

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--T L E M C E N--



المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique
Spécialité : Automatique

Présenté par : KHELIL Ghizlène
YADI Khalil

Thème

**Simulation et commande d'une main
robotisée**

Soutenu publiquement, le 02/07/2023, devant le jury composé de :

M. L.Merad	Professeur	ESSA. Tlemcen	Président
M. B.Charki	Professeur	ESSA. Tlemcen	Directeur de mémoire
M. SM.Abdi	MCB	ESSA. Tlemcen	Examineur 1
Mme A.Ghouali	MCA	ESSA. Tlemcen	Examineur 2
M. R.A.Adjim	Ingénieur de laboratoire	ESSA. Tlemcen	Invité 1

Année universitaire : 2022 /2023

DÉDICACE

Je me permets d'écrire ces quelques mots pour exprimer mon sentiment en cette fin de parcours académique dont je suis très fier. Ça a été de loin les années les plus enrichissantes de ma vie, mais je pense particulièrement à cette année et à ce projet de fin d'études que je suis heureux de dédier aux personnes les plus importantes de mon existence :

À mon binôme, ma cousine et mon amie de toujours **Ghizlène KHELIL** avec qui je partage ce travail sans qui, ce projet n'aurait jamais vu le jour. Je voudrais te remercier pour ton travail, ta présence, ton soutien et ta bonne humeur. Je suis très content d'avoir partagé cette aventure avec toi.

Je dédie ce travail à mes parents, pour leurs soutien inébranlable et pour leur amour inconditionnel, et à qui je dois tout simplement la vie. Je vous remercie du fond du coeur pour tous ces sacrifices ainsi que pour l'éducation que vous m'avez apportée faisant de moi l'homme que je suis actuellement. Je vous aime.

À mes frères, Salih et Faïz qui ont dû endurer mon absence une bonne partie de l'année, surtout quand il s'agissait de me remplacer dans les tâches ménagères.

Je dédie aussi particulièrement ce travail à mes deux grands-pères qui ne sont plus de ce monde, et à qui je souhaite exprimer mon éternelle reconnaissance, eux qui ont changé ma vie par leur présence bienveillante et leurs valeurs inspirantes et qui m'ont transmis à travers leur sagesse la passion et le savoir vivre.

À mes deux grands-mères également et à ma famille à qui je dédie ce modeste travail. Je vous aime.

À mes amis proches, Hadi, Mahfoud, Mahieddine, Linda, Yasmin, Imene, Fadia, Souhila et Hanane pour leur présence et soutien durant les moments de joie, mais aussi lors de moments de doutes et de stress et qui ont rendu ces dernières années et ces derniers mois plus agréables. Je suis très content de vous avoir autour de moi. Merci.

À mes collègues présents dans le laboratoire durant ces six derniers mois, avec qui l'atmosphère a été conviviale, et avec qui j'ai partagé des moments de collaboration et de discussions stimulantes.

Et enfin j'aimerais remercier tout ceux qui ont contribué de près ou de loin au bon déroulement de ce projet et de toutes ces années.

Khalil YADI

*"Whatever you do in this life,
it's not legendary unless your
friends are there to see it."*

- Barney Stinson, "The Bro Code" *How I Met Your Mother*

Penser à tout ce chemin que j'ai parcouru pour en arriver ici me rend fière et un peu nostalgique. Il y a eu des regrets, des obstacles, des moments rudes, mais les personnes qui m'ont entourées sont parvenues à diluer ces pensées sombres ; rien n'aurait été pareil sans leur aide et soutien , je leur dédie ce travail :

À mon meilleur ami, mon cousin, mon binôme **Khalil**, pour son dévouement, sa motivation, ses blagues et son soutien incommensurable malgré mes sautes d'humeur et ma paresse, ton amitié m'est tellement chère, je suis heureuse de ce qu'on a accompli ensemble.

Je dédie ce travail à mes parents, merci d'avoir foi en moi, de votre patience. Merci Maman, pour tes sacrifices, ton amour, tes encouragements et ton appui dans mon travail, même si tu ne comprends pas tout à fait ce que je fais.

À mon frère et ma soeur, ma petite *Sofia* et mon petit *Ramzi* (plus si petit), merci de me soutenir, de m'aimer, et de ne jamais me laisser me reposer en paix, je vous souhaite un parcours encore plus fructueux que le mien.

À mon grand-père, mon papi qui repose en paix, qui a fait de moi la femme que je suis aujourd'hui et qui aurait aimé être présent parmi nous, tu as mon affection et ma reconnaissance éternelles pour tout ce que tu m'as appris, j'espère te rendre fier.

À ma famille, mes grand-mères, pour leur soutien.

À mes amies de toujours, Linda et Yasmin, vous êtes mon baume au coeur, je ne vous remercierai jamais assez pour votre présence, vos rires et vos idées atypiques, vous êtes extraordinaires.

À El-Hadi, Mahieddine, Nadir, Meryem, Salim, Zyad, Kamar, qui prennent toujours un peu de temps pour m'écouter me plaindre et m'éclairer dans mes moments d'égarements (aussi nombreux soient-ils).

A Souha, Tadjou et Amen, dont la présence, l'amitié et le soutien restent indéfectibles depuis tellement longtemps.

À Mahfoud, merci d'avoir toujours cru en moi et d'avoir vu en moi ce potentiel que je refusais de voir. Tu m'as aidé à aller au bout de moi-même.

À Réda, merci pour tes précieux conseils, tes encouragements, et ton aide sans limites.

À Monsieur **SM.Khefif**, qui m'a aidé à garder la tête sur les épaules, aussi lourde de doutes fût-elle. Merci pour votre sagesse, la machine est lancée.

À tous les enseignants qui ont embelli mon parcours académique : M. Chiali, M. Benallal, M. Mokhtari, M.Ladjeroud, M.Maliki, M.Houbad, M.Habchi, M.Belaroussi, M.Kerboua. Vous m'avez transmis l'amour du savoir, merci.

Enfin, je dédie ce travail à toute personne ayant contribué de près ou de loin à ce projet, Merci à vous.

Ghizlène KHELIL

REMERCIEMENTS

Nous tenons tout d'abord à exprimer notre profonde gratitude à notre encadrant, **M. CHERKI Brahim**, pour avoir accepté de nous diriger dans notre travail de recherche ; ses efforts incessants ainsi que ses précieux conseils nous ont permis de mener à bien la réussite de ce projet.

Nous tenons également à remercier tout particulièrement **M. ADJIM Ramz-Eddine Abderrezak**, responsable du FabLab, pour ses conseils, son aide constante et sa bonne humeur qui ont été source de motivation lors de ces six mois de travail.

Nous souhaitons de même remercier **M. MERAD Lotfi**, pour avoir accepté de présider l'honorable jury. Que **M. ABDI Sidi-Mohammed** et **Mme GHOUALI Amel** soient aussi remerciés d'avoir accepté d'évaluer notre travail en tant qu'examineurs.

Nous tenons enfin à exprimer nos remerciements à **M. ROUISSAT Boucherit**, directeur de l'ESSAT, ainsi qu'à **M. KHEFIF Sidi-Mohammed** pour nous avoir accordé sa confiance en nous permettant d'avoir accès au FabLab.

Les aspirations antérieures ne percent pas les murs de la destinée.

- Aphorismes d'Ibn Ata Allah -

TABLE DES MATIÈRES

Remerciements	iii
Introduction générale	1
1 État de l'art	3
1.1 Introduction	3
1.2 Historique de la main artificielle	3
1.2.1 Évolution de la main robotique anthropomorphique	5
1.3 Les actionnements et sous-actionnements	6
1.3.1 Un doigt actionné à chaque phalange	7
1.3.2 Un doigt sous-actionné avec système de transmission	7
1.3.3 Un doigt simplement sous-actionné	7
1.4 La préhension	7
1.5 Conclusion	9
2 Modèle 3D	10
2.1 Introduction	10
2.2 Présentation du modèle 3D utilisé	10
2.3 Conclusion	14
3 Génération de la trajectoire et simulation	15
3.1 Introduction	15
3.2 Génération du modèle 3D	15
3.3 Étude de la trajectoire	19
3.3.1 Approche par variables articulaires	19
3.3.2 Approche par variables opérationnelles	26
3.4 Conclusion	30

4	Préhension et détection de corps	31
4.1	Introduction	31
4.2	Simulation	31
4.2.1	Préhension d'un cube	36
4.2.2	Préhension d'un cylindre	37
4.3	Conclusion	44
A	Programme Matlab 1	47
B	Programme Matlab 2	48
C	Programme Matlab 3	49
D	Programme Matlab 4	51
E	Programme Matlab 5	52

TABLE DES FIGURES

1.1	Le prototype original “Hüfner hand” (à gauche). La main électrique utilisée par <i>Reiter</i> pour la première prothèse myoélectrique (à droite)	4
1.2	Photographie de la première prothèse myoélectrique “Russian hand”	4
1.3	De gauche à droite : Photographie de la main UTAH III, Modèle I de la main Belgrade/USC	5
1.4	Développement des mains robotiques au 21 ^{ème} siècle	6
1.5	Modèle d’un doigt à 3 phalanges (tirée de [9])	6
1.6	Classification des postures de préhension par Napier [12]	7
1.7	La main sous forme de boîte noire	8
2.1	“Robotic hand concept” de Nerman Turkic	11
2.2	Modèle modifié utilisé de la main	11
2.3	Base de la main	12
2.4	Assemblage du doigt	12
2.5	Dimensions des pièces composants le doigt (en mm)	13
2.6	Liaison entre les doigts et la base de la main	13
2.7	Différentes positions des doigts	13
3.1	Modèle généré	16
3.2	Bloc “HAND_BASE”	17
3.3	Bloc “Finger Assembly”	17
3.4	Liaison rotoïde	18
3.5	Propriétés des liaisons rotoïdes	18
3.6	Modèle 3D de la main généré sur Matlab	19
3.7	Allure de la variation des angles dans le temps	20
3.8	Schéma des articulations du doigt	21
3.9	Région admissible de l’effecteur du doigt	22
3.10	Réponse après ajout du P.I.D au système	23
3.11	Mouvement de la main	24

3.12	Trajectoire de l'effecteur final des doigts	24
3.13	Schéma Simulink global	25
3.14	Schéma des articulations	26
3.15	Configuration coude haut et coude bas	27
3.16	Contrôle de position par MGI (pour $\Phi = 45^\circ$ à gauche, et $\Phi = 90^\circ$ à droite)	28
3.17	Schéma Simulink global avec modèle géométrique inverse	29
4.1	Ajout du solide au système	32
4.2	Les conditions <i>if</i> en entrée	33
4.3	Bloc "Rigid transform Sensor"	33
4.4	Repère lié au capteur	34
4.5	Schéma Simulink global avec détection de collisions	35
4.6	Cube $100 \times 100 \times 100mm$	36
4.7	Préhension d'un cube	36
4.8	Trajectoires des effecteurs finaux des doigts	37
4.9	Cylindre	38
4.10	Problème collision	38
4.11	Emplacement des capteurs sur la phalange 1 et 2	39
4.12	Préhension d'un cylindre	40
4.13	Trajectoires des effecteurs finaux des doigts	40
4.14	Préhension d'un cylindre penché	41
4.15	Trajectoires des effecteurs finaux des doigts	41
4.16	Préhension d'une barre	42
4.17	Les conditions <i>if</i> appliquées sur chaque θ	42
4.18	Schéma Simulink global avec multiple détection de collisions par doigt	43

LISTE DES TABLEAUX

3.1	Paramètres de Denavit-Hartenberg du modèle du doigt	21
3.2	Les paramètres a_i de DH des différents doigts	22

INTRODUCTION GÉNÉRALE

Charles Darwin a écrit que “L’homme n’aurait jamais atteint sa place prépondérante dans le monde sans l’usage de ses mains”.

La main devient donc symbole de travail, de dignité et de pouvoir, d’où la valorisation de tous les travaux accomplis à l’aide des mains, sans utilisation de mécanismes artificiels. Il s’agit là d’une preuve irréfutable de l’admiration que portent les humains à cet organe depuis les temps anciens.

Cependant, elle demeure un référentiel inégalé par les chercheurs dans le domaine artificiel. En effet, même s’il est possible de réaliser des mains robotiques, ressemblantes au premier coup d’oeil à la main humaine, leurs caractéristiques, elles, ne permettent pas de réaliser les tâches voulues avec la même finesse et précision que la main humaine.

Les ingénieurs tentent donc tant bien que mal d’égaliser la supériorité et complexité de ce membre, mais beaucoup de questions subsistent, comme le choix de concevoir une main anthropomorphe, ou une structure plus “simplifiée”. Se pose également la question du choix des actionneurs aux liaisons rotoïdes de la main : doit-on établir un système sous-actionné, ou faire le choix ambitieux de placer un actionneur à chaque articulation ? Toutes ces questions ont pour objectif de rapprocher le plus l’humain de la robotique humanoïde, et à trouver le meilleur compromis entre ressemblance humaine et faisabilité mécanique.

Mais quel est l’objectif de tous ces efforts fournis finalement ?

Le développement du secteur robotique de type humanoïde, fait qu’il est nécessaire de pouvoir faire évoluer les mains robotiques dans un milieu humain, afin qu’elles puissent être fonctionnelles dans un milieu conçu pour les humains. L’une des premières tâches que les mains robotiques sont amenées à accomplir s’associe au mouvement de préhension.

Afin de réaliser cette action, nous sommes évidemment contraints de choisir la manière avec laquelle on veut approcher cette planification de mouvements. Plusieurs méthodes existent, que l’on exposera au fur et à mesure de ce document.

Ce mémoire se divisera en 4 chapitres :

Le premier chapitre, retracera le parcours des mains robotiques et l'évolution technologique des mains robotiques anthropomorphiques, tout en introduisant le principe d'actionnement et de sous-actionnement, ainsi que la préhension, et la manière dont elle est pensée afin d'être intégrée dans une simulation 3D.

Le second chapitre sera dédié à notre choix du modèle 3D de main anthropomorphique. Nous y apportons des modifications via le logiciel Solidworks afin de réaliser quelques simplifications correspondantes à nos besoins dans ce projet.

Le chapitre trois quant à lui, sera entièrement dédié au contrôle de la main, ou plus précisément, à la génération de la trajectoire des doigts, à partir de leur réalisation de mouvements de flexion et d'extension. Nous expliquerons également au fil de ce chapitre la manière avec laquelle nous avons couplé le logiciel SolidWorks au logiciel Matlab/Simulink.

Pour finir, le quatrième chapitre se concentrera sur l'intégration du mouvement de préhension à la simulation du modèle 3D de la main anthropomorphique. Cette préhension se fera à travers l'utilisation de divers objets de formes différentes. Nous exposerons les contraintes liées à notre modèle et au logiciel, ainsi que le choix de planification du mouvement de préhension.

CHAPITRE

1

ÉTAT DE L'ART

1.1 Introduction

Dans le but de réaliser une synthèse de la main robotique et de s'aventurer dans le chemin périlleux d'une simulation en 3 dimensions, il est nécessaire de mettre à notre connaissance les travaux antérieurs effectués par les pairs, en stipulant les motivations derrière de tels travaux.

Les principales visées de la conception d'une main robotique ont été soit prothétiques soit industrielles, menant ainsi à des dispositifs très complexes (certains ont présenté jusqu'à quarante actionneurs), ou à d'autres très simples, limitant l'éventail d'activité. Nous tenterons à travers ce chapitre de présenter au mieux l'historique de la main robotique, ainsi que celui de la modélisation de la préhension, afin de souligner la philosophie qui se trouve derrière les mains robotiques de type préhenseurs.

1.2 Historique de la main artificielle

La raison majeure qui a poussé l'homme à se questionner sur l'intérêt de tels dispositifs est sans doute la plus sanglante de toutes : l'augmentation considérable des guerres. En effet, ces dernières ont été le théâtre d'un nombre faramineux d'amputés, créant ainsi un problème non-négligeable pour la classe dirigeante.

Samuel Decker fut l'un des premiers amputés de guerre au 19^{ème} siècle. Refusant de se laisser aller aux conditions difficiles de cet handicap, il confectionna ses propres prothèses. Il devint un pionnier dans la confection de prothèses et membres artificiels, mais également l'initiateur d'un mouvement encore plus grand : la confection de mains artificielles et leur intégration dans notre quotidien.

Cet événement a donné place à de nouvelles perspectives : un grand nombre de scientifiques ont exploré l'idée. On peut notamment citer *Jacob Hüfner* (1874-1968)

et son **Hüfner hand** (voir Fig.1.1) à la fin de la seconde guerre mondiale : il s'agit d'une prothèse VC (*Voluntary Close*), qui permet à son utilisateur d'initier des mouvements de préhension [16]. Ce dispositif a été repris par la suite par *Reinhold Reiter* [5], et est considéré comme la première prothèse myoélectrique (Fig1.1); le système était le même que le précédent, mais comprenait un électroaimant de contrôle et était non-portatif [15].

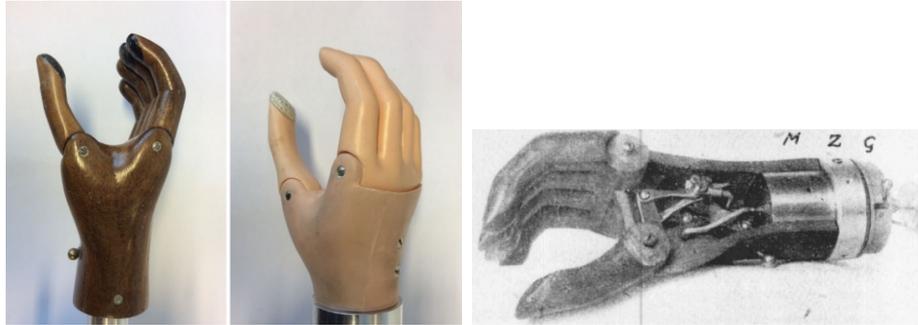


FIGURE 1.1 – Le prototype original “Hüfner hand” (à gauche). La main électrique utilisée par *Reiter* pour la première prothèse myoélectrique (à droite)

Les résultats de performance de cette prothèse furent publiés en 1948, la même année où les transistors furent correctement développés. Beaucoup de suggestions pour des prothèses myoélectriques furent formulées les années suivantes, et ce sont les chercheurs soviétiques qui ont été les premiers à intégrer les transistors dans leurs recherches. Le scientifique russe *Alexander Kobrinski* en 1960 créa une prothèse myoélectrique, appelée **The Russian hand** (voir Fig.1.2), qui fut un véritable succès en son temps. Le scientifique a eu l'ingéniosité de placer les batteries sur une ceinture, reliée de manière filaire à la prothèse, et il a utilisé des transistors pour réduire l'encombrement. Ce dispositif présentait un bon nombre de problèmes (faible force de préhension, interférences électriques, mouvement lent), mais sa commercialisation a néanmoins vu le jour en Angleterre et au Canada en plus de l'URSS [15].

