3}r_s \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)) \cos(\delta) \cos(\gamma(t)))/(2r_w)
\end{aligned}$$

$$\begin{aligned}
B_{23} = & - (r_s \sin(\gamma(t)) \cos(\delta) \cos(\phi(t)) \cos(\psi(t)) - r_s \sin(\psi(t)) \cos(\delta) \cos(\gamma(t)) \cos(\phi(t)) \\
& + 2 r_s \sin(\theta(t)) \sin(\delta) \cos(\gamma(t)) \cos(\psi(t)) + 2 r_s \sin(\gamma(t)) \sin(\psi(t)) \sin(\theta(t)) \sin(\delta) \\
& + 2 r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\delta) \cos(\psi(t)) \cos(\theta(t)) - 2 r_s \sin(\phi(t)) \sin(\psi(t)) \sin(\delta) \cos(\gamma(t)) \cos(\theta(t)) \\
& - \sqrt{3} r_s \cos(\delta) \cos(\gamma(t)) \cos(\psi(t)) \cos(\theta(t)) - \sqrt{3} r_s \sin(\gamma(t)) \sin(\psi(t)) \cos(\delta) \cos(\theta(t)) \\
& + \sqrt{3} r_s \sin(\gamma(t)) \sin(\phi(t)) \sin(\theta(t)) \cos(\delta) \cos(\psi(t)) \\
& - \sqrt{3} r_s \sin(\phi(t)) \sin(\psi(t)) \sin(\theta(t)) \cos(\delta) \cos(\gamma(t)) / (2 r_w)
\end{aligned}$$

$$B_{31} = \frac{r_s (\sin(\phi(t)) \cos(\delta) + \sin(\delta) \cos(\phi(t)) \cos(\theta(t)))}{r_w}$$

$$B_{32} = - \frac{r_s (\sin(\phi(t)) \cos(\delta) - 2 \sin(\delta) \cos(\phi(t)) \cos(\theta(t)))}{2 r_w} - \frac{\sqrt{3} r_s \sin(\theta(t)) \cos(\delta) \cos(\phi(t))}{2 r_w}$$

$$B_{33} = \frac{2 r_s \sin(\delta) \cos(\phi(t)) \cos(\theta(t)) - r_s \sin(\phi(t)) \cos(\delta) + \sqrt{3} r_s \sin(\theta(t)) \cos(\delta) \cos(\phi(t))}{2 r_w}$$

$$B_{41} = \frac{r_s \cos(\delta)}{r_w}$$

$$B_{42} = - \frac{r_s \cos(\delta)}{2 r_w}$$

$$B_{43} = - \frac{r_s \cos(\delta)}{2 r_w}$$

$$B_{51} = \frac{r_s \sin(\theta(t)) \sin(\delta)}{r_w}$$

$$B_{52} = \frac{r_s \sin(\theta(t)) \sin(\delta)}{r_w} + \frac{\sqrt{3} r_s \cos(\delta) \cos(\theta(t))}{2 r_w}$$

$$B_{53} = \frac{r_s \sin(\theta(t)) \sin(\delta)}{r_w} - \frac{\sqrt{3} r_s \cos(\delta) \cos(\theta(t))}{2 r_w}$$

$$B_{61} = - \frac{r_s \sin(\phi(t)) \cos(\delta)}{r_w} - \frac{r_s \sin(\delta) \cos(\phi(t)) \cos(\theta(t))}{r_w}$$

$$B_{62} = \frac{r_s \sin(\phi(t)) \cos(\delta)}{2 r_w} - \frac{r_s \sin(\delta) \cos(\phi(t)) \cos(\theta(t))}{r_w} + \frac{\sqrt{3} r_s \sin(\theta(t)) \cos(\delta) \cos(\phi(t))}{2 r_w}$$

$$B_{63} = \frac{r_s \sin(\phi(t)) \cos(\delta)}{2 r_w} - \frac{r_s \sin(\delta) \cos(\phi(t)) \cos(\theta(t))}{r_w} - \frac{\sqrt{3} r_s \sin(\theta(t)) \cos(\delta) \cos(\phi(t))}{2 r_w}$$

