



ÉCOLE
CENTRALE LYON



Centre de réadaptation
Marie Enfant
CHU Sainte-Justine
Le centre hospitalier
universitaire mère-enfant

Université 
de Montréal

Rapport de stage d'application

Rapport de stage du projet : conception d'un bras robotique en
impression 3D capable de jouer au ping-pong

Auteur

Quentin GALLOUEDEC
Élève ingénieur à l'École Centrale de Lyon
quentin.galloeudec@ec116.ec-lyon.fr

Commanditaire

Maxime RAISON
B.Ing. , DEA et Ph.D
Département de génie mécanique
École Polytechnique de Montréal
maxime.raison@polymtl.ca

Durée du stage : 14 semaines
14 décembre 2018

Résumé

Selon une étude réalisée par NASDAB entre 2004 et 2005[5], environ 8% des amputations sont réalisées sur des enfants de moins de 16 ans. A cela, on peut rajouter les enfants atteints d'une pathologie congénitale ou qui souffrent d'une malformation de naissance (récents exemples d'enfants nés sans bras ou sans main). Pour limiter le handicap lié à cette situation, de nombreux travaux ont été réalisés dans le domaine des prothèses. Cependant, le coût des prothèses existant est très important.

Mon stage de recherche a pour but de démontrer que l'impression 3D associée à des systèmes de commande simples permet d'obtenir un bras robotisé à faible coût avec des performances acceptables. J'ai donc passé quatorze semaines au Centre de Réadaptation Marie Enfant (CRME) avec pour objectif de construire un bras robotisé simple que l'on mettra au défi de jouer au ping-pong.

Le bras devant être développé intégralement, j'ai commencé par établir l'architecture générale, pour ensuite dimensionner les différents composants du bras. Enfin j'ai réalisé un prototype en impression 3D et réalisé l'algorithme de contrôle.

Au terme de mon étude, le système permet de détecter la balle et de la placer selon les trois dimensions de l'espace. Cependant, le bras ne permet pas encore de jouer au ping-pong. Il reste encore un travail à réaliser sur la commande des moteurs.

Ce stage m'a permis d'avoir une première expérience dans le domaine de la recherche, ce qui était nécessaire dans la construction de mon projet professionnel. A cette occasion, j'ai également pu voyager et apprécier la culture montréalaise, et m'essayer à la vie d'expatrié.

Abstract

According to a study published by NASBAD between 2004 and 2005[5], about 8% of people who undergo an amputation are children under age 16. To these we may add those who are suffering from congenital diseases or those who are born without an arm or a hand (recent cases of child who are born without a member). To limit the disability caused by this situation, numerous projects has been led to develop prostheses. However, they are expensive.

The aim of my research internship is to prove that the 3D printing with some simple command systems can be a means to create a cheap robotic arm with suitable performances. I spent eighteen weeks in Marie Enfant Rehabilitation Center (CRME) to develop a simple robotic arm capable to play tennis table.

The arm must be developed from start to finish, I began to established the global architecture. Then, I dimensionned the parts of the arm. Finally, I created a prototype based on 3D printing and wrote the motors control algorithm.

As it is, the system is able to position the ball in the sapce, and it can move. However, it can't actually play tennis table. There is still some work to do on the motors command.

This internship has allowed me to have a first experience in the research domain, which was necessary in my professionnall project construction. I took this opportunity to travel, and appreciate the culture of Montreal, and to experience the expatriate life.

Table des matières

1	Présentation du projet	5
2	Environnement de travail	7
2.1	Présentation du centre	7
2.2	Histoire du centre	7
2.3	Pavillon de recherche	7
3	Études préliminaires	8
3.1	Architecture générale du bras robot	8
3.2	Moteurs	9
3.2.1	Exigence de vitesse	9
3.2.2	Exigence de couple	10
3.3	Détection de la balle par caméras	11
3.3.1	FPS	11
3.3.2	Une seule caméra?	11
3.3.3	Vision à deux caméras	13
3.3.4	Détermination de x_b et y_b	13
3.3.5	Détermination de z_b	14
3.3.6	Résolution pour la vision à deux caméras	15
3.3.7	Conclusion sur les performances requises	16
4	Réalisation du prototype	16
4.1	Mécanique du robot	16
4.2	Commande des moteurs	19
4.3	Algorithme de détection de la balle	20
5	Conclusion du projet	22
6	Le stage de recherche dans la construction de mon projet professionnel	24
6.1	Choix du stage	24
6.2	Aspects marquants	24
6.3	Valorisation	24
7	Conclusion du stage	25
A	Code Matlab pour la détection de la balle	26
B	Logiciels utilisés et version	27