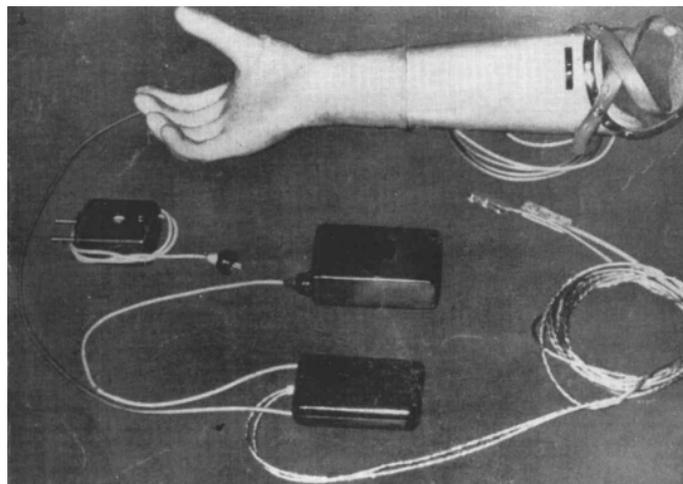


FIGURE 1.2 – Photographie de la première prothèse myoélectrique “Russian hand”

1.2.1 Évolution de la main robotique anthropomorphique

La fin des années soixante est marquée par l'avènement de la robotique, et c'est seulement à cette période que l'idée qu'une machine anthropomorphique¹ puisse penser par elle-même s'est démocratisée.

Cette période se caractérise par la crise de désindustrialisation que subit l'Europe [8] ; la course économique n'est que plus grande et c'est à partir de là qu'affluent les idées quant à la modernisation des chaînes de production, en misant sur le développement des systèmes de manipulation. La volonté de créer des robots manipulateurs avait pour but d'augmenter les performances des productions industrielles : il était avancé que la dextérité d'une machine serait profitable pour des tâches telles que les assemblages, les inspections, etc, sans omettre que leur développement serait également profitable au secteur militaire.

T.Okada a proposé en 1982 une architecture à trois doigts, sans que la paume ne participe à la prise : la transmission entre moteurs électriques et les doigts se faisait grâce à des tendons [14]. Le développement se maintint avec la main Utah/M.I.T (voir Fig1.3), qui avait un actionnement pneumatique mais utilisait des tendons afin de transmettre le mouvement [11]. Par la suite, il y eut la main Belgrade USC (voir Fig1.3), pensée tout comme la main Okada, pour réaliser des tâches de préhension en s'adaptant à tout type de formes. Elle est composée de cinq doigts, avec un pouce fixe, et comporte des servomoteurs situés à l'intérieur de la structure du poignet. L'actionnement de chaque phalange est transmis à l'aide de liaisons mécaniques afin d'éviter l'utilisation de câbles ou de tendons. Il est à ajouter que la main est équipée de capteurs de position et de force[3].

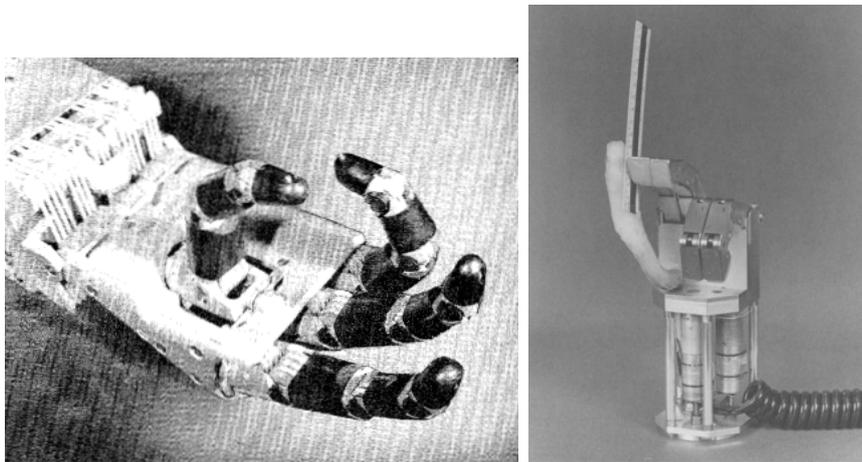
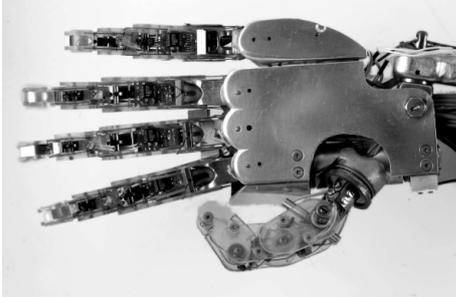


FIGURE 1.3 – De gauche à droite : Photographie de la main UTAH III, Modèle I de la main Belgrade/USC

Cette course à la main parfaite a permis un pas de géant dans le domaine des systèmes de manipulation. On peut notamment citer la main Shadow [6] ou encore la main ACT [17] qui présentent des similitudes extraordinaires avec la main humaine (voir Fig.1.4).

1. Qui a la forme, l'apparence d'une main humaine.

Il est à noter que bien d'autres informations peuvent être obtenues à partir du rapport de Biagiotti [4] pour de plus amples approfondissements dans le développement technologique des mains robotiques.



(a) Main Shadow



(b) Main ACT

FIGURE 1.4 – Développement des mains robotiques au 21^{ème} siècle

1.3 Les actionnements et sous-actionnements

Dans cette étude qui traite des types d'actionnements, nous tiendrons compte seulement des mouvements de flexion et d'extension des doigts.

La main humaine représente un incroyable préhenseur, que les chercheurs ont tenté de reproduire au mieux, cependant le nombre élevé de degrés de liberté de la main rend compliqué le choix de l'actionneur.

Un doigt est composée de trois phalanges, liées entre elles grâce à des tendons et des ligaments, et dans la plupart des modèles de mains robotiques, les chercheurs s'en tiennent à ce nombre. Il est important, lors de la modélisation d'une main robotique, de définir le type de mécanisme, qui fera office de tendons et donc, d'actionneurs. Nous pouvons représenter le type de système que l'on désire étudier dans un plan, comme dans la figure 1.5.

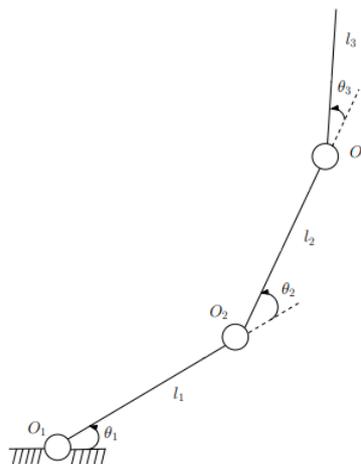


FIGURE 1.5 – Modèle d'un doigt à 3 phalanges (tirée de [9])

1.3.1 Un doigt actionné à chaque phalange

Ce type d'actionnement représente le moyen le plus sûr d'obtenir les mains robotiques versatiles ayant un mouvement stable, et un contrôle complexe. Il s'agit en général de placer un moteur en chaque liaison rotoïde O_1 , O_2 et O_3 de la figure 1.5, accompagnant le tout d'un arsenal de câbles et de logiciels pour le faire fonctionner.

1.3.2 Un doigt sous-actionné avec système de transmission

Ce type d'actionnement consiste à ne placer l'actionneur qu'en une seule liaison rotoïde (O_1 par exemple), et placer un mécanisme de transmission sur le reste des liaisons (O_2 et O_3), tels que des ressorts, des câbles, des mécanismes pneumatiques ou électriques.

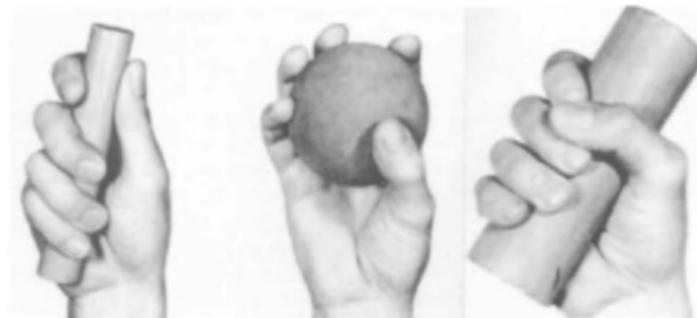
Ainsi, lorsque l'actionneur enclenche le mouvement, celui-ci est transmis au reste des liaisons par les mécanismes cités ci-dessus.

1.3.3 Un doigt simplement sous-actionné

A la différence du sous-paragraphe 1.3.2, ce type d'actionnement consiste à appliquer un actionneur à une des liaisons rotoïdes (O_3 par exemple), et à n'appliquer aucun couple, directement, sur le reste des liaisons (O_1 et O_2). Leur mouvement est donc dû seulement au couple appliqué à O_3 .

1.4 La préhension

La préhension, ou “Grasping” en anglais, est l'action par laquelle la main effectue la tâche de saisir, maintenir un objet, afin d'atteindre un objectif de manipulation dudit objet. Cette manipulation n'est cependant pas anodine : l'action de prendre une boîte, par exemple, diffère de l'action de prendre un oeuf, ce qui représente principalement le type de classification que Napier utilise dans la figure 1.6. Il a maintenu auprès de ses pairs que l'influence d'une posture de la main s'établit d'après plusieurs facteurs : la taille de l'objet, son poids, sa forme, les caractéristiques de surface (texture, température...), et, le plus important, l'objectif de préhension [12]. C'est donc un des principes majeurs auquel plusieurs études de mains robotiques se réfèrent.



A. POWER GRASP B. PRECISION GRASP C. COAL HAMMER

FIGURE 1.6 – Classification des postures de préhension par Napier [12]

Ainsi, pour manipuler un objet, deux variables d'entrée s'imposent : la tâche à réaliser, et le type d'objet à manipuler (voir Fig.1.7).

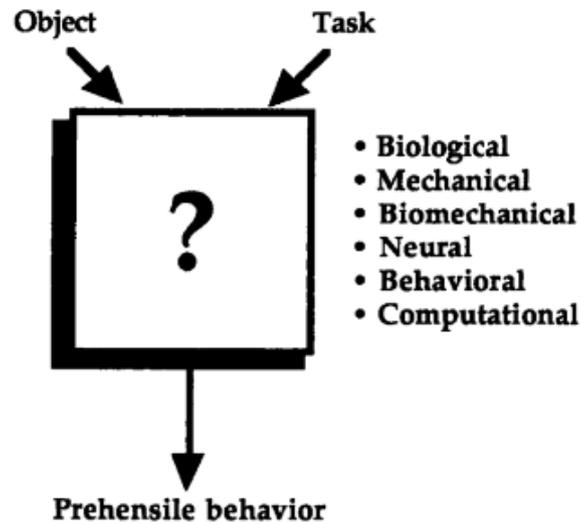


FIGURE 1.7 – La main sous forme de boîte noire

La préhension est un sujet qui a beaucoup fait parler de lui de par sa pluridisciplinarité (biomécanique, robotique, animation, neurosciences). Tous ces domaines doivent néanmoins franchir une étape commune : la simulation.

Plusieurs méthodes ont été mises en place, afin de pouvoir réaliser la simulation de l'opération de saisie d'objet. Nous citons :

- **Opération basée sur des données** : cette opération suggère l'existence d'une base de données suffisante de postures de préhension afin de proposer une solution existante dans cette base, puis qui sera optimisée afin de répondre aux contraintes liées, en utilisant des techniques de cinématique inverse. La nouvelle prise obtenue sera par la suite incorporée à la base de données.
- **Opération guidée par des capteurs et à base d'apprentissage** : cette opération consiste à construire une représentation de la tâche après récupération des données sensorielles à partir des différents capteurs. A partir de là, un modèle, même inexact, a la possibilité d'être corrigé à partir des informations obtenues, sensorielles ou par apprentissage.
- **Opération basée sur la géométrie de la tâche** : elle se concentre sur la détermination des emplacements des contacts sur un objet pour assurer des propriétés mécaniques spécifiques : les algorithmes géométriques sont utilisés pour déterminer les emplacements optimaux des contacts sur un objet, de manière à garantir certaines propriétés mécaniques lors de la saisie, telles que la fermeture en force et en forme (Force et form closure). Dans ce contexte, la main est représentée par un ensemble de contacts. L'inconvénient de cette méthode est qu'elle ne fournit pas la configuration que la main est supposée entreprendre afin de réaliser la saisie de l'objet.

1.5 Conclusion

La modélisation d'une main à cinq doigt représente un enjeu majeur si l'on décide de choisir une main comportant des doigts actionnés à chaque phalange. En effet, le nombre de degrés de liberté de celle-ci rendent l'application de la simulation à un modèle réel assez compliquée étant donné le nombre d'actionneurs nécessaires pour la faire fonctionner.

A travers ce chapitre, nous avons pu explorer l'historique des mains robotiques, en plus de nous être familiarisés avec la notion de **Grasping**, qui représente une des applications les plus courantes des mains robotiques.

CHAPITRE

2

MODÈLE 3D

2.1 Introduction

Lors de notre étude, nous avons utilisé le modèle 3D d'une main rattachée à un socle, disponible en Open-source, modifiable à travers le logiciel de conception et modélisation 3D "*SolidWorks*".

Nous avons comme objectif, au cours de ce chapitre, de spécifier les différentes caractéristiques de ce modèle, en plus des modifications et simplifications que nous avons été amenés à réaliser, afin d'être le plus fidèle à un modèle réel de main.

2.2 Présentation du modèle 3D utilisé

Un large panel de modèles 3D de mains anthropomorphes est disponible sur internet, en Open-source. Nous avons choisi le modèle "Robotic hand concept" de Nerman Turkic depuis la bibliothèque en ligne de modèles 3D SolidWorks "GRAB-CAD COMMUNITY" [1] (voir Fig.2.1).

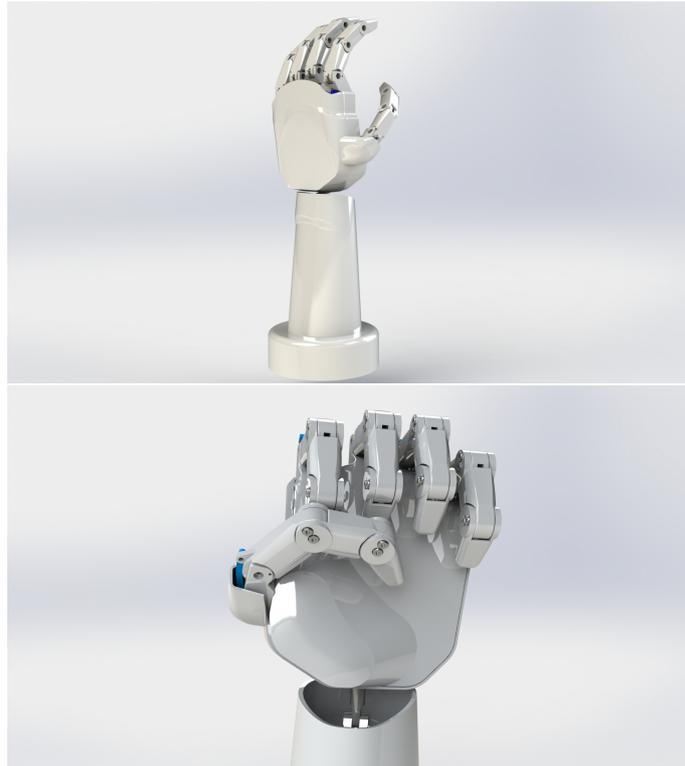
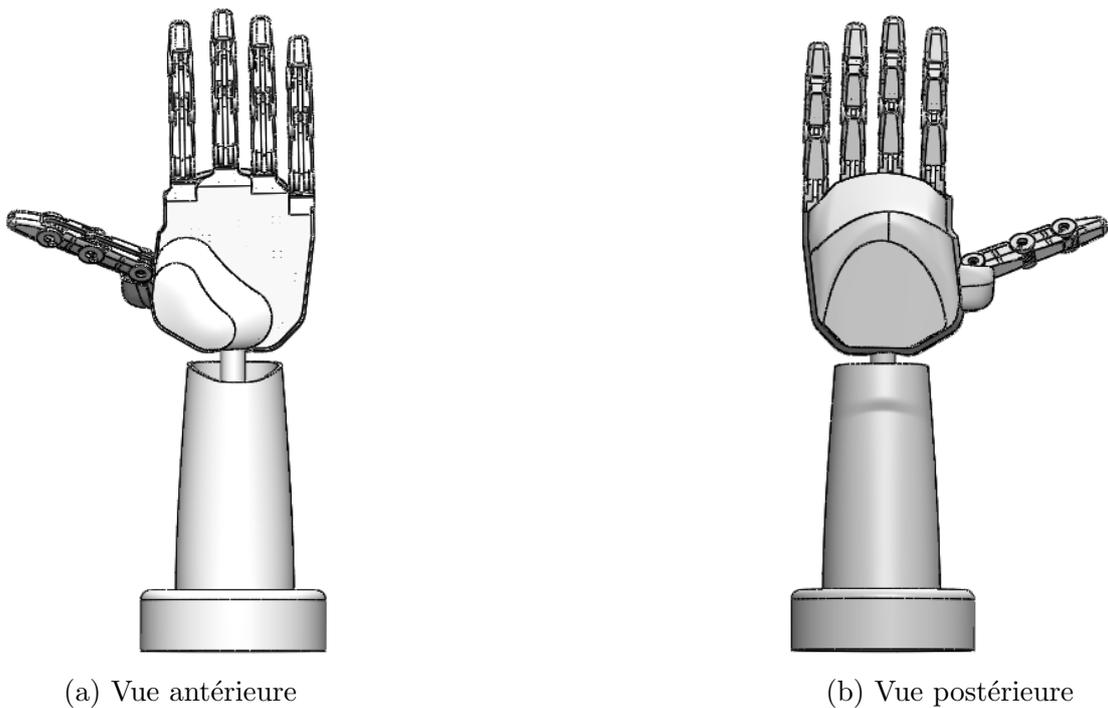


FIGURE 2.1 – “Robotic hand concept” de Nerman Turkic

Notre but étant d’appliquer une commande et de simuler le comportement d’un système actionné en chaque phalange, nous avons allégé le modèle en supprimant les éléments non nécessaires à notre étude notamment les mécanismes d’actionnement des doigts et du poignet déjà présents dans le modèle d’origine (voir Fig.2.2).



(a) Vue antérieure

(b) Vue postérieure

FIGURE 2.2 – Modèle modifié utilisé de la main

Le socle et la base de la main de la figure 2.3 sont fixés dans l'espace tandis que les doigts représentent la seule partie capable de se mouvoir.