Les matrices du modèle linéaire

$$A_l = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Avec :

$$\begin{aligned} A_1 = & (g l m_b^2 r_s^2) / (j_s j_{xy} + j_s l^2 m_b + m_b r_s^2 j_{xy} + m_s r_s^2 j_{xy} + l^2 m_b m_s r_s^2) \\ & + (g j_s l m_b) / (j_s j_{xy} + j_s l^2 m_b + m_b r_s^2 j_{xy} + m_s r_s^2 j_{xy} + l^2 m_b m_s r_s^2) \\ & + (g l m_b m_s r_s^2) / (j_s j_{xy} + j_s l^2 m_b + m_b r_s^2 j_{xy} + m_s r_s^2 j_{xy} + l^2 m_b m_s r_s^2) \end{aligned}$$

$$A_2 = (g l^2 m_b^2 r_s^2) / (j_s j_{xy} + j_s l^2 m_b + m_b r_s^2 j_{xy} + m_s r_s^2 j_{xy} + l^2 m_b m_s r_s^2)$$

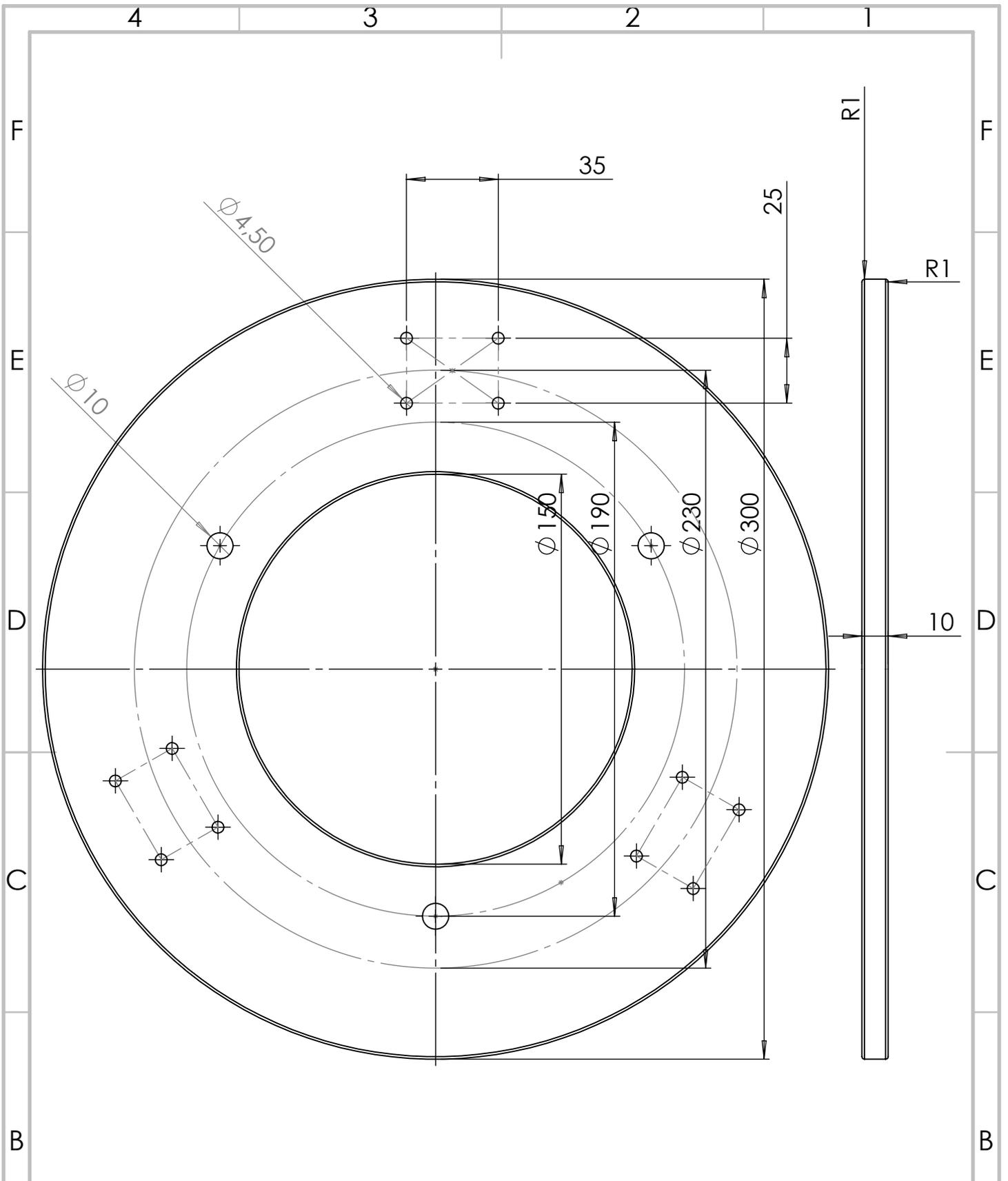
$$B_l = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_1 & b_2 & b_2 \\ 0 & b_3 & -b_3 \\ b_4 & b_4 & b_4 \\ 0 & -b_5 & b_5 \\ b_6 & b_7 & b_7 \end{pmatrix}$$

