Rapport de Projet

Majeure : *Biomécanique et robotique médicale*

Une image contenant arme

Description générée automatiquement**Réalisation d’un bras de robot chirurgien pour de la télé opération**

*Projet réalisé par :*

Warith HUSSAIN

Claire GINOUX

Aurélien MEAUX  
  
*Responsable de projet :* Ugo VOLLHARDT

*Responsable de majeure :* Alex CALDAS

# Remerciements

Tout d’abord, nous tenions à remercier toutes les personnes qui nous ont aidés a concrétiser ce projet. Ils nous ont soit aidés, soit guidés, soit inspirés dans la création et la fabrication de notre solution :

• Monsieur Alex CALDAS, responsable de la majeur dans laquelle nous étudions : Robotique médicale et biotechnologies. Il nous a longuement aidé à travers ses conseils, méthodes. Il nous a, de plus, fournit le matériel nécessaire pour le projet. De plus, nous avons pu nous aider de ses cours enseignés afin de réaliser certaines parties.

• Monsieur Ugo VOLHARDT, responsable du projet, il nous a aidé continuellement à l’avancement et à la réalisation du projet par le biais de ses conseils, cours et méthodes. Nous avons grâce à lui, pu en apprendre davantage sur le métier d’ingénieur. Nous le remercions pour son temps durant nos réunions.

• Monsieur Karim AIT, enseignant-chercheur en automatique et électronique. Son cours de « Systèmes articulés » qui nous as été enseigné nous a été d’une grande aide pour la réalisation technique du projet.

• L’ESME Sudria et l’ensemble des enseignants pour leur aides et leurs cours enseignés

# Tables des matières

[Rapport de Projet 1](#_Toc92053417)

Remerciements………………………………………………………………………………………..2

[Tables des matières 3](#_Toc92053418)

[I. Introduction 4](#_Toc92053419)

II. Cahier des charges …………………………………………………………………………… 5

[1. Contexte 5](#_Toc92053420)

[2. Problématique 5](#_Toc92053421)

[3. Réponse à la problématique 5](#_Toc92053422)

III. Projet 7

1. Modélisation 3D du bras………………………………………………………………………...7

2. Détermination du modèle géométrique direct et inverse………………………………….…9

3. Software………………………………………………………………………………….……...11

4. Hardware…………………………………………………………………………………………14

IV. Conclusion……………………………………………………………………………….………15

# 

# Introduction

Actuellement en dernière année du cycle ingénieur à l’ESME Sudria, nous avions notre projet de fin d’études à réaliser, qui a été choisi en amont par notre directeur de majeure.

Dans le cadre de ce projet, nous souhaitions ainsi utiliser les connaissances acquises jusqu’à lors et acquérir de l’expérience sur le thème de la biomécanique et de la robotique médicale.

Ce projet de fin de cursus est l’occasion pour nous d’appliquer nos connaissances à un projet motivant permettant l’amélioration perpétuelle de la condition de vie des patients.

Aujourd’hui, la robotique médicale prend tout son sens car c’est un marché en pleine émergence, de nombreuses technologies révolutionne ce domaine et permettent aux patients du monde entier d’accéder à de meilleurs soins.

Le rapport ci-présent détaille le travail fourni pour notre projet depuis le début du semestre.

Le projet consiste à concevoir un bras chirurgical qui seras contrôlé à distance par la reproduction de mouvement d’un chirurgien via une caméra de type Kinect.



1. Cahier des charges

# Contexte

Le monde actuel est rempli d'innovations technologiques qui se modifient et s'améliorent de jour en jour. Nous pouvons observer cela au niveau du domaine médicale. Les connaissances acquises grâce aux recherches scientifiques nous pousse à développer des outils de travails encore plus efficace afin de satisfaire la nécessité.

Les études de médecine afin de devenir chirurgiens se comptent sur une dizaine d’année. Malgré ces nombreuses années d’études, les précisions nécessaires pour certaines interventions ne peuvent être fait à 100% par un chirurgien.