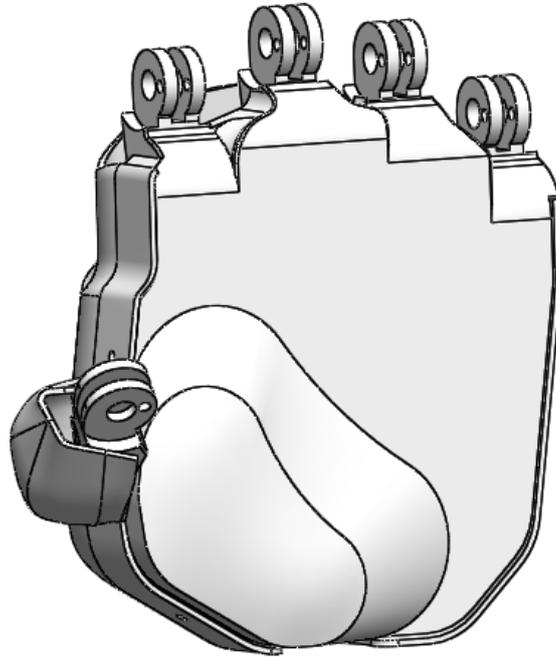


FIGURE 2.3 – Base de la main

La caractéristique principale de ce modèle est que tous les doigts sont identiques avec les mêmes dimensions (voir Fig.2.5). Ces derniers sont composés de 3 pièces distinctes qui s'emboîtent entre elles (voir Fig.2.4), et qui représentent les 3 phalanges du doigt. Chacune des extrémités d'une pièce de phalange est coaxiale avec la pièce adjacente.

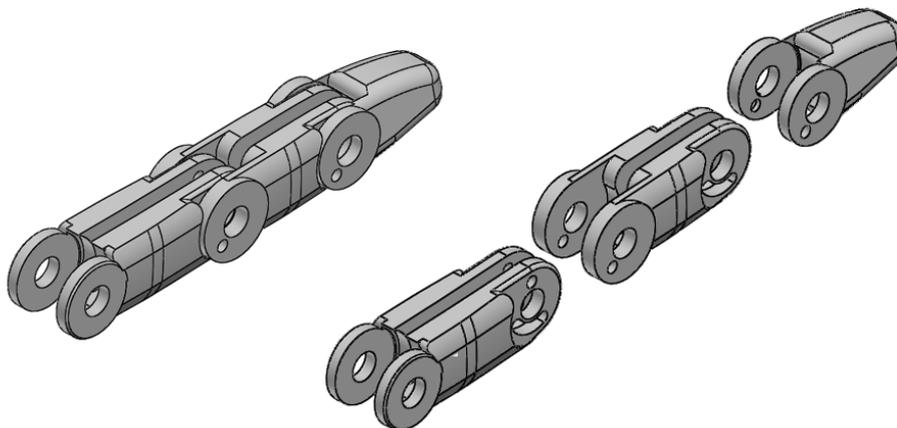


FIGURE 2.4 – Assemblage du doigt

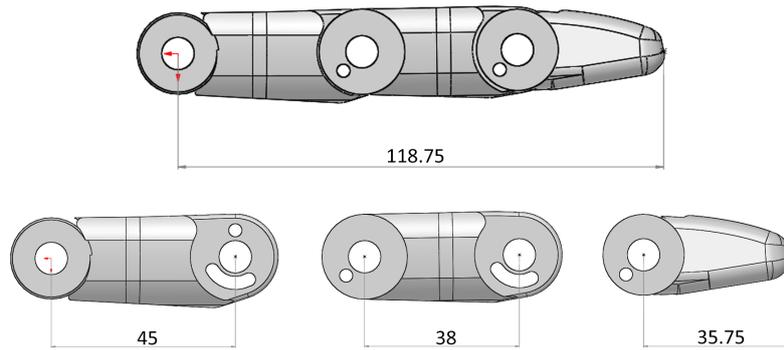


FIGURE 2.5 – Dimensions des pièces composant le doigt (en mm)

L'extrémité de la dernière phalange, qui représente l'extrémité du doigt est, elle, coaxiale avec la pièce de la base de la main (voir Fig.2.6).

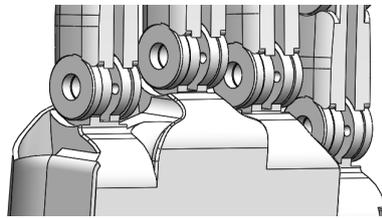
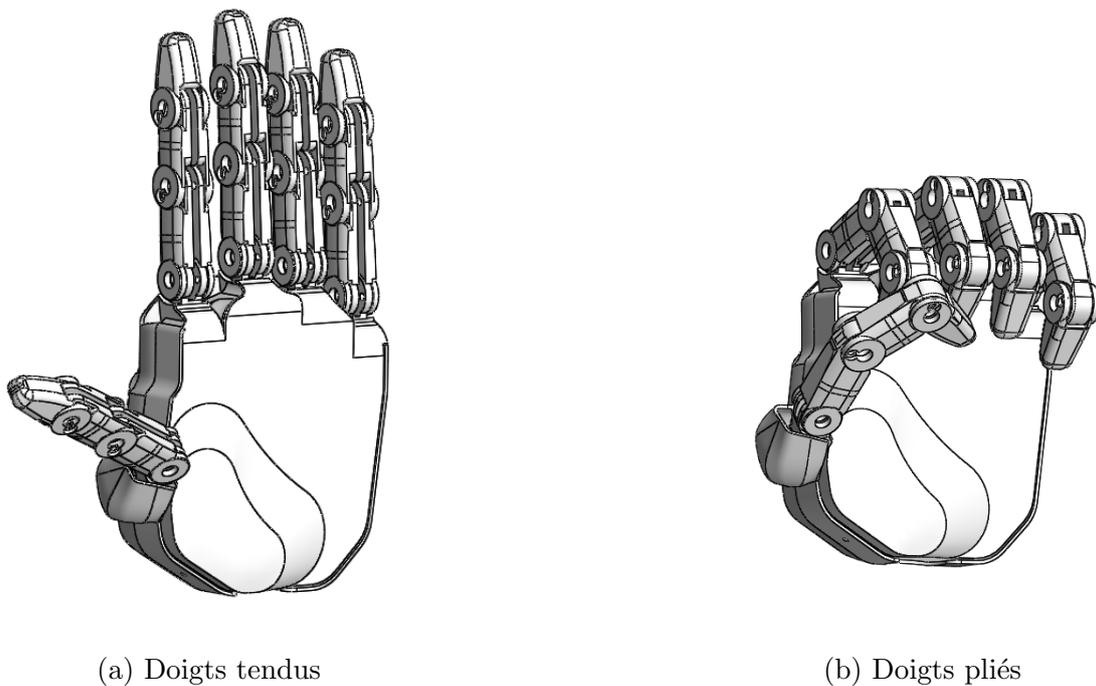


FIGURE 2.6 – Liaison entre les doigts et la base de la main

Ces coaxialités entre les phalanges des doigts leur permettent des mouvements de rotation autour de ces axes. Nous pouvons donc obtenir différentes positions des doigts comme montré dans la figure 2.7.



(a) Doigts tendus

(b) Doigts pliés

FIGURE 2.7 – Différentes positions des doigts

2.3 Conclusion

Une fois le modèle simplifié selon nos besoins et les liaisons responsables des rotations établies, nous pouvons effectuer une étude sur la dynamique du système et appliquer une commande afin de contrôler la position des doigts. Étant donné que les doigts ont tous les mêmes dimensions, et qu'ils sont découplés, l'étude de la dynamique d'un seul doigt suffit. On dupliquera le modèle dynamique cinq fois.

CHAPITRE

3

GÉNÉRATION DE LA TRAJECTOIRE ET SIMULATION

3.1 Introduction

Nous avons, au cours du chapitre précédent, présenté le modèle 3D de la main sur lequel nous allons appliquer une commande de position des doigts. Pour cela nous avons utilisé la bibliothèque “Simscape Multibody” de Simulink au sein du logiciel Matlab qui fournit un environnement de simulation multicorps pour les systèmes mécaniques 3D [2]. Cette bibliothèque nous permet de modéliser des systèmes multicorps à l’aide de blocs représentant des corps, des articulations, des contraintes, des éléments de force et des capteurs. Elle représente donc un outil puissant qui formule et résout les équations de mouvement pour le système mécanique complet et de générer une animation 3D pour visualiser la dynamique du système [2].

“Simscape Multibody” nous donne notamment la possibilité d’importer des assemblages CAO complets, y compris toutes les masses, inerties, articulations, contraintes et géométries 3D, dans notre modèle [2]. Dans notre cas, nous avons dû installer le plug-in “Simscape Multibody Link” à Solidworks pour pouvoir le jumeler avec Matlab et pouvoir exporter l’assemblage 3D de la main (voir Fig.2.2) en fichier “.xml”.

3.2 Génération du modèle 3D

À partir du fichier obtenu depuis Solidworks, en exécutant la commande *smimport* (nom-du-fichier.xml) dans la fenêtre de commande de Matlab, ce dernier génère automatiquement le programme associé à notre assemblage dans l’environnement Simulink comme sur la figure 3.1.

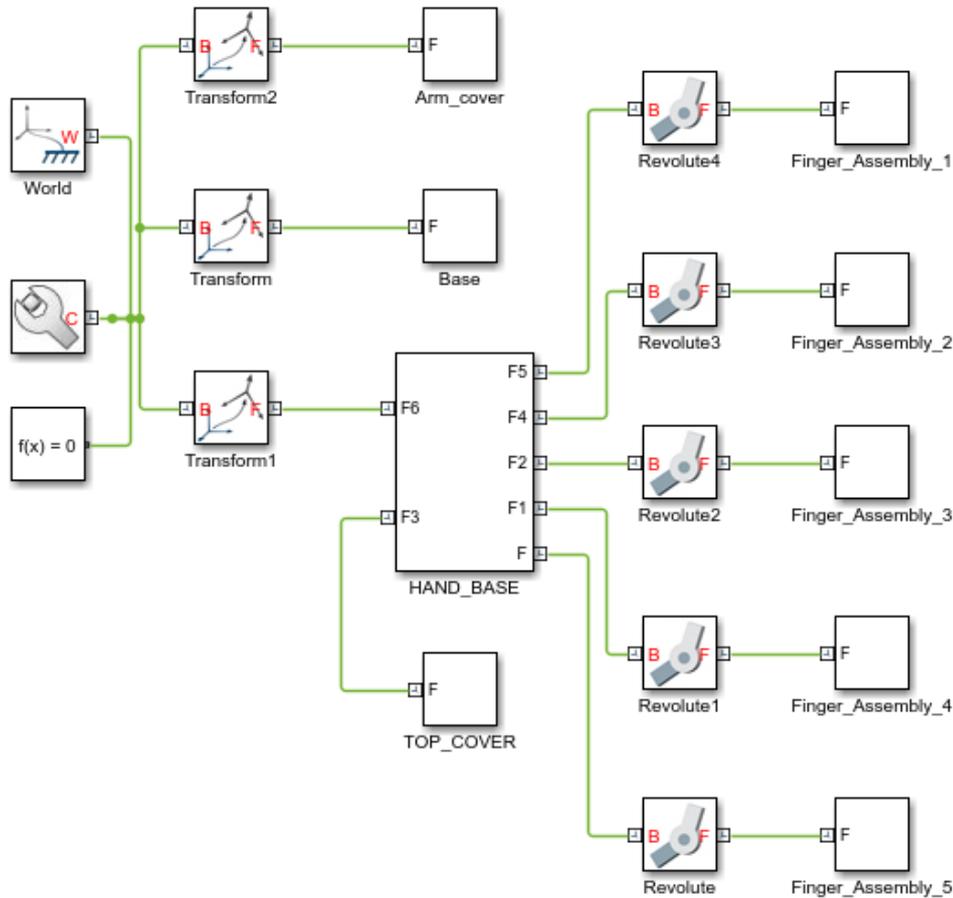


FIGURE 3.1 – Modèle généré

Nous pouvons constater depuis la figure 3.1 que le code généré contient des blocs représentant tous les corps présents dans notre modèle et qui sont reliés entre eux selon l'assemblage du modèle SolidWorks. Les corps fixes de notre modèle sont reliés à un repère monde (à gauche), tandis que les autres sont chacun reliés à un repère relatif.

Le bloc "HANDBASE" (voir.Fig.3.2) comporte cinq sorties, chacune liée au repère relatif de chaque doigt.

Les blocs "Finger Assembly" présents à droite du schéma de la figure 3.1 représentent les doigts, contenant les différentes pièces et liaisons. (voir Fig.3.3)

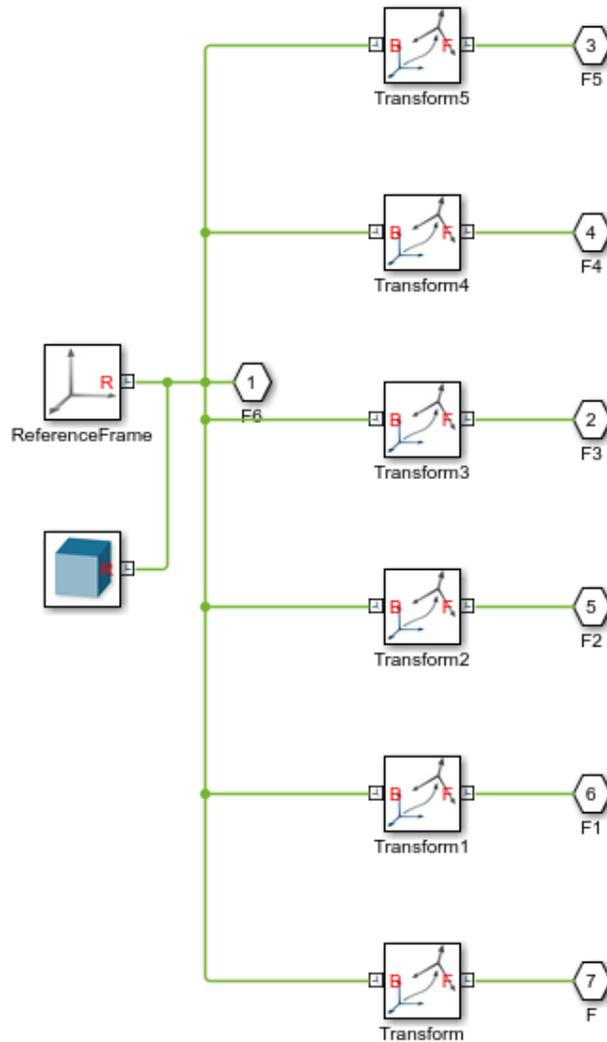


FIGURE 3.2 – Bloc “HAND_BASE”

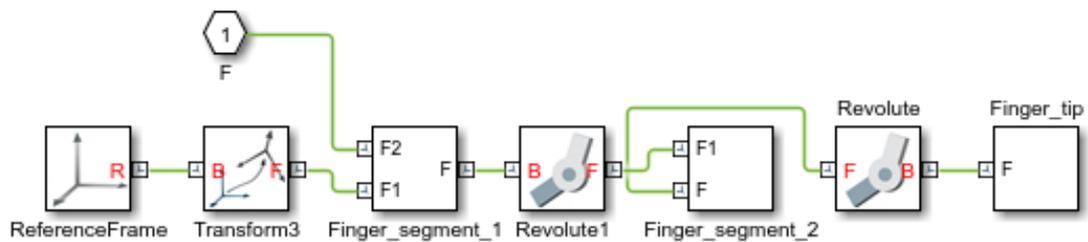


FIGURE 3.3 – Bloc “Finger Assembly”

Nous pouvons remarquer à partir de la figure 3.3 que les blocs “Finger segment 1”, “Finger segment 2” et “Finger tip” représentent respectivement la phalange une, deux et trois du doigt.

Les contraintes de coaxialité définies dans notre modèle 3D sur SolidWorks entre les phalanges des doigts sont automatiquement générées sous forme de blocs “Revolute Joint” qui représentent l’action rotoïde entre les solides (voir Fig.3.4). Ces derniers nous donnent la possibilité de contrôler l’angle de rotation en entrée et de récupérer

la valeur en sortie, ce qui nous servira plus tard à intégrer une boucle de régulation à notre système (voir Fig.3.5).

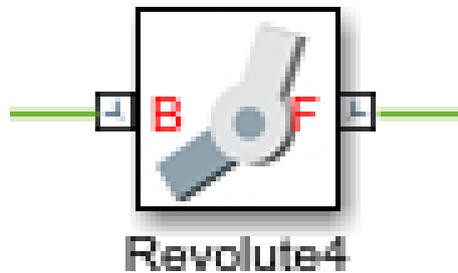


FIGURE 3.4 – Liaison rotoïde

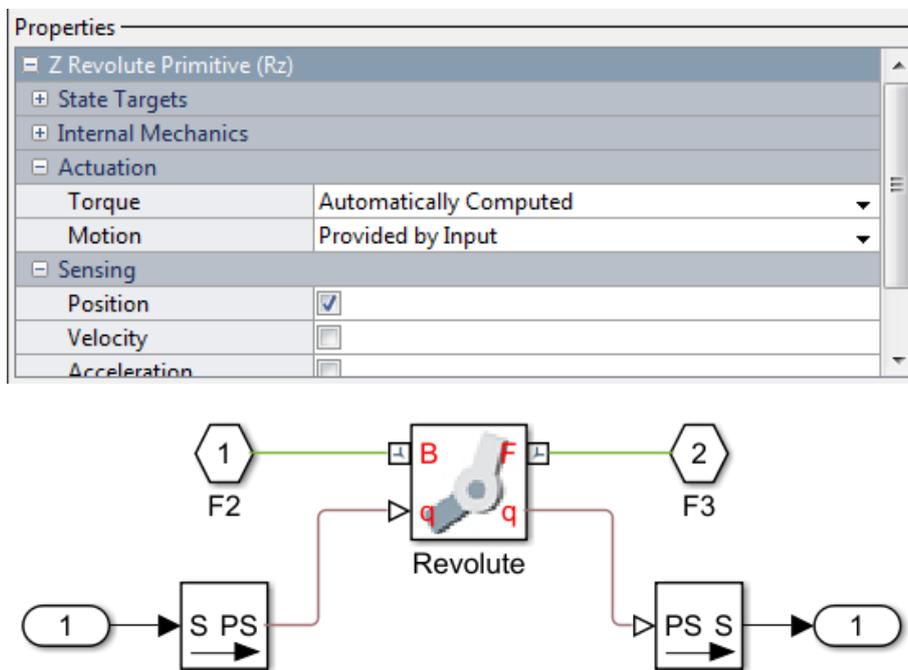


FIGURE 3.5 – Propriétés des liaisons rotoïdes

L'exécution du programme de la figure 3.1 nous génère le modèle 3D de la main sur l'interface 3D de SimMechanics de Matlab (voir Fig.3.6).