Avec :

$$\begin{aligned} b_1 = & (j_s r_s \cos(\delta)) / (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w) \\ & + (m_b r_s^3 \cos(\delta)) / (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w) \\ & + (m_s r_s^3 \cos(\delta)) / (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w) \\ & + (l m_b r_s^2 \cos(\delta)) / (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w) \end{aligned}$$

$$\begin{aligned}
b_2 = & - (j_s r_s \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& - (m_b r_s^3 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& - (m_s r_s^3 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& - (l m_b r_s^2 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
b_3 = & (\sqrt{3} m_b r_s^3 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& + (\sqrt{3} m_s r_s^3 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& + (\sqrt{3} j_s r_s \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& + (\sqrt{3} l m_b r_s^2 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
b_4 = & -r_s \sin(\delta) / (j_z r_w) \\
b_5 = & (\sqrt{3} r_s^2 \cos(\delta) j_{xy}) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& + (\sqrt{3} l m_b r_s^3 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& + (\sqrt{3} l^2 m_b r_s^2 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
b_6 = & (r_s^2 \cos(\delta) j_{xy}) / (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w) \\
& + (l m_b r_s^3 \cos(\delta)) / (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w) \\
& + (l^2 m_b r_s^2 \cos(\delta)) / (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w) \\
b_7 = & - (r_s^2 \cos(\delta) j_{xy}) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& - (l m_b r_s^3 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w)) \\
& - (l^2 m_b r_s^2 \cos(\delta)) / (2 (j_s r_w j_{xy} + m_b r_s^2 r_w j_{xy} + m_s r_s^2 r_w j_{xy} + j_s l^2 m_b r_w + l^2 m_b m_s r_s^2 r_w))
\end{aligned}$$

.4 Annexe D :Les plans 2D des pièces utilisées dans la conception



SAUF INDICATION CONTRAIRE:
 LES COTES SONT EN MILLIMETRES
 ETAT DE SURFACE:
 TOLERANCES:
 LINEAIRES:
 ANGULAIRES:

FINITION:

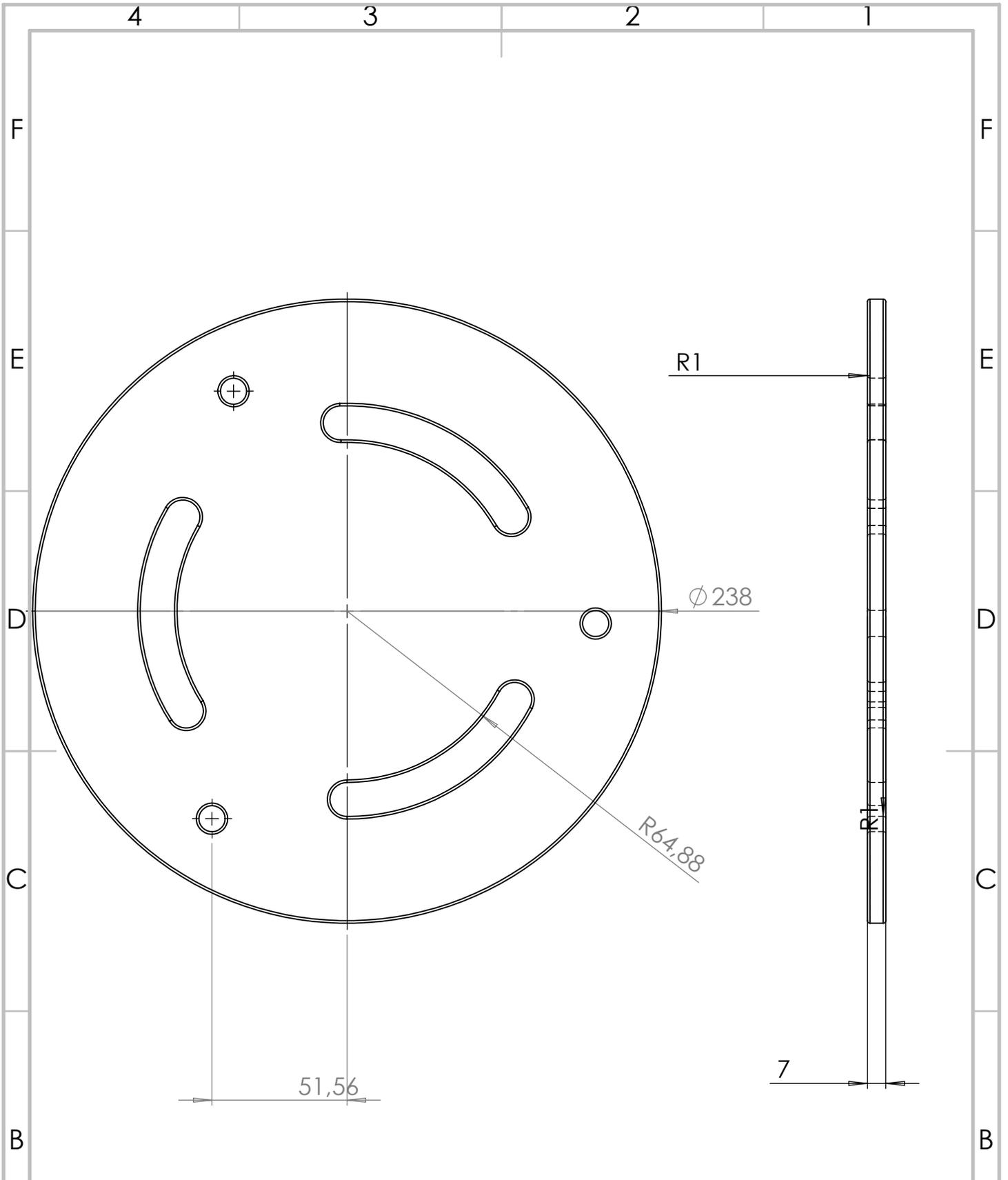
CASSER LES
 ANGLES VIFS

NE PAS CHANGER L'ECHELLE

REVISION

	NOM	SIGNATURE	DATE	
AUTEUR				
VERIF.				
APPR.				
FAB.				
QUAL.				
				MATERIAU:
				MASSE:

TITRE:	
No. DE PLAN	
Base en bas	A4
ECHELLE:1:2	FEUILLE 1 SUR 1



SAUF INDICATION CONTRAIRE:
 LES COTES SONT EN MILLIMETRES
 ETAT DE SURFACE:
 TOLERANCES:
 LINEAIRES:
 ANGULAIRES:

FINITION:

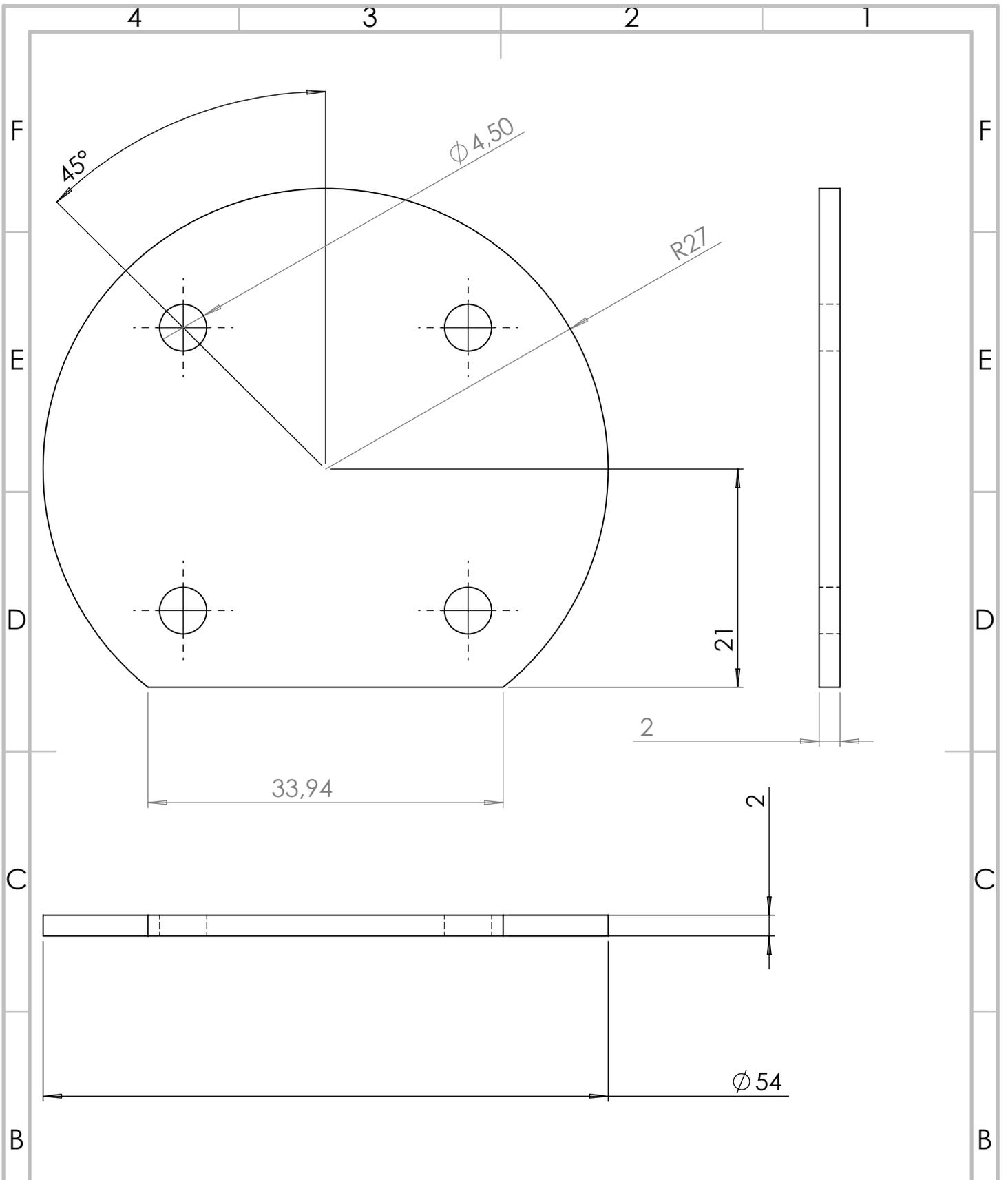
CASSER LES
 ANGLES VIFS

NE PAS CHANGER L'ECHELLE

REVISION

	NOM	SIGNATURE	DATE		
AUTEUR					
VERIF.					
APPR.					
FAB.					
QUAL.					

TITRE:	
No. DE PLAN	
Etagière	A4
ECHELLE:1:2	FEUILLE 1 SUR 1



SAUF INDICATION CONTRAIRE:
 LES COTES SONT EN MILLIMETRES
 ETAT DE SURFACE:
 TOLERANCES:
 LINEAIRES:
 ANGULAIRES:

FINITION:

CASSER LES
 ANGLES VIFS

NE PAS CHANGER L'ECHELLE

REVISION

	NOM	SIGNATURE	DATE		
AUTEUR					
VERIF.					
APPR.					
FAB.					
QUAL.					

TITRE:

MATERIAU:

No. DE PLAN

Pièce_2

A4

MASSE:

ECHELLE:2:1

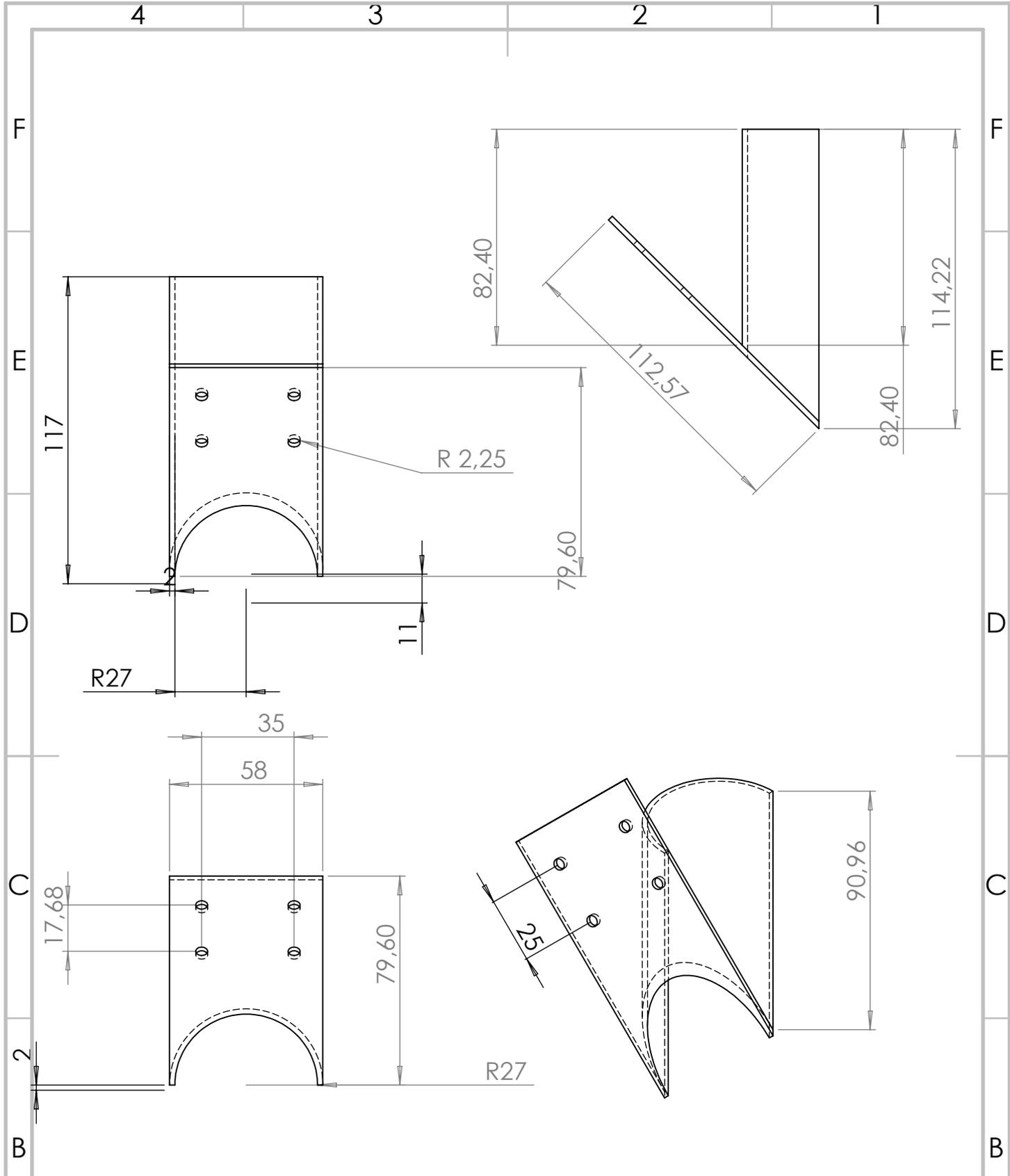
FEUILLE 1 SUR 1

4

3

2

1



SAUF INDICATION CONTRAIRE:
LES COTES SONT EN MILLIMETRES
ETAT DE SURFACE:
TOLERANCES:
LINEAIRES:
ANGULAIRES:

FINITION:

CASSER LES
ANGLES VIFS

NE PAS CHANGER L'ECHELLE

REVISION

NOM	SIGNATURE	DATE			
AUTEUR					
VERIF.					
APPR.					
FAB.					
QUAL.					