Pour pallier ce problème de précision, des outils sont développés pour assister le chirurgien, lors des opérations.

# Problématique

Aujourd’hui, de nombreuses opérations chirurgicales sont extrêmement complexes et peuvent être rare, de ce fait il est possible qu’une certaine opération délicate ne soit pas réalisable par tous les chirurgiens du monde.

On se retrouve donc avec des chirurgiens spécialisés qui pourrait opérer mais qui ne se trouve pas sur place (possiblement dans un autre pays) et cela pose donc un réel problème pour l’accès au soin dans le monde.

De plus, les mains d’un chirurgien ne pouvant être aussi précis que le fonctionnement du corps humain, nous cherchons à développer un bras robotisé qui permettrais aux chirurgiens d’opérer avec une précision accrue.

# Réponse à la problématique

C’est dans ce contexte que nous souhaitons réaliser un bras chirurgical motorisé possédant comme organe effecteur un outil chirurgical type scalpel. Ce bras a pour objectif de pouvoir accompagner le personnel médical lors d’opération en reproduisant les mouvements d’un chirurgien à distance. Il permettra ainsi d’opérer des patients à des milliers de kilomètres de manière simplifié et donc de faciliter l’accès à de meilleurs soins.

Pour cela, nous allons utiliser une Kinect Microsoft qui va pouvoir détecter les mouvements de bras du chirurgien et les retransmettre. Ces données seront alors récupérées par un Raspberry Pi 3B+ qui servira de liaison avec le bras motorisé, et pourra donc contrôler les moteurs. A partir de là nous avons ainsi une reproduction des mouvements du chirurgien par le bras.

Ainsi, on peut définir notre projet de manière simplifié à travers le schéma synoptique :

Kinect Microsoft

Bras motorisé

Raspberry PI

# Projet

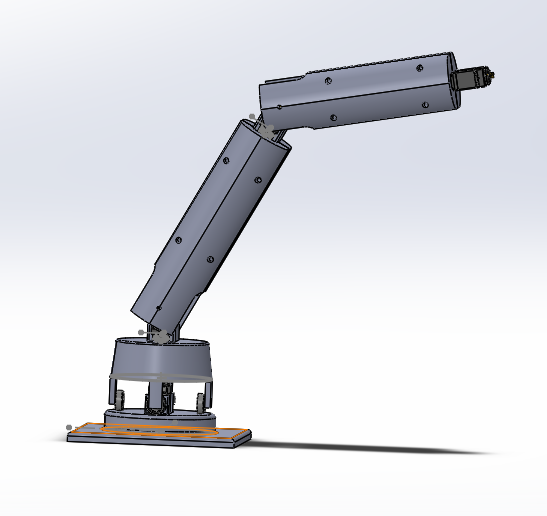
1. Modélisation 3D du bras

Afin de concevoir notre bras chirurgical, nous avons utilisé le logiciel de CAO SolidWorks pour la partie modélisation. Nous avons choisi de faire un bras contenant 4 rotoïdes et qui utilise 4 moteurs.

Assemblage du bras entier à l’exception de l’embout permettant le maintien de l’outil médical :

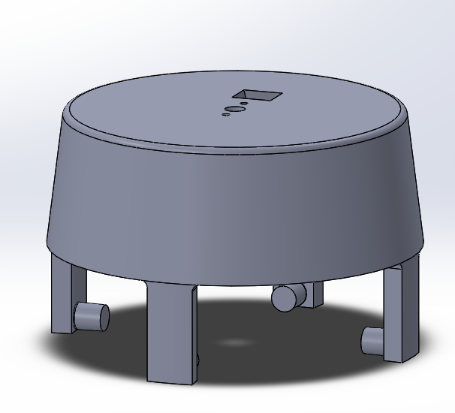
Une image contenant texte

Description générée automatiquement



*Vue de côté Vue de coté*

Le bras chirurgical est donc composé d’une base avec un tronc pour une rotation contrôlant la position en X :