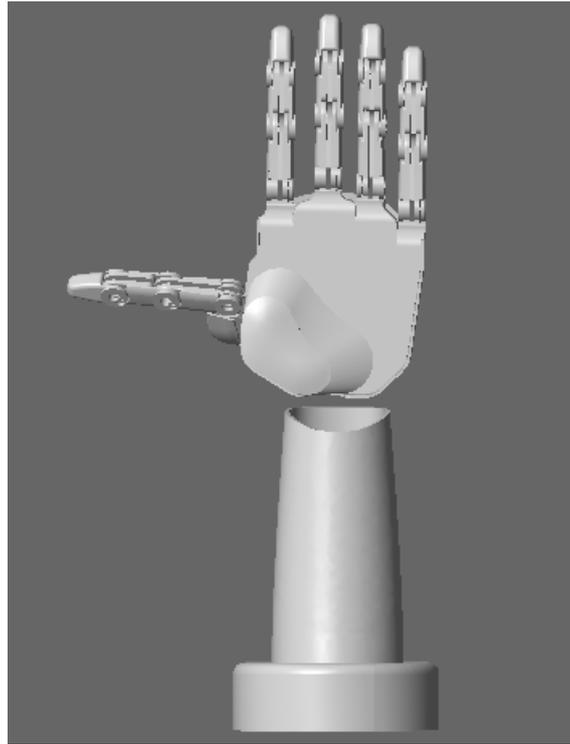


FIGURE 3.6 – Modèle 3D de la main généré sur Matlab

3.3 Étude de la trajectoire

3.3.1 Approche par variables articulaires

Les variables articulaires de notre modèle sont les angles θ entre les phalanges. Pour générer et tracer la trajectoire dans le plan (x, y) à partir de la variation d'angles, nous devons calculer le modèle géométrique direct du système. Les doigts étant identiques, l'étude d'un seul doigt suffit pour concevoir le comportement global du système.

Avant cela, définissons l'allure de la trajectoire. Dans le domaine de la robotique, l'interpolation polynomiale du cinquième degré est souvent utilisée. Cette dernière assure une trajectoire lisse qui est continue en positions, vitesses et accélérations [10]. Le polynôme est obtenu en utilisant les conditions aux limites suivantes :

$$\begin{aligned}\theta(0) &= \theta_{in}, \quad \theta(T) = \theta_f, \\ \dot{\theta}(0) &= 0, \quad \dot{\theta}(T) = 0, \\ \ddot{\theta}(0) &= 0, \quad \ddot{\theta}(T) = 0,\end{aligned}\tag{3.1}$$

où θ_{in} est la valeur initiale, θ_f la valeur finale désirée et T le temps final désiré.

La résolution des six contraintes (3.1) donne la fonction d'interpolation suivante :

$$r(t) = 10 \left(\frac{t}{T}\right)^3 - 15 \left(\frac{t}{T}\right)^4 + 6 \left(\frac{t}{T}\right)^5$$

La trajectoire entre θ_{in} et θ_f est déterminée par l'équation

$$\theta(t) = \theta_{in} + r(t).D, \quad 0 \leq t \leq T$$

où $D = \theta_f - \theta_{in}$.

Pour un mouvement réaliste du modèle et pour éviter les collisions des doigts entre eux et avec la paume de la main, nous avons défini $\theta_{in} = 0^\circ$, $\theta_f = 70^\circ$ et $T = 2s$.

$$\theta(t) = 70 \left(10 \left(\frac{t}{T} \right)^3 - 15 \left(\frac{t}{T} \right)^4 + 6 \left(\frac{t}{T} \right)^5 \right) \quad (3.2)$$

Nous pouvons observer l'allure de la fonction (3.2) sur la figure 3.7.

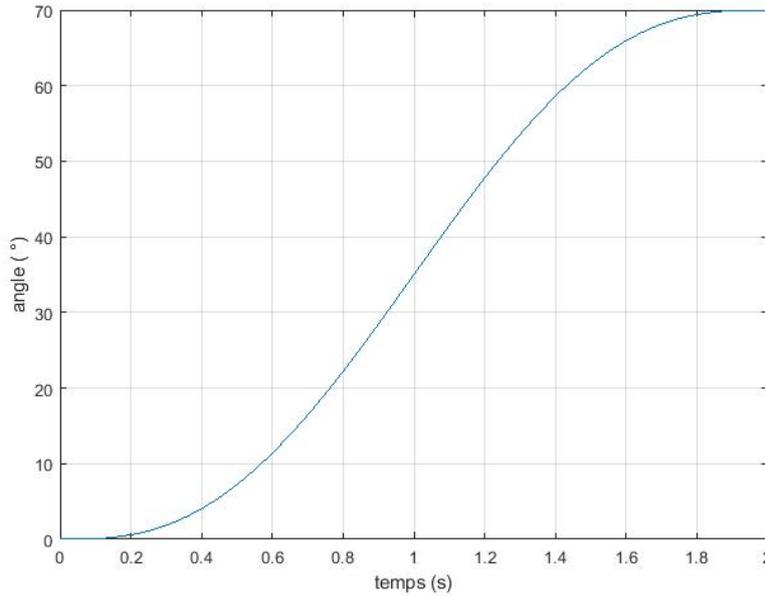


FIGURE 3.7 – Allure de la variation des angles dans le temps

Une fois l'allure du mouvement définie, nous passons au calcul du modèle géométrique direct (MGD). C'est un modèle de transformation entre l'espace opérationnel et l'espace articulaire. Dans notre cas, le MGD nous permet d'obtenir la position (x, y) de l'effecteur terminal qui représente l'extrémité du doigt. Les doigts peuvent être simplifiés par un schéma comme représenté sur la figure 3.8.

Le modèle du doigt représente une chaîne plane ouverte avec 3 liaisons rotoïdes (système 3R) avec une extrémité fixe et une extrémité libre. Pour déterminer le modèle géométrique direct, nous utilisons la convention de Denavit-Hartenberg [7] qui, avec seulement 4 paramètres, nous permet de calculer la position relative de deux corps solides à l'aide de la matrice de passage de Denavit-Hartenberg.

$$T_{i-1,i} = \begin{bmatrix} \cos(\theta) & -\cos(\alpha)\sin(\theta) & \sin(\alpha)\sin(\theta) & r\cos(\theta) \\ \sin(\theta) & \cos(\alpha)\cos(\theta) & -\sin(\alpha)\cos(\theta) & r\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

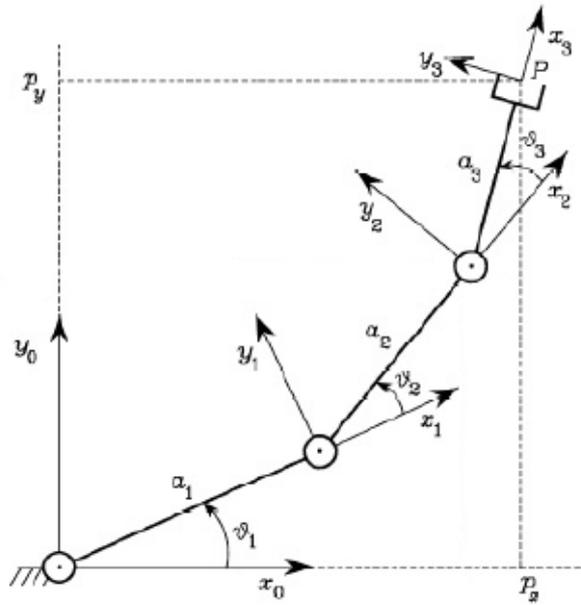


FIGURE 3.8 – Schéma des articulations du doigt

Les paramètres de notre modèle de doigt sont définis selon la figure 3.8 dans la tableau 3.1.

Axe i	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3

TABLE 3.1 – Paramètres de Denavit-Hartenberg du modèle du doigt

Nous pouvons obtenir les matrices de transformation homogènes en remplaçant les paramètres du tableau 3.1 dans la matrice (3.3). Les articulations présentes étant toutes rotoïdes les matrices ont toutes la même structure suivante :

$$A_i^{i-1}(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & \vdots & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & 0 & \vdots & a_i \sin(\theta_i) \\ 0 & 0 & 1 & \vdots & 0 \\ \dots & \dots & \dots & & \dots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \quad i = 1, 2, 3,$$

Le modèle géométrique direct est donc donné par l'équation

$$T_3^0 = A_1^0 A_2^1 A_3^2$$

En définitive,

$$T_3^0 = \begin{bmatrix} \cos(\theta_{123}) & -\sin(\theta_{123}) & 0 & \vdots & a_1 \cos(\theta_1) + a_2 \cos(\theta_{12}) + a_3 \cos(\theta_{123}) \\ \sin(\theta_{123}) & \cos(\theta_{123}) & 0 & \vdots & a_1 \sin(\theta_1) + a_2 \sin(\theta_{12}) + a_3 \sin(\theta_{123}) \\ 0 & 0 & 1 & \vdots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \quad (3.4)$$

Nous avons alors depuis la dernière colonne de (3.4) les expressions des coordonnées relatifs x , y et z de l'effecteur final P.

$$x_P = a_1 \cos(\theta_1) + a_2 \cos(\theta_{12}) + a_3 \cos(\theta_{123}) \quad (3.5)$$

$$y_P = a_1 \sin(\theta_1) + a_2 \sin(\theta_{12}) + a_3 \sin(\theta_{123}) \quad (3.6)$$

$$z_P = 0 \quad (3.7)$$

a_1 (mm)	a_2 (mm)	a_3 (mm)
45	38	35.75

TABLE 3.2 – Les paramètres a_i de DH des différents doigts

Pour des angles donnés et pour des longueurs connues présentées dans le tableau 3.2 (voir Fig.2.5), les équation (3.5), (3.6) et (3.7) nous donnent la position dans l'espace du point P. Ceci nous permet de tracer l'évolution de la trajectoire ainsi que la région admissible (voir Fig.3.9).

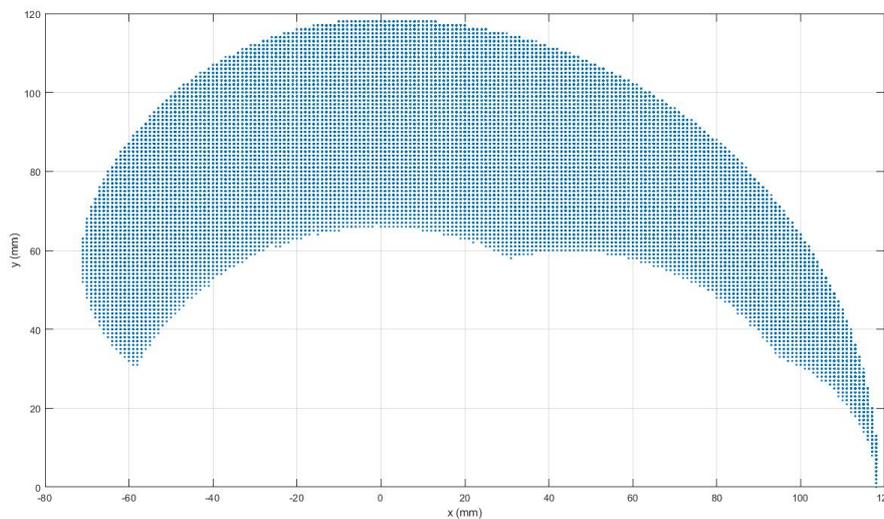


FIGURE 3.9 – Région admissible de l'effecteur du doigt

Notons que, les repères relatifs des phalanges ayant été définis et fixés dans l'assemblage SolidWorks que nous avons importé, nous avons dû réadapter le programme pour avoir un modèle cohérent et qui répond à nos exigences. Notamment l'index et le majeur où le repère lié à la phalange 1 était inversé, et le pouce où la position initiale ne correspondait pas à notre situation de repos voulue.

Nous pouvons maintenant intégrer une boucle de commande pour une meilleure performance du système à chaque entrée θ à l'aide d'un régulateur PID. Pour déterminer les coefficients de ce dernier, nous avons utilisé la toolbox Matlab "PID Tuner", qui, suivant la réponse en boucle fermée du système nous donne les paramètres optimaux du régulateur. Nous pouvons voir sur la figure 3.10 que la sortie après l'ajout du régulateur PID se confond avec l'entrée.

$$\begin{aligned} P &= 0.8221, \\ I &= 622.4789, \\ D &= 0. \end{aligned}$$

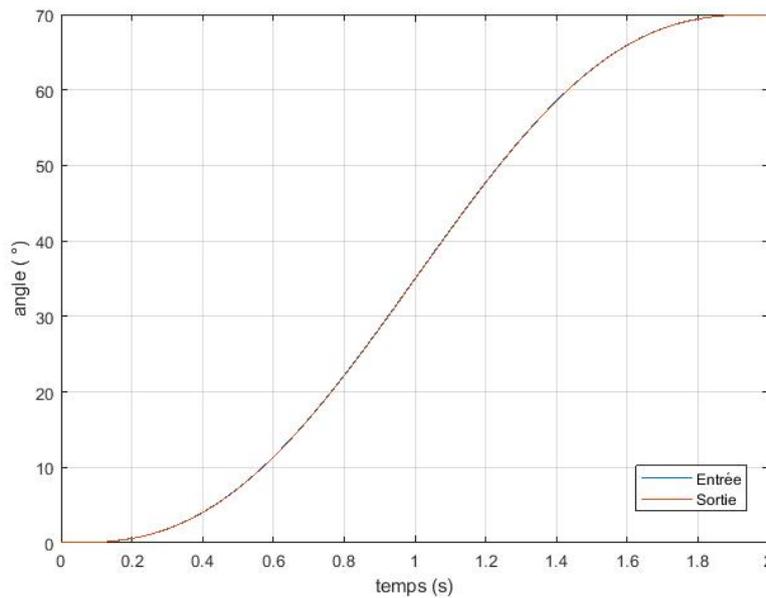


FIGURE 3.10 – Réponse après ajout du P.I.D au système

En intégrant convenablement toutes ces équations dans notre programme Simulink nous obtenons le mouvement d'un seul doigt. En dupliquant alors pour les 5 doigts (voir Fig.3.13), nous obtenons le mouvement global comme montré dans la figure 3.11 et la trajectoire (voir Fig.3.12) qui est identique pour tous les doigts.