TITRE:

MATERIAU:

No. DE PLAN

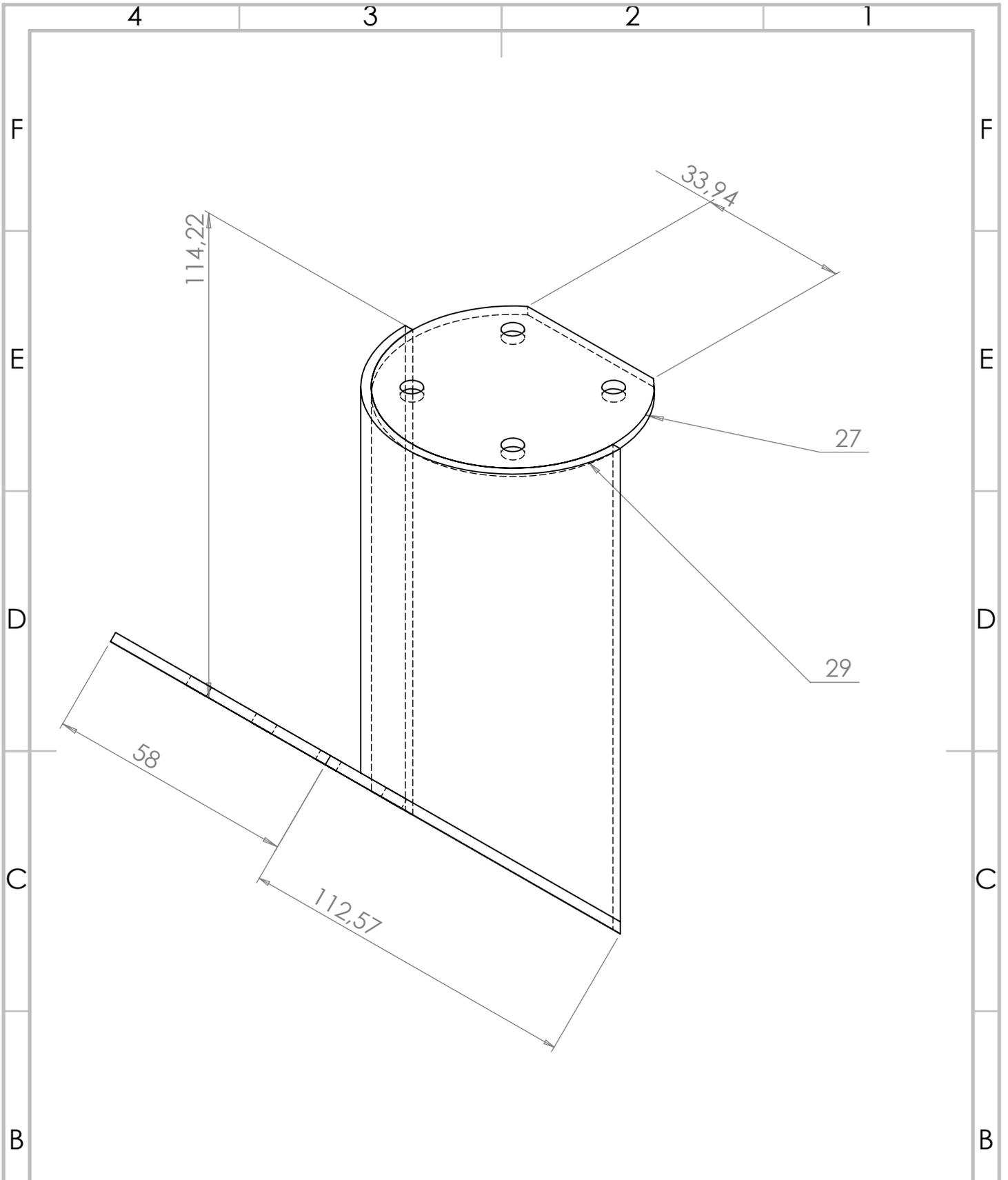
Pièce_1

A4

MASSE:

ECHELLE:1:2

FEUILLE 1 SUR 1



SAUF INDICATION CONTRAIRE:
 LES COTES SONT EN MILLIMETRES
 ETAT DE SURFACE:
 TOLERANCES:
 LINEAIRES:
 ANGULAIRES:

FINITION:

CASSER LES
 ANGLES VIFS

NE PAS CHANGER L'ECHELLE

REVISION

	NOM	SIGNATURE	DATE		
AUTEUR					
VERIF.					
APPR.					
FAB.					
QUAL.					

TITRE:

No. DE PLAN

Support compèt A4

ECHELLE:1:2

FEUILLE 1 SUR 1