Une image contenant texte, boîtier, stationnaire, accessoire

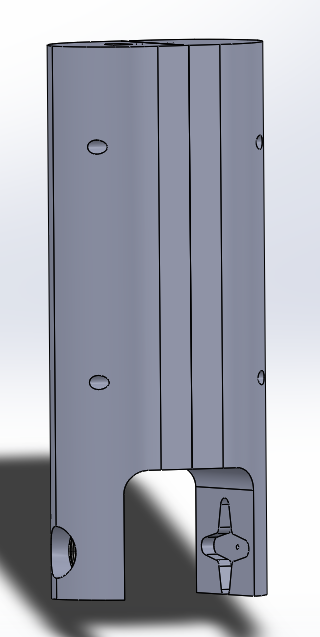
Description générée automatiquement

*Base du bras Tronc du bras*

Nous avons choisi de faire cette partie en 2 pièces séparés afin d’obtenir un meilleur fonctionnement au niveau de la rotation du bras. La base nous permet de bien positionner notre moteur et le tronc d’avoir une bonne rotation grâce à l’intégration de 4 roulements à billes.

Nous avons opté pour une base carrée, afin d’avoir une meilleure répartition des masses et une meilleure stabilité. Le choix de ce carré était aussi d’avoir une surface plus importante pour pouvoir éventuellement ajouter une accroche comme une pince afin de fixer le bras à différents endroits ou des lestes pour stabiliser davantage le bras.

Il possède aussi un bras et un avant-bras pour simuler l’équivalent d’un coude :



Une image contenant mur, intérieur

Description générée automatiquement

*Bras Avant-bras*

Cette partie nous donne la possibilité de contrôler la profondeur et la hauteur de notre bras.

Le bras et l’avant-bras sont basés sur le même modèle de modélisation qui permet d’avoir d’un côté le bon maintien du moteur encré en partie dans le bras et de l’autre un espace permettant la mise en place d’un roulement à billes. L’avant-bras est légèrement plus petit que le bras ce qui lui donne l’avantage d’être aussi plus léger.

A la suite de l’avant-bras, il nous reste une partie qui permettra d’ajouter et d’utiliser un outil chirurgical. Il y aura à ce niveau un dernier moteur qui orientera l’outil médical qui pour notre cas est testé avec un scalpel.

1. Détermination des modèles géométriques direct et inverse

Dans le cadre de ce projet, nous avions besoin de déterminer avec précision les positions articulaires du bras robotisé afin de reproduire au mieux les mouvements réalisés par le chirurgien.

En effet, lorsque le chirurgien réalise des mouvements, nous avons la Kinect Microsoft qui nous permet d’en récupérer leur position et leur orientation. Pour pouvoir ainsi les reproduire sur le bras, nous devons partir de la position et de l’orientation connu pour lui transmettre les positions articulaires à avoir. Pour réaliser cela, nous avons besoin de calculer le modèle géométrique inverse.

C’est ainsi que nous avons déterminer dans un premier temps le modèle géométrique direct de notre bras afin de pouvoir dans un second temps calculer le modèle géométrique inverse.

Le modèle géométrique direct nous permet, à l’aide des positions articulaires, de calculer la position et l’orientation de l’organe effecteur.

Nous avons alors commencé par représenter les différents axes du bras afin d’en connaitre tous les paramètres géométriques utiles pour nos calculs.

Z2

X2

X4

Z1

X1

Y1

Y4

Z4

X3

Y2

Y3

Z3

Y0

Z0

X0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Axe |  | d | a |  |
| 1 |  |  | 0 |  |
| 2 |  | 0 |  | 0 |
| 3 |  | 0 | 0 |  |
| 4 |  |  | 0 | 0 |

A partir de ces paramètres nous avons pu utiliser la matrice de Denavit-Hartenberg qui permet de connaître la position relative de deux solides avec seulement 4 paramètres au lieu de 6 et nous simplifie ainsi les calculs.