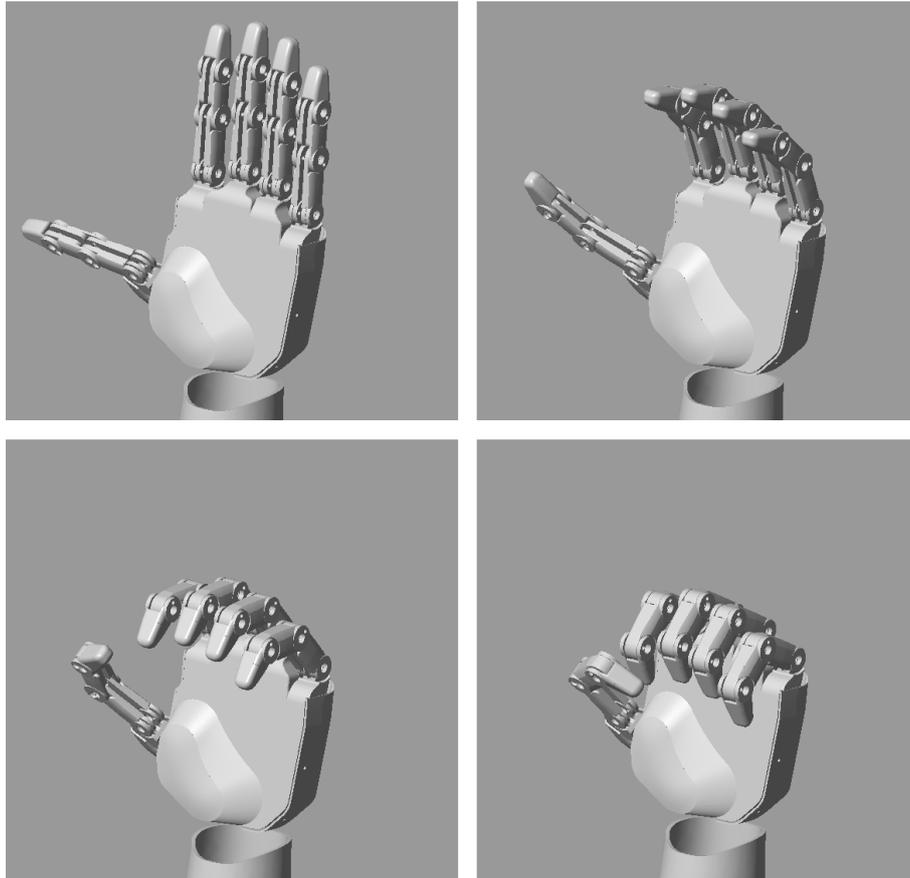


FIGURE 3.11 – Mouvement de la main

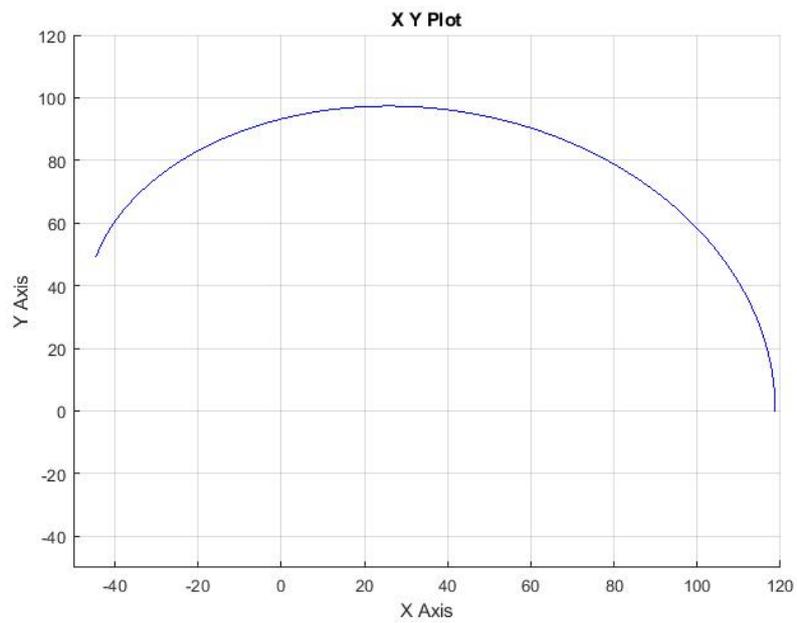


FIGURE 3.12 – Trajectoire de l'effecteur final des doigts

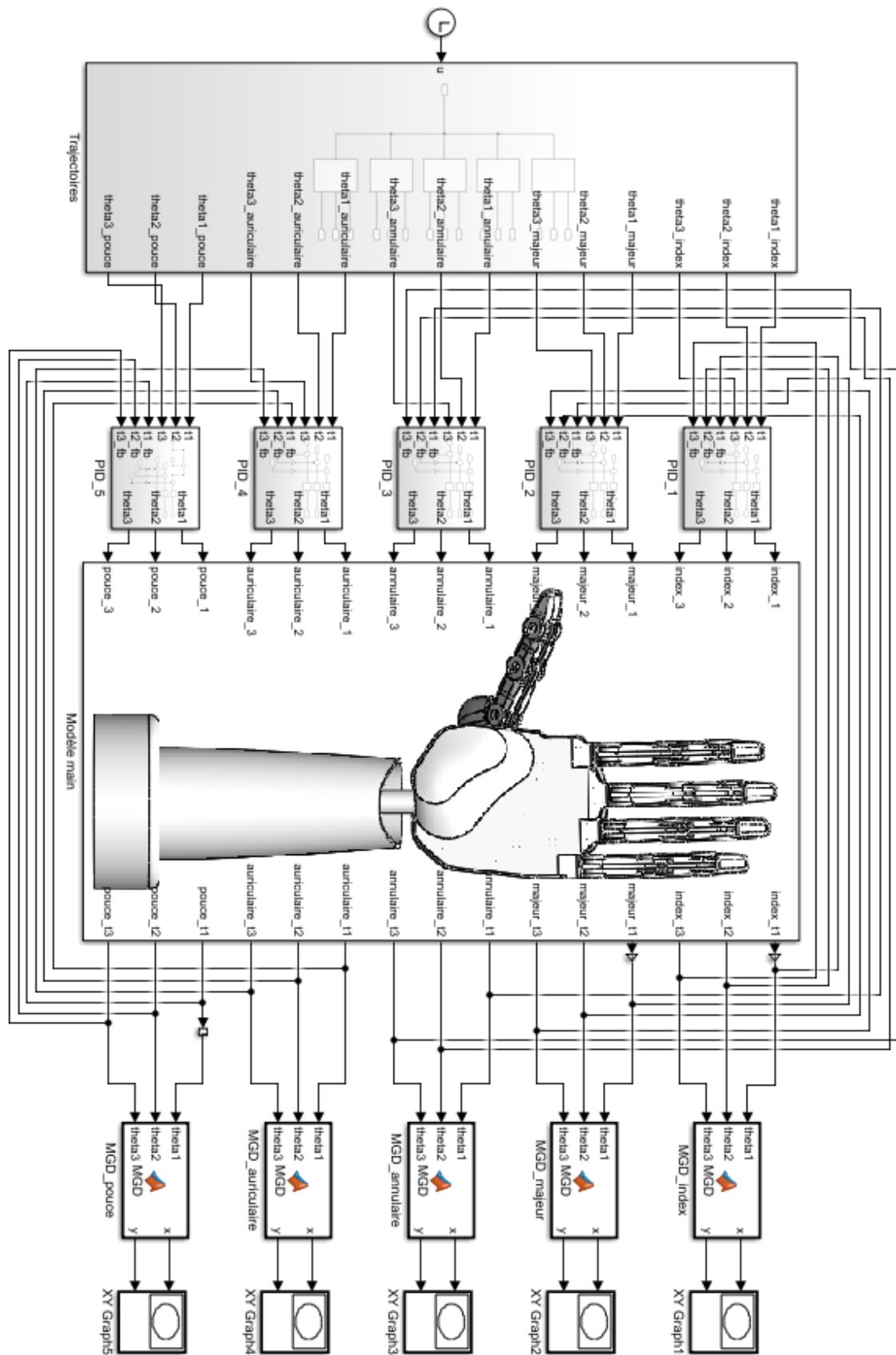


FIGURE 3.13 – Schéma Simulink global

3.3.2 Approche par variables opérationnelles

Dans le sous-paragraphe 3.3.1 nous nous sommes intéressés à la manière de générer une trajectoire à partir des variables articulaires, et nous avons tracé la trajectoire sur le plan (x, y) à l'aide du modèle géométrique direct. Dans ce paragraphe nous tentons de concevoir une entrée pour le système à l'aide des variables opérationnelles de notre système.

Les variables opérationnelles de notre modèle sont définies comme étant les coordonnées x et y de l'effecteur final P (voir Fig.3.8). Pour un point ayant des coordonnées (x, y) et une orientation donnée, nous devons obtenir les valeurs des angles entre chaque articulation pour atteindre ce point. Nous avons alors calculé le MGI (abréviation de : modèle géométrique inverse), qui représente un modèle de transformation entre l'espace articulaire et l'espace opérationnel. Nous spécifions la position et l'orientation de l'effecteur à l'aide de p_x , p_y et Φ l'angle avec l'axe x_0 (voir Fig.3.15). Une première équation est alors :

$$\Phi = \theta_1 + \theta_2 + \theta_3 \quad (3.8)$$

Dans ce qui suit, nous noterons $\cos(\theta_i)$ et $\sin(\theta_i)$ respectivement par c_i et s_i , et $\cos(\Phi)$ et $\sin(\Phi)$ respectivement par c_Φ et s_Φ .

Soit W l'origine du repère 2 comme sur la figure 3.14 ; les coordonnées de W sont

$$\begin{aligned} p_{Wx} &= p_x - a_3 c_\Phi \\ &= a_1 c_1 + a_2 c_{12} \end{aligned} \quad (3.9)$$

$$\begin{aligned} p_{Wy} &= p_y - a_3 s_\Phi \\ &= a_1 s_1 + a_2 s_{12} \end{aligned} \quad (3.10)$$

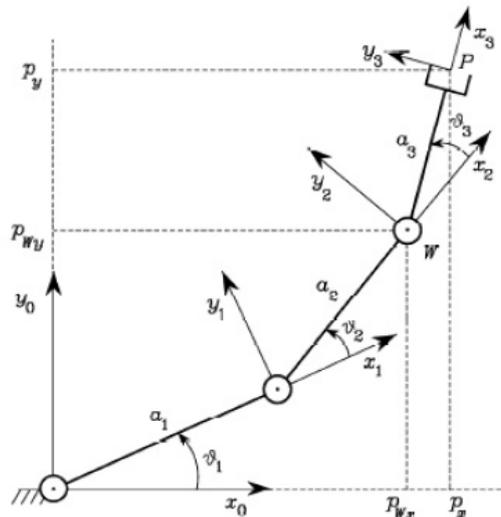


FIGURE 3.14 – Schéma des articulations

Les équations (3.9) et (3.10) ne dépendent que de θ_1 et θ_2 , nous pouvons déduire

$$p_{Wx}^2 + p_{Wy}^2 = a_1^2 + a_2^2 + 2a_1a_2c_2$$

$$c_2 = \frac{p_{Wx}^2 + p_{Wy}^2 - a_1^2 - a_2^2}{2a_1a_2}$$

Nous savons que $-1 \leq c \leq 1$ alors

$$s_2 = \pm \sqrt{1 - c_2^2}$$

La solution positive correspond au coude haut et la solution négative correspond au coude bas (voir Fig.3.15).

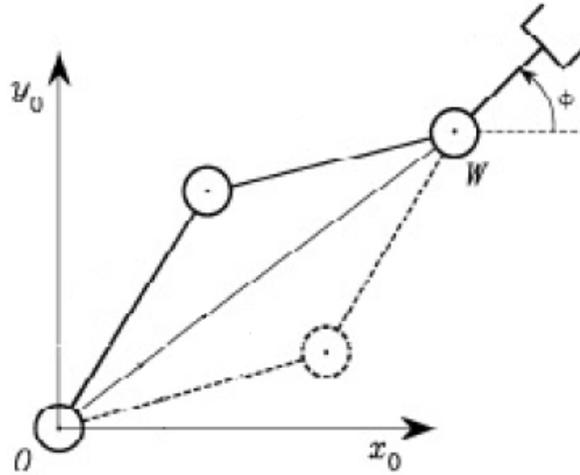


FIGURE 3.15 – Configuration coude haut et coude bas

Nous avons donc

$$\theta_2 = \text{Atan}(s_2, c_2)$$

Nous pouvons déduire alors

$$s_1 = \frac{(a_1 + a_2c_2)p_{Wy} - a_2s_2p_{Wx}}{p_{Wx}^2 + p_{Wy}^2}$$

$$c_1 = \frac{(a_1 + a_2c_2)p_{Wx} + a_2s_2p_{Wy}}{p_{Wx}^2 + p_{Wy}^2}$$

d'où

$$\theta_1 = \text{Atan2}(s_1, c_1)$$

$$\theta_3 = \Phi - \theta_1 - \theta_2$$

La création de trajectoires à partir des variables opérationnelles nécessite la connaissance en tout point de l'orientation et la valeur de Φ . Or, il est théoriquement impossible d'obtenir une relation pour le calcul de Φ , puisqu'aucune ne relie l'orientation aux variables x et y sans passer par les variables articulaires θ_i , et se pose également,

le problème de l'existence de solutions multiples pour une seule et même orientation. Nous pouvons tester pour des points (x, y) présents dans la région admissible (présentée plus haut (voir Fig. 3.9)) qui ont une orientation Φ connue, comme la position de repos où nous savons que les $\theta_i = 0$ (ou bien quand $\Phi = \theta_3$ nous savons que $x = 118.75 \times \cos(\Phi)$ et $y = 118.75 \times \sin(\Phi)$) et voir que le MGI donne de bons résultats (voir Fig. 3.16), mais étant donné que les entrées représentent des constantes (voir le schéma Simulink Fig. 3.17), Il reste très compliqué d'obtenir un mouvement continu de la trajectoire de chaque doigt, au lieu de ça, le mouvement sera saccadé, chaque point correspondant à un angle Φ , ce dernier très limité en nombre. Pour définir l'orientation Φ de n'importe quel point présent dans la région admissible, nous sommes tenus d'utiliser les θ_i , mais une telle approche serait totalement incohérente étant donné que θ_3 se calcule lui, à l'aide de la relation (3.8), et donc, en utilisant Φ . Face à cette contradiction qui ne permet pas de résoudre aisément le problème, une solution s'offre à nous : *l'apprentissage*.

Un apprentissage a pour but de créer une base de données au système : il faudrait calculer chaque Φ à partir du modèle géométrique direct (voir sous-paragraphe 3.3.1), correspondant à chaque variation de θ_1 , θ_2 et θ_3 , pour ensuite, associer les résultats à des coordonnées x et y .

Cette approche nous aurait permis par la suite, d'utiliser notre base de données en entrée, afin de faire varier Φ en fonction des coordonnées (x, y) du point de l'effecteur, et obtenir les valeurs des θ_i en sortie.

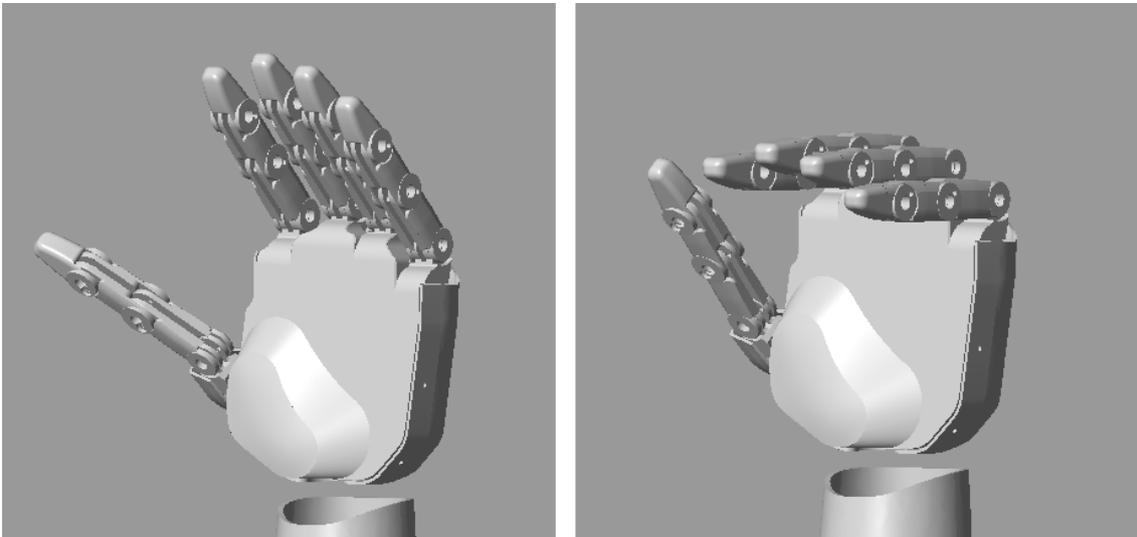


FIGURE 3.16 – Contrôle de position par MGI (pour $\Phi = 45^\circ$ à gauche, et $\Phi = 90^\circ$ à droite)

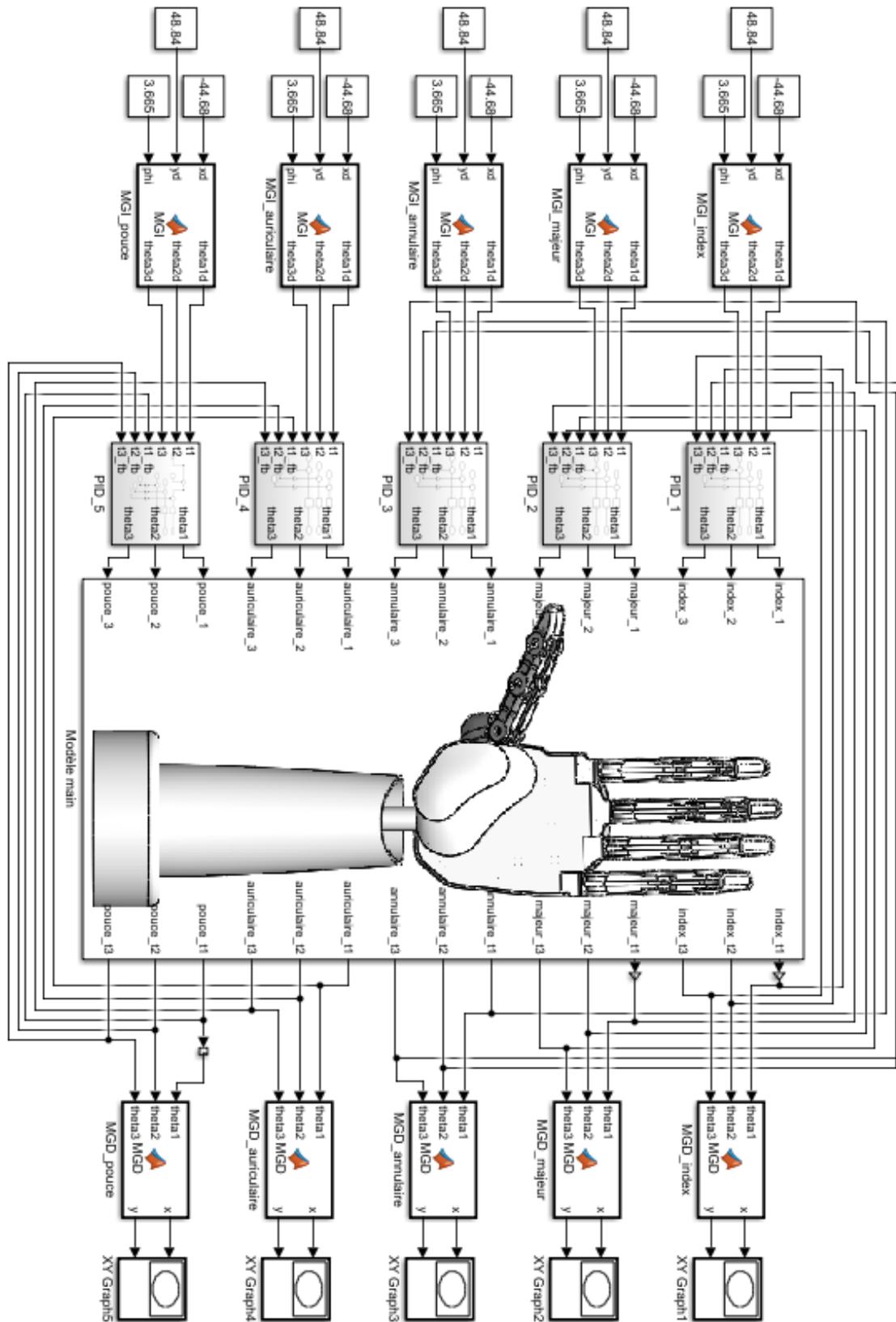


FIGURE 3.17 – Schéma Simulink global avec modèle géométrique inverse

3.4 Conclusion

Dans ce chapitre, nous avons, dans un premier temps mis en place le programme permettant le mouvement des doigts, et en un second temps, nous avons étudié la trajectoire de notre système par deux approches différentes. L'approche par variables opérationnelles, bien que très intéressante à étudier, nous aurait demandé beaucoup de temps, et un matériel de calcul très performant et très puissant. N'ayant ni la première, ni la seconde ressource de manière efficiente, nous avons dû laisser tomber cette approche. Dans ce qui va suivre, nous avons préféré continuer et pousser un peu plus notre étude avec la première approche, qui représente le moyen le plus simple pour se représenter une trajectoire.

CHAPITRE

4

PRÉHENSION ET DÉTECTION DE CORPS

4.1 Introduction

Dans le but d’approfondir notre étude, nous avons intégré dans ce chapitre la notion de “Grasping”, très importante dans le domaine de la robotique et notamment pour les bras manipulateurs. Le “Grasping”, du verbe “to grasp” qui signifie “saisir”, est l’action qui consiste à saisir et prendre des objets. Il existe plusieurs approches pour effectuer l’action de saisir un objet dans l’espace : à l’aide d’une caméra qui analyse l’objet à prendre, à l’aide de capteurs ultrasons qui permettent de calculer la distance entre l’effecteur et l’objet en question, ou bien à l’aide de capteurs de force qui mesurent la force appliquée à l’effecteur qui devient différente de zéro au contact de l’objet.

4.2 Simulation

Dans nos simulations, pour une trajectoire fixe (fixée dans le paragraphe 3.3.1), nous avons créé un capteur virtuel qui nous permet d’avoir en temps réel la longueur de la projection du vecteur de distance entre le repère lié à l’objet à saisir et celui lié à l’effecteur final des doigts, dans le plan $X - Y$ du repère de mesure. Le repère lié à l’objet étant relié à son centre de gravité, nous avons besoin de connaître les dimensions et la forme de l’objet à tenir (le diamètre du cercle pour un cylindre, la longueur d’un coté du carré pour un cube).

L’objet est ajouté à notre programme Simulink et est relié comme montré sur le haut de la figure 4.1. À travers le bloc “Rigid Transform” nous pouvons le déplacer

dans l'espace et le mettre ainsi à l'endroit adéquat pour tester les simulations.

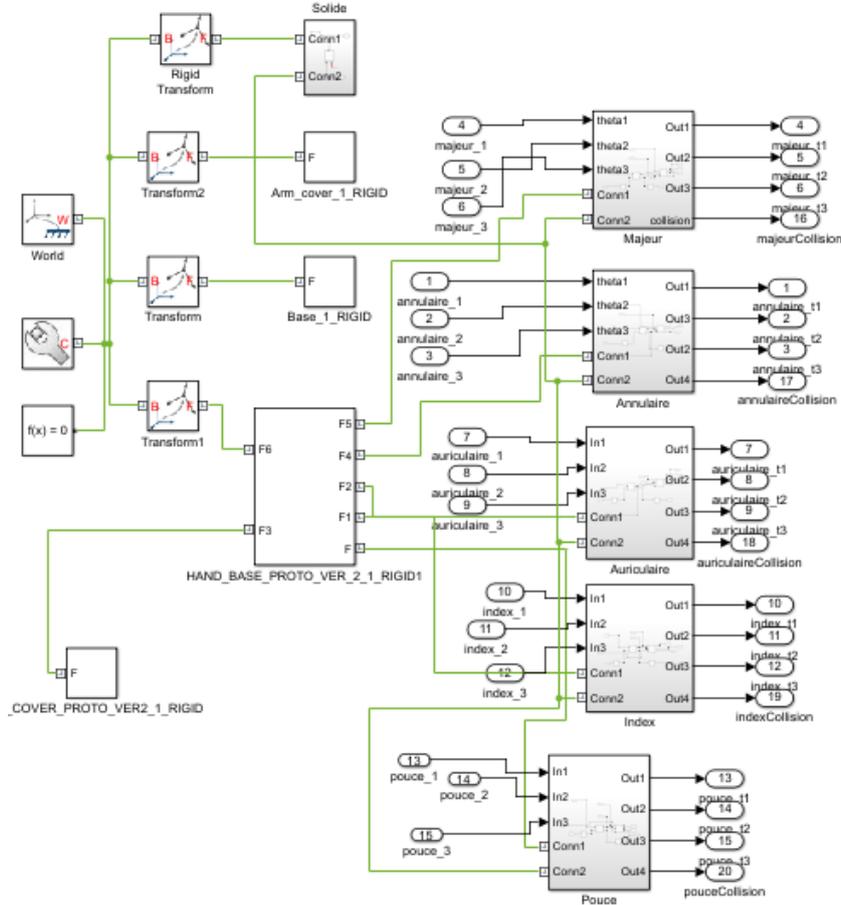


FIGURE 4.1 – Ajout du solide au système

Pour avoir la distance entre le repère associé à l'effecteur du doigt (voir Fig.4.4) et celui associé à l'objet, nous avons utilisé le bloc "Rigid transform sensor" (voir Fig.4.3) auquel sont liés les deux repères. Ce bloc nous retourne la valeur de la projection du vecteur de distance dans le plan $X - Y$ du repère de mesure en sortie. De cette valeur, selon la forme de l'objet à saisir, est soustraite la valeur du rayon pour un cylindre (ou de la demi longueur d'un côté pour un cube) pour avoir la distance qui sépare l'effecteur de la surface de l'objet. De cette manière, nous pouvons visualiser la variation de la distance et savoir que le point d'impact correspond au moment où la distance entre les corps est nulle. Il suffira alors de récupérer ces données et de les injecter en entrée sur une condition *if* qui désactive l'entrée quand la distance est inférieure ou égale à zéro (voir Fig.4.2).

Par ailleurs, relier la sortie à l'entrée provoque une erreur, cela crée une boucle fermée bloquée. Pour palier cette erreur, nous avons appliqué un léger délai de l'ordre de $0.01s$ à l'information de retour, ce qui peut provoquer en conséquence de légères imperfections. Dans un cas réel, nous pouvons à partir de la condition *if*, laisser une légère marge et désactiver l'entrée pour une valeur de la distance légèrement inférieure à zéro pour avoir vraiment l'action de préhension sur l'objet. Notons que, par valeur négative de la distance nous exprimons la distance dans la direction opposée.

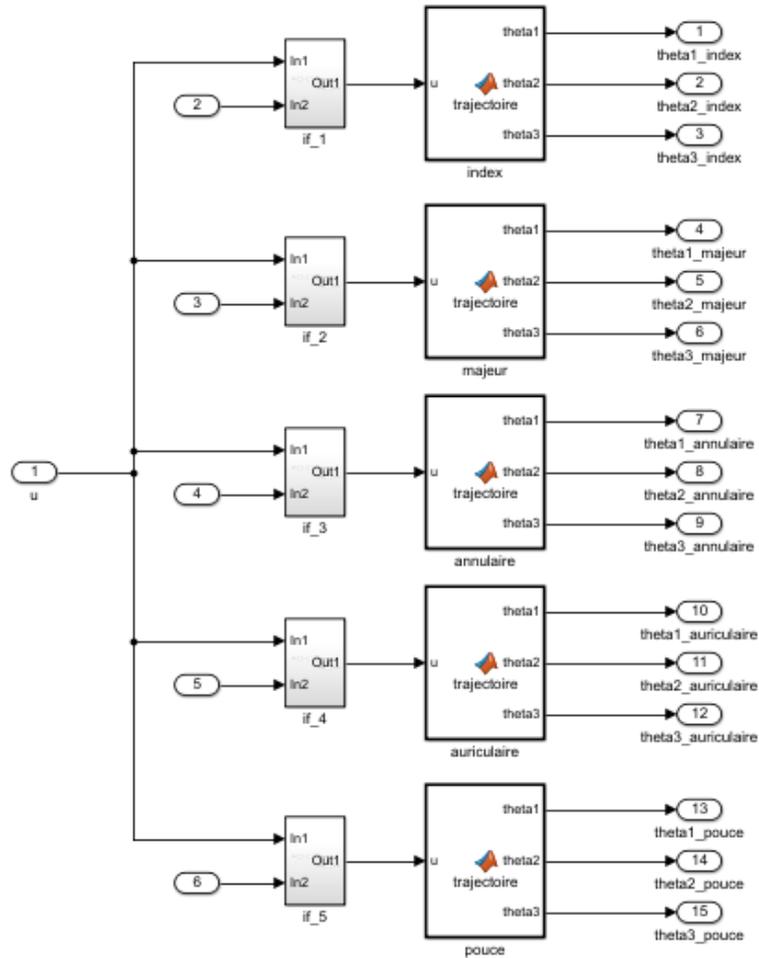
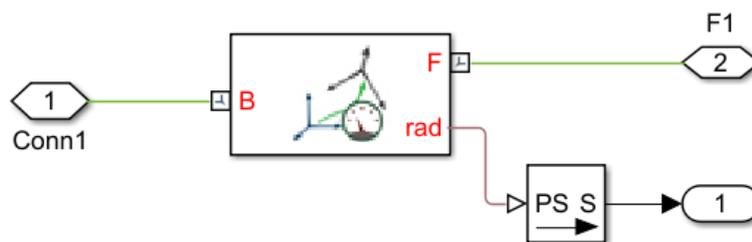
FIGURE 4.2 – Les conditions *if* en entrée

FIGURE 4.3 – Bloc “Rigid transform Sensor”

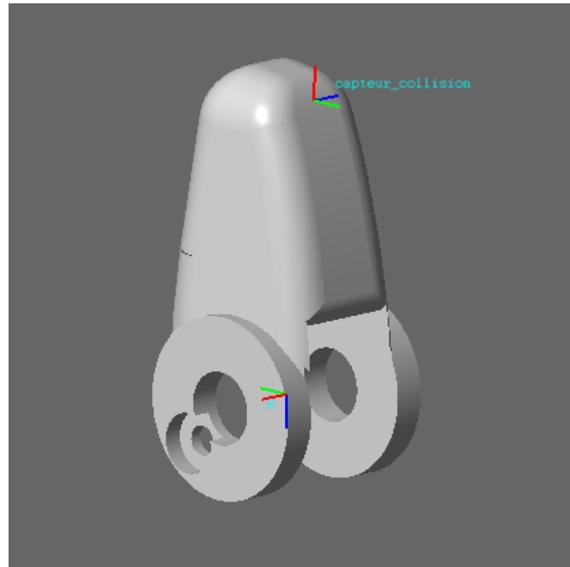


FIGURE 4.4 – Repère lié au capteur

En effectuant les changements nécessaires dans le programme Simulink sur la figure 3.13 et en ajoutant toutes les conditions *if* nous obtenons le programme présenté sur la figure 4.5.

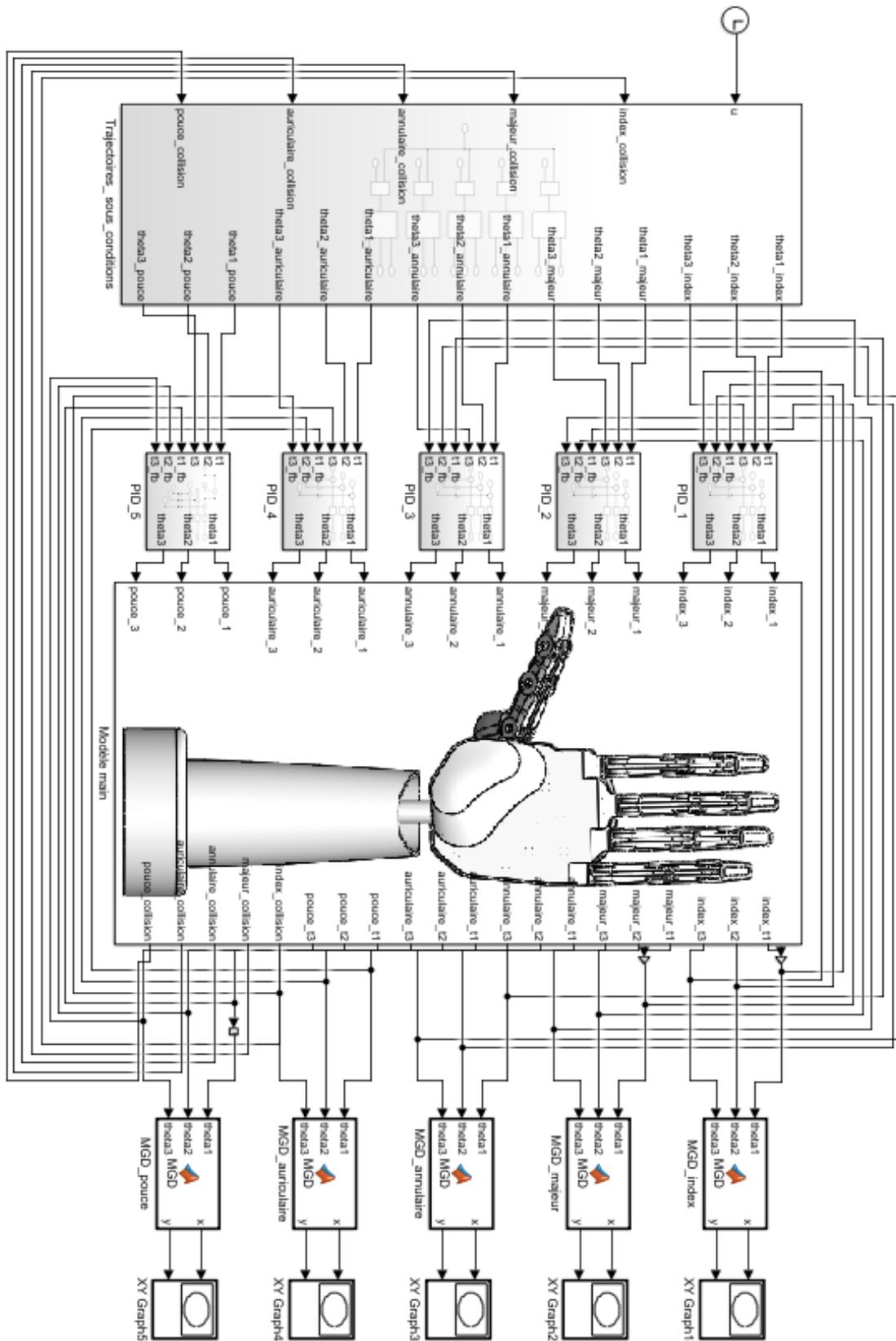


FIGURE 4.5 – Schéma Simulink global avec détection de collisions

4.2.1 Préhension d'un cube

Comme premier test, nous avons pris un cube de $100 \times 100 \times 100mm$ pour sa géométrie simple comme sur la figure 4.6.

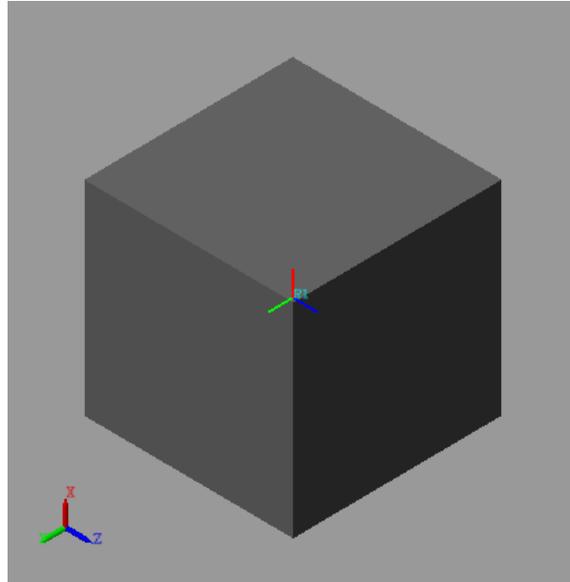


FIGURE 4.6 – Cube $100 \times 100 \times 100mm$

En opérant à des changements dans le programme pour spécifier la forme, nous obtenons le résultat montré dans la figure 4.7.

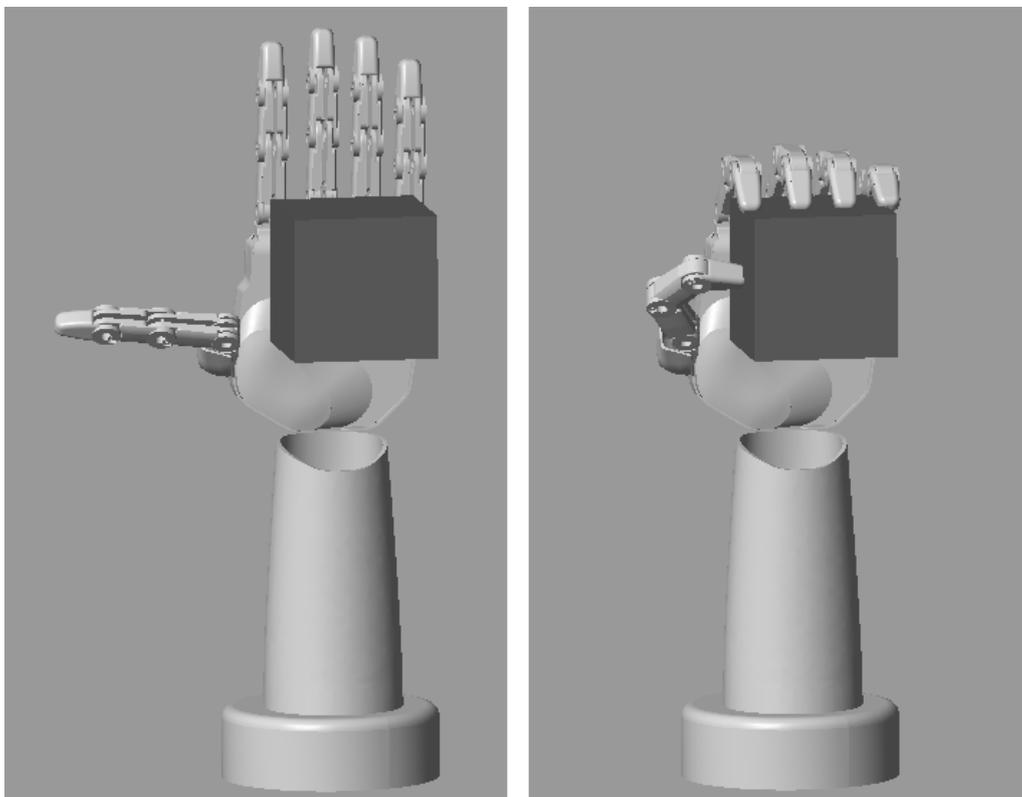


FIGURE 4.7 – Préhension d'un cube

Nous pouvons observer l'ensemble des trajectoires pour chaque doigt dans la figure 4.8.

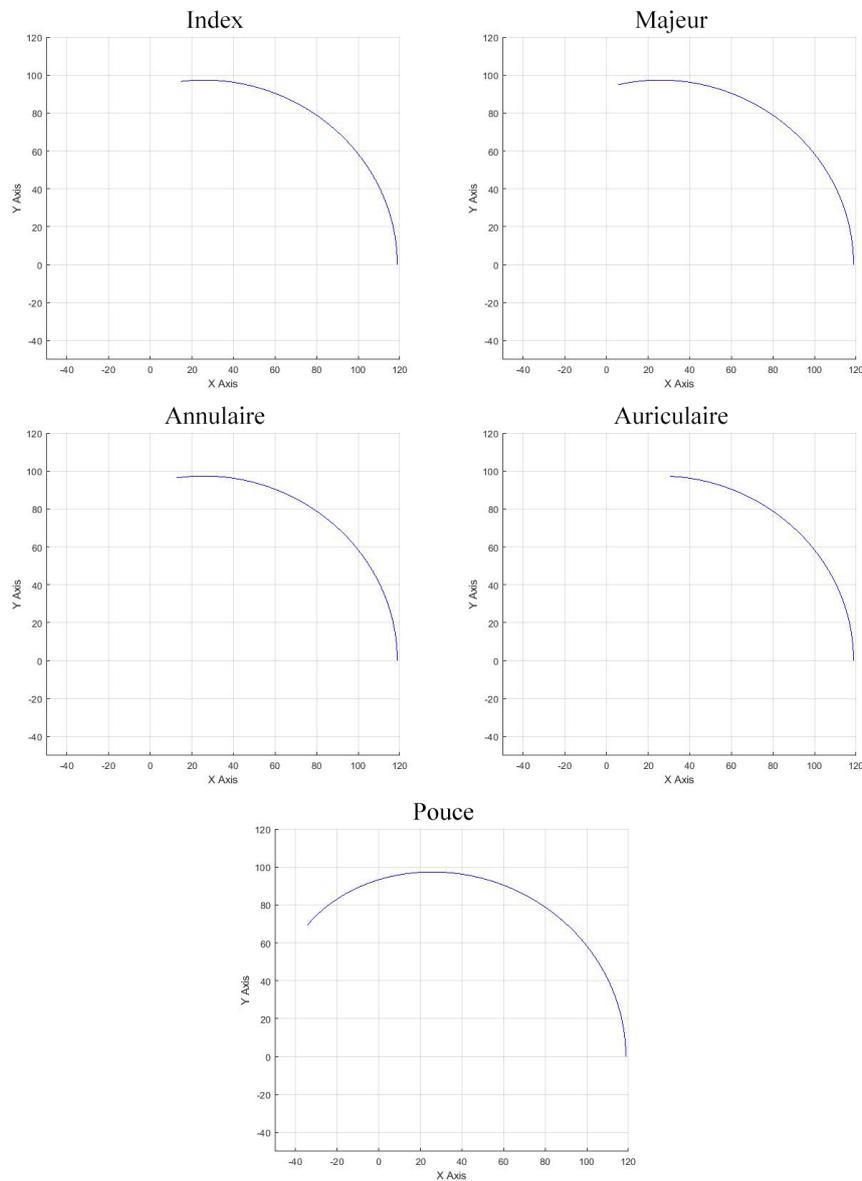


FIGURE 4.8 – Trajectoires des effecteurs finaux des doigts

4.2.2 Préhension d'un cylindre

Pour une forme un peu plus complexe, nous avons testé des cylindres de tailles et d'orientations différentes. La première simulation a été effectuée sur le cylindre montré dans la figure 4.9 de 70mm de rayon et de 220mm de long.