*Matrice de Denavit-Hartenberg*

Nous avons ainsi obtenu la matrice suivante :

Par la suite, nous avons repris ce calcul afin de déterminer le modèle géométrique inverse qui nous permet donc de connaitre les positions articulaires du bras à partir de la position et l’orientation de l’organe effecteur.

On obtient la matrice ci-dessous :

A partir de cette matrice, nous pouvons obtenir les positions articulaires du bras qui nous permettent d’atteindre une position donnée.

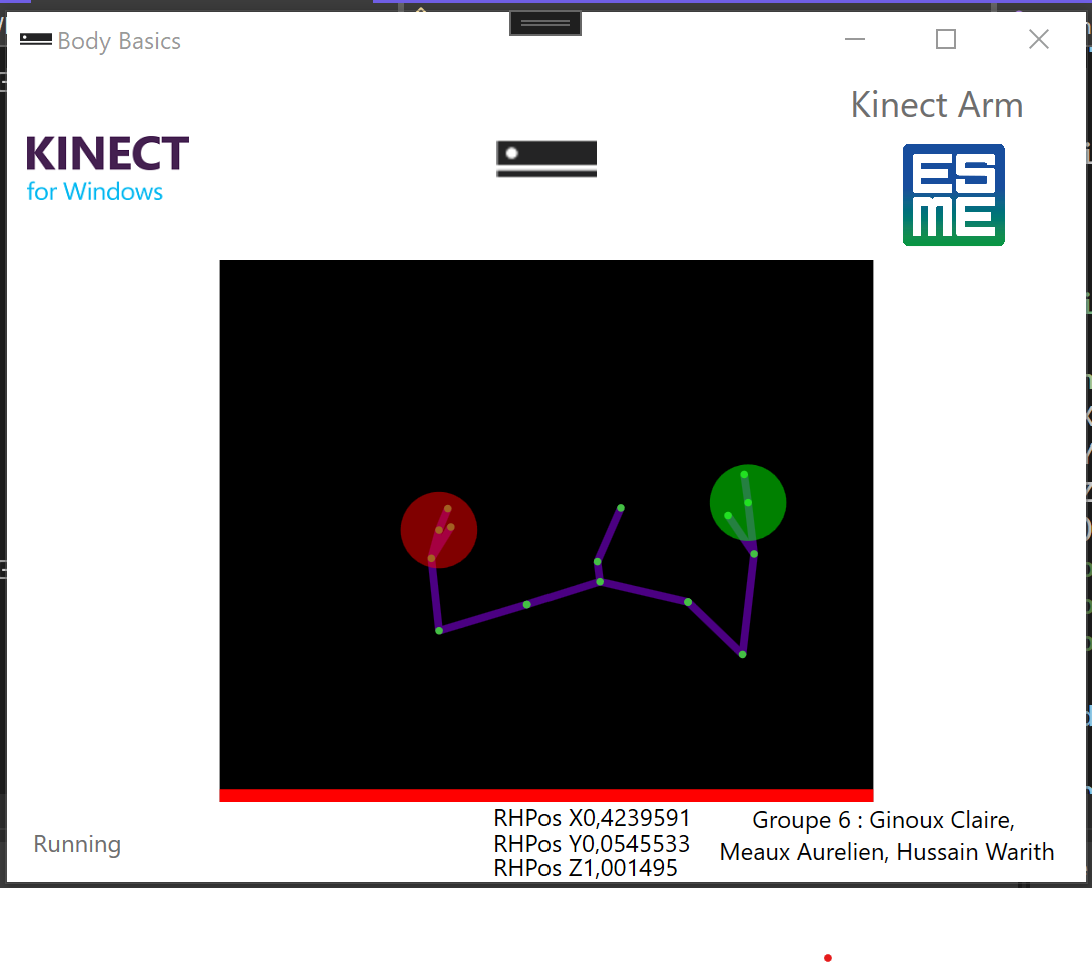
1. Software

En ce qui concerne la partie du code, nous avons procéder en 3 grandes étapes :

* La récupération des données de la Kinect
* Le contrôle du bras
* La communication entre les deux.