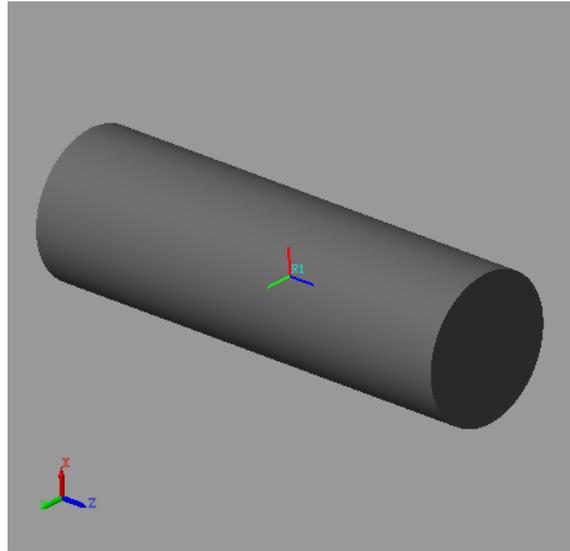


FIGURE 4.9 – Cylindre

Étant donné que la mesure de la distance ne se fait qu'entre l'effecteur final du doigt présent dans la troisième phalange (voir Fig.4.4) et l'objet à saisir, le mouvement du doigt n'est arrêté que lorsque cette distance précisément est nulle. Or, dans ce cas ci, nous nous sommes confrontés à un problème de collision des autres phalanges avec l'objet à prendre. Nous pouvons voir dans la figure 4.10 que dans certains cas, notamment quand le cylindre est placé trop haut pour la trajectoire de base fixée, quand l'extrémité de la phalange trois est au contact de l'objet, une partie des phalanges traverse le cylindre.

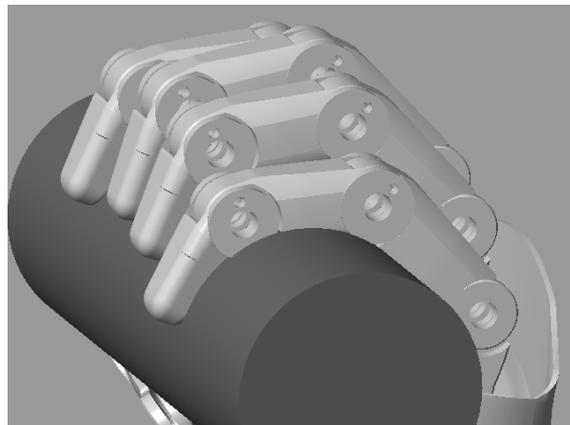


FIGURE 4.10 – Problème collision

Pour remédier à ce problème, nous pouvons définir un ou plusieurs capteurs dans chaque phalange (voir Fig.4.11) pour calculer la distance entre l'objet en question avec toutes les phalanges. Dans ce cas, nous agissons non pas sur le mouvement global du doigt mais sur chaque angle θ_1 , θ_2 et θ_3 entre les différentes phalanges (voir Fig.3.8). Ceci nous mène à des changements dans le schéma Simulink (voir Fig.4.5) et au lieu d'appliquer les conditions *if* à l'entrée du système nous les appliquons à l'entrée de chacun des trois angles θ_1 , θ_2 et θ_3 (voir Fig. 4.17). Cette approche est très efficace mais très lourde et très gourmande en ressources (voir schéma Simulink 4.18). Pour cela, nous n'avons pas pu l'appliquer à chaque doigt en raison de la

faible puissance de nos machines. Dans les figures 4.12, 4.14 et 4.16 qui vont suivre, nous n'avons appliqué cette approche que sur le pouce qui en nécessitait vraiment pour la fidélité de la préhension.

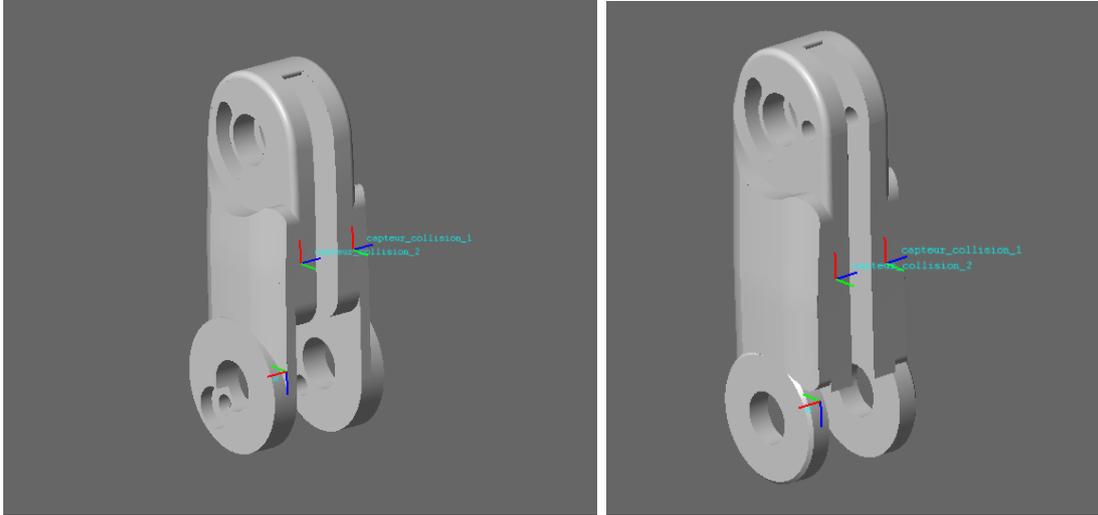


FIGURE 4.11 – Emplacement des capteurs sur la phalange 1 et 2

Pour un cylindre droit nous pouvons voir alors l'action de la préhension dans la figure 4.12 ainsi que les trajectoires de l'extrémité des doigts dans la figure 4.13.

Pour ce même cylindre, mais incliné, nous pouvons voir sur les figures 4.14 et 4.15 l'action de la préhension et les trajectoires respectivement.

Pour la préhension d'une barre, nous pouvons voir le résultat de l'exécution dans la figure 4.16.

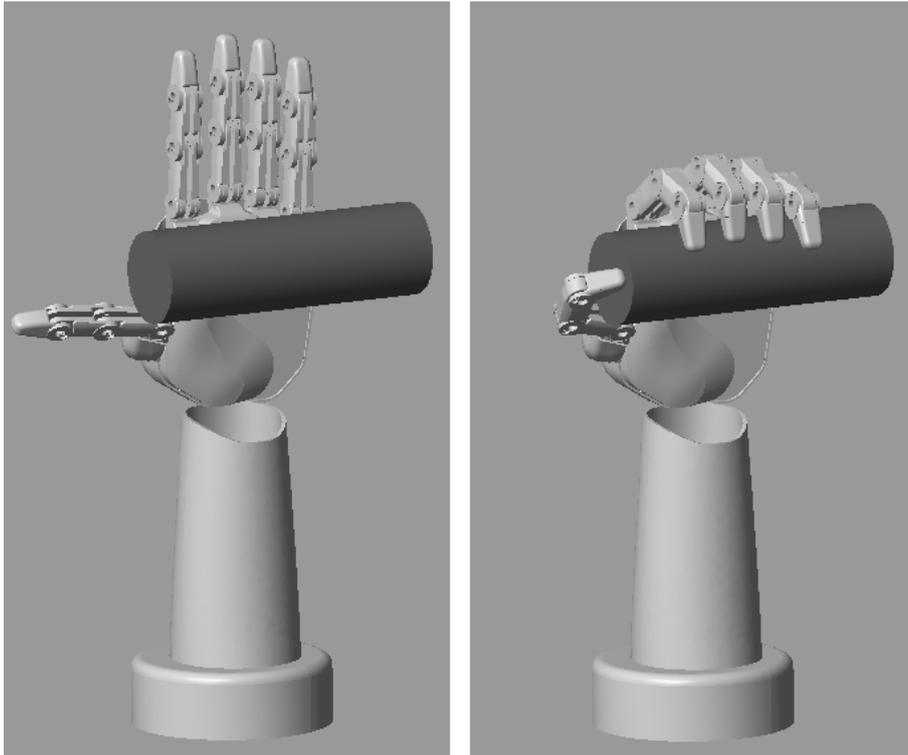


FIGURE 4.12 – Préhension d'un cylindre

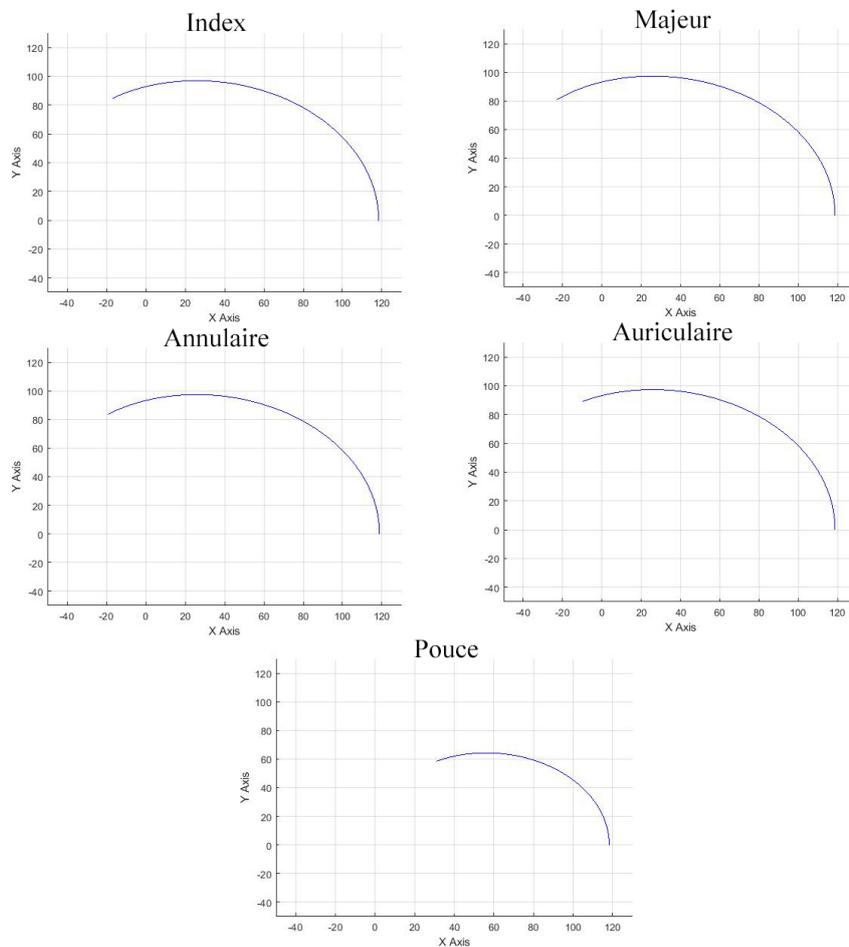


FIGURE 4.13 – Trajectoires des effecteurs finaux des doigts

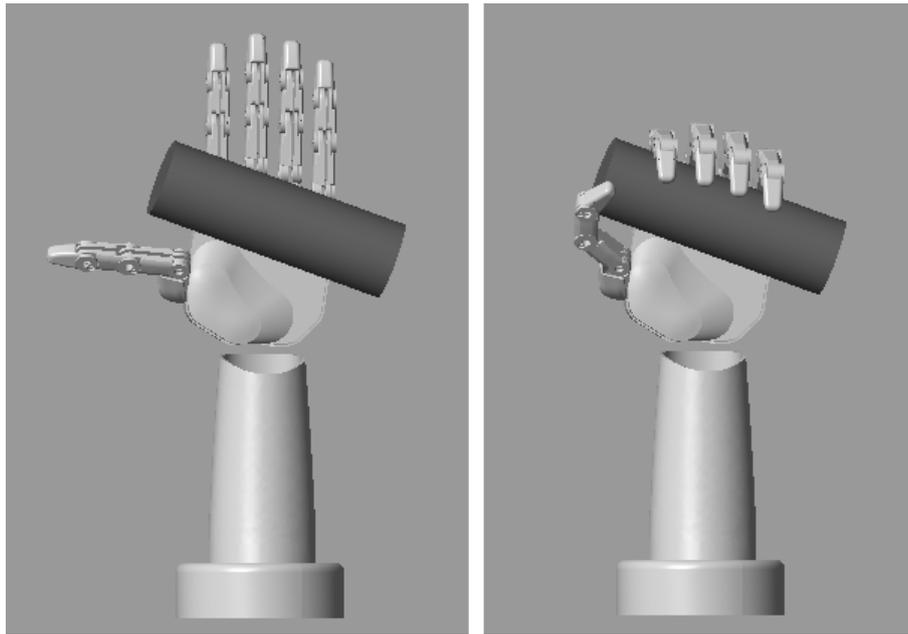


FIGURE 4.14 – Préhension d'un cylindre penché

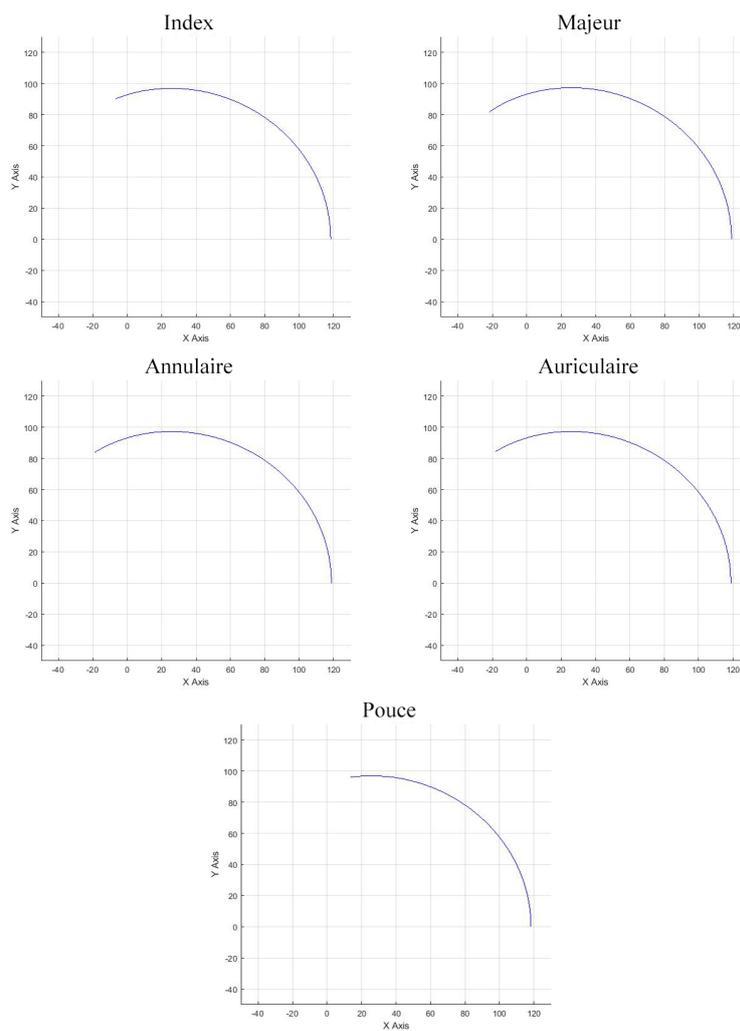


FIGURE 4.15 – Trajectoires des effecteurs finaux des doigts

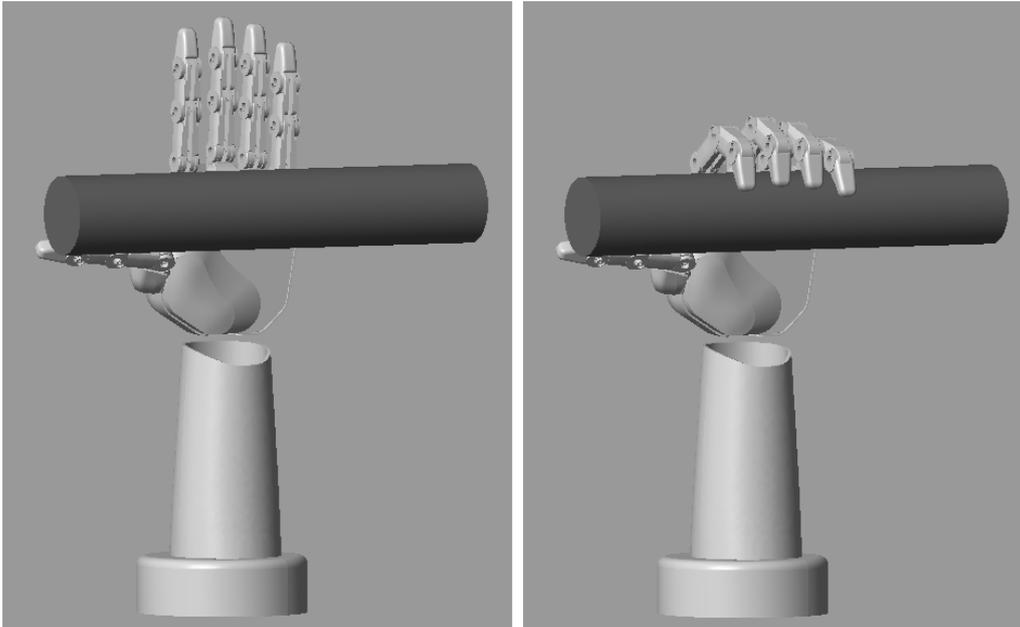
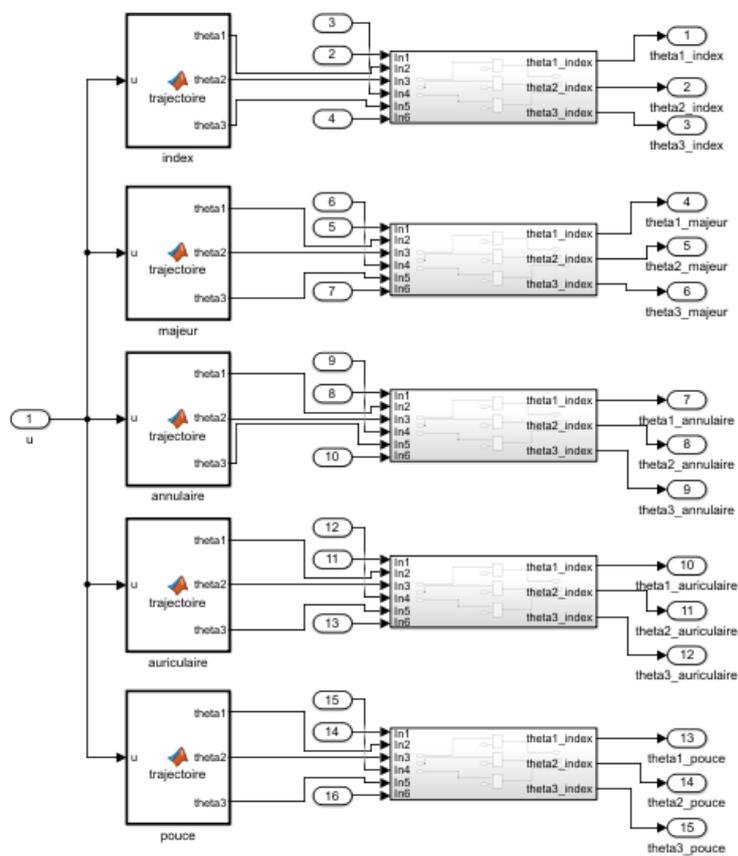