La Microsoft Kinect est une caméra initialement conçue pour la Xbox, et qui a été rendu en Open source afin de permettre au développeur de créer du contenu avec cette caméra. La Kinect, ayant pour capaciter de traquer les articulations d’un corps, nous pouvons récupérer la position de la main par rapport à la Kinect.

Pour cela, nous avons récupéré un code d’exemple ayant une interface graphique, codée en C#, qui permet de visualiser des squelettes que la Kinect peut capter. N’ayant pas besoin des articulations autres que celles de la main nous avons donc enlevé toutes les articulations du bas du corps, ce qui nous laisse le haut du dos, la tête, les 2 épaules, les 2 coudes et les mains.



Nous avons donc une moitié de squelette avec seulement les informations nécessaires. Nous avons ensuite récupéré les positions X, Y et Z de la main droite. Ces positions sont données par rapport à la Kinect, il faut donc donner une bonne position pour la Kinect, et ne pas bouger par rapport la caméra.

Pour pallier ce problème, nous allons récupérer la position du milieu de dos, et la comparer à la position de la main, ce qui nous permettrai de pouvoir bouger et ne pas forcément avoir de position précise par rapport à la caméra. Un calibrage sera donc fait pour récupérer juste la distance d’un bras tendu afin de pouvoir utiliser le bras avec diffèrent gabaries de personne.

Nous obtenons donc la position sur l’interface graphique et des petites informations supplémentaire, comme la connexion de l’ordinateur avec la Kinect. Nous pouvons aussi ajouter la connexion établie avec la Raspberry Pi.

La Kinect est capable de reconnaitre si la main est fermée, ouverte ou en position lasso

Une image contenant texte

Description générée automatiquement Une image contenant texte

Description générée automatiquement Une image contenant texte

Description générée automatiquement

Nous allons donc utiliser cette fonctionnalité afin de permettre à l’utilisateur d’arrêter le control du bras chirurgical et de le réactiver.

Le contrôle du bras se code en python sur la Raspberry Pi. Afin de pouvoir faciliter l’exécution et rendre propre notre code, nous avons fait des classes pour chaque fonctionnalité. Le contrôle d’un moteur se fait avec la classe MotorControl.

Elle prend en paramètre le GPIO sur lequel est connecté le moteur, une position initial qui est par défaut 7 (qui correspond à une position du moteur à 90° pour une fréquence de PWM à 50 Hz), une position minimale que le moteur dois pas dépasser qui est par défaut 2 (qui correspond à 0° pour une fréquence de PWM à 50 Hz), une position maximale que le moteur dois pas dépasser qui a pour valeur par défaut de 12 (qui correspond à 180° pour une fréquence de PWM à 50 Hz) et enfin la fréquence qui correspond à la fréquence de PWM qui est par défaut 50.

Le but d’utiliser toutes ces variables en paramètre, permet de configurer chaque moteur en fonction de ses capacités et de l’utilisation voulu, les 2 moteurs du bras n’ont besoin d’atteindre des positions entre 0° et 100°, tandis que la base doit pouvoir pivoter de 0° à 180° voir même 360°.

Les moteurs tournent trop vite, donc nous avons créé une méthode permettant de ralentir la vitesse, qui consiste à aller de la position actuelle à une position donnée en paramètre en s’incrémentant de 0.1 et en mettant un time.sleep de 0.2 secondes qui permet de stopper le programme. Cette façon de procéder permet de ralentir considérablement la vitesse de rotation des moteurs.

Le fichier ProjectMain permet de lancer tout le projet, c’est ce qui doit être lancé pour faire fonctionner le bras.

On initialise tous les moteurs, avec les paramètres nécessaires, et on les entre dans un dictionnaire. Ce dictionnaire est ensuite envoyé à la classe Control qui pour l’instant permet de contrôler le bras indépendamment de la Kinect.

La classe Control utilise les threads qui nous permet de faire tourner les 4 moteurs en même temps, cela évite d’attendre un par un les moteurs d’atteindre leurs positions et nous pourrons donc faire des découpes plus propres et droite.

Pour l’instant, cette class demande en boucle à l’utilisateur d’entrer une position à atteindre et actionne le threading pour que chaque moteur atteint sa position.

Nous allons par la suite grâce au modèle géométrique et cinématique et les positions récupéré par la Kinect, retranscrire les positions articulaires à atteindre pour contrôler les moteurs et atteindre les positions que nous souhaitons pour le bras chirurgical.

Enfin nous avons codé un serveur et un client UDP pour permettre la communication entre la Raspberry qui contrôle le bras et l'ordinateur qui récupère les données de la Kinect.

Le protocole UDP étais le plus pratique pour notre projet, car il permet de transférer des données sans se soucier de la réception. Utiliser dans le streaming, cette méthode est utilisée car sur le nombre de donnée envoyé, une trame en moins peut être négliger.

La Raspberry étant celle qui recevra les données, nous l'avons configuré en tant que Serveur et l'ordinateur avec la Kinect seras en tant que Client. Afin de ne pas avoir à récupérer l'adresse IP a chaque connexion Wifi de la Raspberry, nous avons fixé l'adresse IP en dur, afin que celle-ci ne change pas. Le client se connecte ensuite à cette adresse IP fixé et envoie les données.

1. Hardware

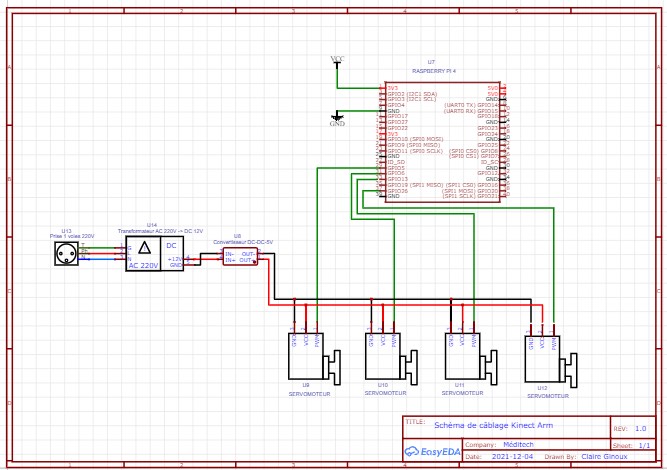
Concernant la partie Hardware de notre projet, les composants sont nombreux et ont chacun leurs particularités différentes.

En effet pour réaliser ce projet nous avons utilisé un transformateur AC/DC permettant la conversion de la tension de secteur de 220V à 12V. Puis, nous utilisons un convertisseur DC/DC permettant de convertir le 12V en 5V afin d’alimenter nos différents moteurs.

Concernant les moteurs nous avons utilisé 4 servomoteurs dont 2 possédants un couple très élevé de 35 kg/cm^3 permettant aux moteurs de résister aux différents efforts dû au poids.

Enfin, évidemment, la carte électronique Raspberry Pi permettant le contrôle des différents moteurs.

Vous trouverez dessous, le schéma de câblage correspondant à notre partie Hardware.



# Conclusion

Pour conclure, ce projet est très intéressant et nous permet d’appliquer un grand nombre de nos connaissances. De plus, il nous reste encore une bonne partie avant le rendu final mais nous sommes plutôt satisfaits de l’avancement étant donné que la partie électronique et informatique est opérationnelle.

Nous allons donc nous investir encore durant cette dernière ligne droite afin de rendre un projet qui soit fonctionnel et rendu dans les temps.