FIGURE 4.16 – Préhension d'une barre

FIGURE 4.17 – Les conditions *if* appliquées sur chaque θ

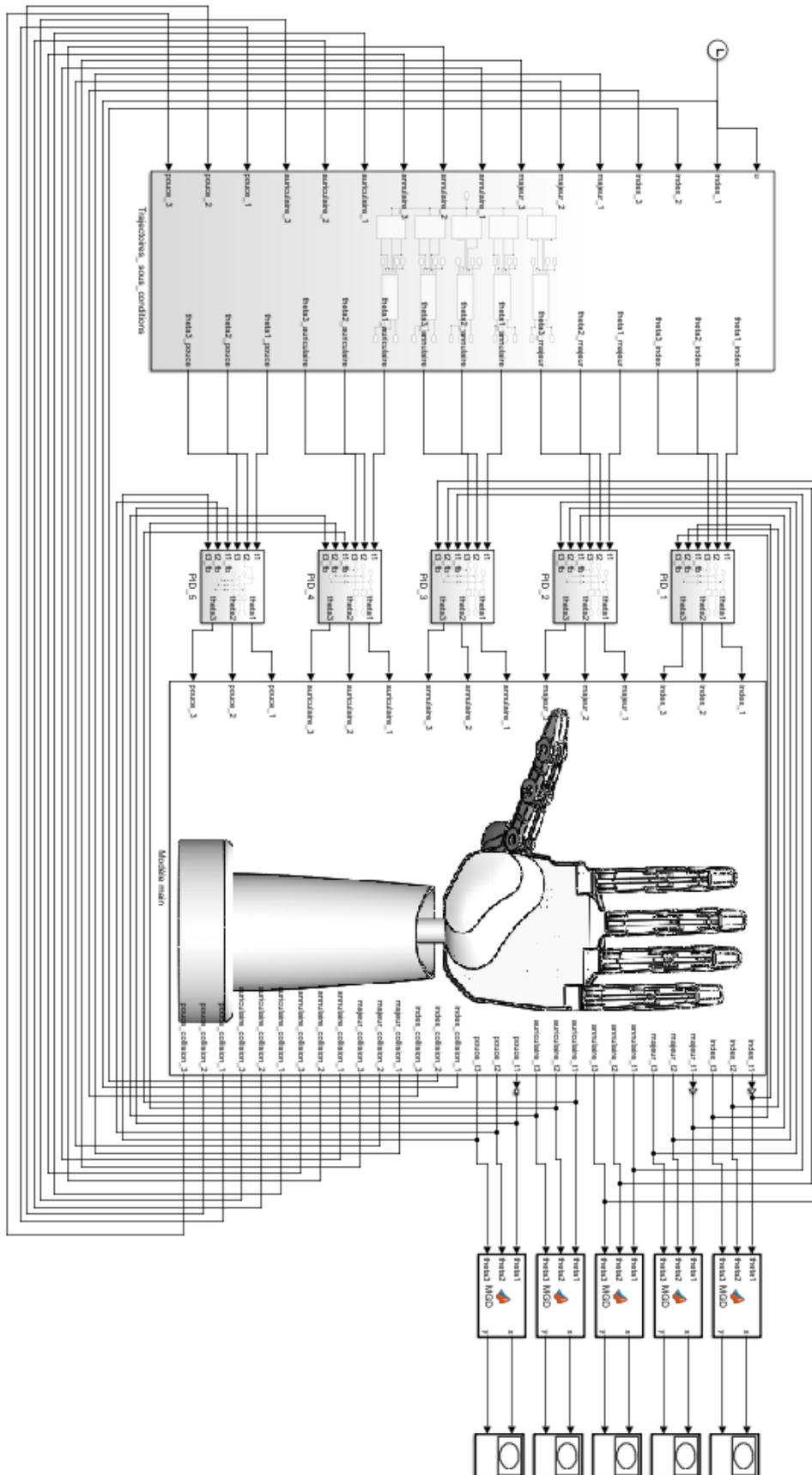


FIGURE 4.18 – Schéma Simulink global avec multiple détection de collisions par doigt

4.3 Conclusion

Nous avons traité dans ce chapitre la question de la préhension d'objets, aspect qui représente un défi dans le domaine de la robotique. Nous avons commencé par élaborer un programme qui permet la détection de collisions entre la main et l'objet à saisir qui doit être de forme connue en calculant de façon dynamique la distance qui sépare les deux corps, puis nous avons effectué des simulations de préhension sur diverses formes simples. Nous pouvons noter à travers notre étude que l'action de préhension est très complexe à réaliser et nécessite une puissance et capacité de calculs conséquentes pour une bonne performance.

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons exploré les travaux menés ayant pour objectif la simulation du membre supérieur : la main. Le premier volet est consacré à l'étude de la trajectoire du doigt en utilisant le modèle géométrique direct. Nous avons montré que ce modèle est capable de générer les postures de flexion et d'extension (une seule étude est suffisante étant donné la similitude des caractéristiques entre chaque doigt). Pour ce faire, nous avons défini des contraintes pour chaque phalange, afin d'être le plus fidèle possible au mouvement du doigt humain et éviter toute collision entre eux.

Pour de meilleures performances, nous avons ajouté une boucle de régulation au système pour obtenir un retour adéquat sur les valeurs en sortie et annuler l'erreur statique. En définitif, ce volet nous a permis d'obtenir les coordonnées cartésiennes dans l'espace de l'effecteur de chaque doigt et une meilleure performance au niveau de la génération de leurs trajectoires.

Nous avons tenté par ailleurs une approche par modèle géométrique inverse pour générer une trajectoire mais une telle approche aurait nécessité une étude supplémentaire plus approfondie sur les réseaux de neurones et sur le traitement de bases de données, ce qui aurait été compliqué par manque de temps et de ressources matérielles.

Un second volet a été dédié à l'exécution d'une préhension de la main. Pour ce faire, lors des expérimentations réalisées, nous avons choisi l'approche qui configure un calcul de distance entre l'effecteur de la troisième phalange et la surface d'un objet de forme et de dimensions connues. Cette approche se définit par la mise en place d'un capteur virtuel, qui réalise la mesure en temps réel de la distance entre le repère lié à l'effecteur et celui lié au centre de gravité de l'objet.

Le manque cruel de temps et de ressources par moment a fait que le résultat de ce projet ne rejoignit pas certains des objectifs que l'on s'était fixés au départ. C'est la raison pour laquelle nous avons mis en place une liste de potentielles idées à développer plus amplement à l'avenir :

- Exploiter le modèle géométrique inverse, en utilisant une approche qui se base

sur l'apprentissage à base de réseaux de neurones et l'intégration d'une base de données des mouvements de préhension.

- Remédier autant que possible aux contraintes liées au manque de ressources matérielles, en l'occurrence nos ordinateurs, ce qui nous permettrait d'effectuer des simulations complexes et d'exécuter des programmes plus lourds et plus performants.
- Utiliser des capteurs de collisions au niveau des phalanges, ce qui aurait donné un résultat d'adaptabilité de la préhension intéressant et nous aurait évité certains ajustements que l'on a dû appliquer. Nous avons essayé de l'appliquer à travers un outil de Simulink **Virtual Reality**, mais le manque de temps ne nous permit pas de développer correctement l'idée.
- Utiliser d'autres logiciels, Matlab étant un logiciel assez complexe à manipuler pour un projet de préhension. Cela nous permettrait de simuler plus facilement cette action. Nous pouvons citer le logiciel **GraspIt!** [13], qui propose un large panel de mains robotiques, ainsi que la possibilité de le coupler au logiciel Matlab. La simplicité de son utilisation et son interface intuitive le rendent plus avantageux.

ANNEXE

A

PROGRAMME MATLAB 1

Ce code Matlab permet de tracer la région admissible de l'effecteur final des doigts.

```
1 a1 = 45;  
2 a2 = 38;  
3 a3 = 35.75;  
4  
5 t1 = linspace(0,90,90)*pi/180;  
6 t2 = linspace(0,70,70)*pi/180;  
7 t3 = linspace(0,70,70)*pi/180;  
8  
9 [T1,T2,T3] = ndgrid(t1,t2,t3);  
10  
11 xM = round(a1*cos(T1)+a2*cos(T1+T2)+a3*cos(T1+T2+T3));  
12 yM = round(a1*sin(T1)+a2*sin(T1+T2)+a3*sin(T1+T2+T3));  
13  
14 plot(xM(:),yM(:),' . ' )  
15 grid on;
```

ANNEXE

B

PROGRAMME MATLAB 2

Ce programme représente la fonction Matlab qui génère la trajectoire de l'effecteur final du doigt par variations des angles θ_1 , θ_2 et θ_3

```
1 function [theta1, theta2, theta3] = trajectoire(u)
2 tf = 2;
3 D = 70;
4
5 r=10*(u/tf).^3-15*(u/2).^4+6*(u/2).^5;
6
7 theta1=r*D;
8
9 theta2=r*D;
10
11 theta3=r*D;
```

ANNEXE

C

PROGRAMME MATLAB 3

Ce programme nous permet de calculer le modèle géométrique direct des doigts.

```
1  clc, clear all, close all;
2
3  % a1 = 45;
4  % a2 = 38;
5  % a3 = 35.75;
6
7  syms theta1 theta2 theta3
8  syms a1 a2 a3
9
10 A_0_1 = [cos(theta1) -sin(theta1) 0 a1*cos(theta1);
11          sin(theta1) cos(theta1) 0 a1*sin(theta1)
12          ;
13          0 0 1 0;
14          0 0 0 1];
15 A_1_2 = [cos(theta2) -sin(theta2) 0 a2*cos(theta2);
16          sin(theta2) cos(theta2) 0 a2*sin(theta2)
17          ;
18          0 0 1 0;
19          0 0 0 1];
20 A_2_3 = [cos(theta3) -sin(theta3) 0 a3*cos(theta3);
21          sin(theta3) cos(theta3) 0 a3*sin(theta3)
22          ;
23          0 0 1 0;
24          0 0 0 1];
```

25 | $T = A_{0_1} * A_{1_2} * A_{2_3};$

ANNEXE

D

PROGRAMME MATLAB 4

Cette fonction Matlab présente dans le programme Simulink nous permet de calculer la position (x, y) de l'effecteur des doigts à partir du modèle géométrique direct.

```
1 function [x,y] = MGD(theta1, theta2, theta3)
2 a1 = 45;
3 a2 = 38;
4 a3 = 35.75;
5
6 x = a1*cos(theta1)+a2*cos(theta1+theta2)+a3*cos(theta1+
   theta2+theta3);
7 y = a1*sin(theta1)+a2*sin(theta1+theta2)+a3*sin(theta1+
   theta2+theta3);
8 end
```

ANNEXE

E

PROGRAMME MATLAB 5

Ce code Matlab calcule les angles θ_1 , θ_2 et θ_3 à partir du modèle géométrique inverse.

```
1 function [theta1d, theta2d, theta3d] = MGI(xd,yd,phi)
2 a1 = 45;
3 a2 = 38;
4 a3 = 35.75;
5
6 pwx = xd - a3*cos(phi);
7 pwy = yd - a3*sin(phi);
8
9 c2 = (pwx^2+pwy^2-a1^2-a2^2)/(2*a1*a2);
10 s2 = sqrt(1-c2^2);
11
12 s1 = ((a1+a2*c2)*pwy-a2*s2*pwx)/(pwx^2+pwy^2);
13 c1 = ((a1+a2*c2)*pwx+a2*s2*pwy)/(pwx^2+pwy^2);
14
15 theta2d = atan2(s2,c2);
16 theta1d = atan2(s1,c1);
17 theta3d = phi - theta1d - theta2d;
18
19 theta1d = rad2deg(theta1d);
20 theta2d = rad2deg(theta2d);
21 theta3d = rad2deg(theta3d);
```

BIBLIOGRAPHIE

- [1] Grab Cad Community robotic hand concept. <https://grabcad.com/library/robotic-hand-concept-1>.
- [2] Simscape Multibody mathworks. <https://www.mathworks.com/products/simscape-multibody.html>.
- [3] George A Bekey, Rajko Tomovic, and Ilija Zeljkovic. Control architecture for the belgrade/usc hand. *Dextrous robot hands*, pages 136–149, 1990.
- [4] L Biagiotti, F Lotti, C Melchiorri, and G Vassura. How far is the human hand. *A review on anthropomorphic robotic end-effectors*, 2004.
- [5] Dudley S Childress. Historical aspects of powered limb prostheses. *Clin Prosthet Orthot*, 9(1) :2–13, 1985.
- [6] Shadow Robot Company. Design of a dextrous hand for advanced clawar applications. Citeseer, 2003.
- [7] Jacques Denavit and Richard S Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. 1955.
- [8] Xavier Daumalin et Philippe Mioche. La désindustrialisation au regard de l’histoire. <http://journals.openedition.org/rives/4471>;DOI:<https://doi.org/10.4000/rives.4471>, mis en ligne le 15 octobre 2014, consulté le 29 mai 2023.
- [9] Martin F Guay. *Analyse cinémato-statique d’un doigt sous-actionné à 3 DDL pour une prothèse de membre supérieur*. PhD thesis, Université Laval, 2012.
- [10] Moussa Haddad, Taha Chettibi, Wisama Khalil, and Halim Lehtihet. Trajectory generation. *Modeling, Performance Analysis and Control of Robot Manipulators*, pages 189–239, 2007.
- [11] Stephen C Jacobsen, Edwin K Iversen, David F Knutti, RT Johnson, and KB Biggers. Design of the utah. MIT Dextrous Hand”, IEEE Conference on Robotics and Automation, 1986.
- [12] Christine L MacKenzie and Thea Iberall. *The grasping hand*. Elsevier, 1994.
- [13] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4) :110–122, 2004.

- [14] Tokuji Okada. Computer control of multijointed finger system for precise object-handling. *IEEE Transactions on Systems, Man, and Cybernetics*, 12(3) :289–299, 1982.
- [15] RN Scott. Myoelectric control of prostheses : A brief history. Myoelectric Symposium, 1992.
- [16] Gerwin Smit. Mechanical evaluation of the “hüfner hand” prosthesis. *Prosthetics and orthotics international*, 2021.
- [17] M Vande Weghe, Matthew Rogers, Michael Weissert, and Yoky Matsuoka. The act hand : Design of the skeletal structure. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 3375–3379. IEEE, 2004.

Résumé

Dans ce travail, nous avons pour objectif de réaliser un jumelage entre le logiciel de modélisation 3D *SolidWorks* et Matlab/Simulink d'un modèle d'une main robotique anthropomorphique actionnée en chaque phalange de chaque doigt. Nous avons alors introduit un historique sur les mains artificielles et une définition de l'action de préhension ainsi que l'approche employée en simulation. Nous avons par la suite décrit le modèle 3D de la main utilisé disponible en Open-Source ainsi que les programmes Simulink responsables de la génération de mouvement des doigts. Nous avons alors appliqué une commande et tracé les courbes des trajectoires, des positions et des régions admissibles de l'effecteur final des doigts. Enfin, nous avons mis en place et simulé des programmes Simulink qui permettent la préhension d'objets de formes et dimensions connues.

Mots clés : Robotique, Simulation, Main robotique, Préhension, Trajectoire, Région admissible, Solidworks, Matlab/Simulink.

Abstract

In this work, our goal is to establish a connection between the 3D modeling software *SolidWorks* and Matlab/Simulink for a model of an anthropomorphic robotic hand actuated in each phalanx of each finger. We have provided an overview of artificial hands and a definition of the grasping action, along with the simulation approach employed. Subsequently, we described the 3D model of the hand used, which is available as Open-Source, as well as the Simulink programs responsible for generating finger movements. We then applied control and plotted curves for the trajectories, positions, and permissible regions of the fingers end effector. Finally, we implemented and simulated Simulink programs that enable the grasping of objects with known shapes and dimensions.

Keywords : Robotics, Simulation, Robotic Hand, Grasping, Trajectory, Permissible Region, SolidWorks, Matlab/Simulink.

ملخص

في هذا العمل، هدفنا هو تحقيق تواصل بين برنامج نمذجة 3D *SolidWorks* وبرنامج Matlab/Simulink لنمذجة يد آلية روبوتية تعمل في كل مفصل لكل إصبع. قدمنا تاريخاً عن الأيدي الاصطناعية وتعريفها لعملية الإمساك، بالإضافة إلى النهج المستخدم في المحاكاة. فيما بعد، وصفنا النموذج 3D المستخدم لليد، والذي يتوفر بشكل Open-Source، بالإضافة إلى برامج Simulink المسؤولة عن توليد حركة الأصابع. ثم قمنا بتطبيق التحكم ورسم منحنيات المسارات والمواقع والمناطق المسموح بها للمنتج النهائي للأصابع. وأخيراً، قمنا بإعداد ومحاكاة برامج Simulink التي تتيح الإمساك بأجسام ذات أشكال وأبعاد معروفة. الكلمات الرئيسية: الروبوتيات، المحاكاة، اليد الروبوتية، الإمساك، المسار، المنطقة المسموح بها، *SolidWorks*، Matlab/Simulink.