Table des figures

1	Le bras JACO a permis à Sébastien Lepage d'accéder à l'emploi. Il est aujourd'hui analyste-programmeur. Source : [2]	5
2	Le bras Forpheus d'Omron jouant au ping-pong au CES 2018	6
3	Le coup droit correspond à un mouvement de la raquette dans un plan incliné par rapport à la surface de la table	8
4	Architecture du bras	8
5	Détection de la balle avec une caméra	12
6	Erreur sur y_b engendrée par une erreur d'un pixel sur l'estimation du rayon . . .	12
7	Paramétrage pour la détermination de x_b et y_b	13
8	Paramétrage pour la détermination de z_b	15
9	Modèle Catia du robot sur une table de ping-pong (vue joueur)	16
10	Modèle Catia du robot sur une table de ping-pong (vue adversaire)	17
11	Photo du bras robot joueur de ping-pong prise le 09/08/2018	17
12	Photo du bras robot joueur de ping-pong prise le 09/08/2018	18
13	Schéma électronique pour la commande des moteurs	19
14	Logiciel MSP V1.7 qui permet d'avoir un retour sur les grandeurs du moteur . .	20
15	Fenêtre de paramétrage de la caméra	22
16	Balle détectée par l'algorithme de vision	22

Introduction

1 Présentation du projet

Le projet consiste en la réalisation d'un bras robotisé à deux degrés de liberté, autonome, qui serait capable de jouer au ping-pong. En utilisant un système de caméras, il localiserait la balle pour commander la frappe de la balle. L'objectif à terme est de donner une première itération à l'utilisation de l'impression 3D dans la réalisation des bras robot dans le secteur de la réadaptation orthopédatrique. Le principal avantage de l'utilisation de l'impression 3D réside dans le coup de la matière première. En effet, pour un bras dont la masse totale serait de 1.5 kg, le prix de la matière première (PLA) s'élèverait à environ 40 CAD.

En faisant des bras *low cost*, on permettrait à de nombreux enfants dont la situation le préconise, d'accéder à un bras robot. La croissance de l'enfant le rendant obsolète en peu de temps, il serait tout à fait possible de le remettre sur mesure assez souvent sans que le prix ne deviennent déraisonnable.

Un exemple d'un bras articulé disponible aujourd'hui sur le marché est présenté figure 1. Ce bras coûte environ 50 000 CAD.



FIGURE 1 : Le bras JACO a permis à Sébastien Lepage d'accéder à l'emploi. Il est aujourd'hui analyste-programmeur. Source : [2]

Si ce genre de bras possède de nombreux degrés de liberté, la manipulation d'un tel bras reste compliquée. Les avancées technologiques actuelles ne permettent pas de reproduire le mouvement d'un bras humain fidèlement. En acceptant ce constat, il paraît donc intéressant de créer une gamme de bras robotisés accessibles à la plupart des demandeurs, dont les performances correspondraient aux fonctions basiques qu'on peut attendre d'un bras robotisé.

Il existe déjà plusieurs robots permettant de jouer au ping-pong. On peut citer l'exemple du robot Forpheus[1] développé par Omron, présenté figure 2.



FIGURE 2 : Le bras Forpheus d'Omron jouant au ping-pong au CES 2018

Ce robot a été fait pour démontrer la puissance de l'IA et des capteurs développés par Omron [3]. Il a été certifié par le Guinness World Records comme étant le premier robot permettant d'apprendre à jouer au ping-pong.

Cependant, dans notre étude, nous recherchons une architecture inspirée de la morphologie humaine. Il faut que la forme du robot s'apparente à un bras humain.

2 Environnement de travail

2.1 Présentation du centre

Le CRME est un service qui offre aux nouveau-nés, enfants et adolescents présentant des troubles de la motricité ou du langage des services de réadaptation. Sa devise, vieille de 70 ans : "Faire mieux et plus, ensemble pour l'enfant et sa famille".

2.2 Histoire du centre

En 1937, la clinique Bacille Calmette-Guérin est fondée par madame Simone David-Raymond. Son but premier est d'isoler les enfants malades pour mieux les traiter et aussi pour éviter la contamination. Cette clinique évolue et devient une école de puéricultrices. Elle dispose en plus d'un panel étendu de moyens en matière de radiologie entre autres.

La clinique devient un hôpital en 1961, spécialisé en pédiatrie et en réadaptation pour l'enfant. Les services de réadaptation grandissent au fil des années et de l'arrivée de médecins spécialistes. C'est dans les années 1980 que l'hôpital Marie Enfant devient un centre reconnu à l'échelle suprarégionale. Son développement est marqué par plusieurs réalisations dont on peut citer la poussette "Double Buggy" (une poussette double adaptée aux enfants handicapés).

C'est en octobre 1991 que le pavillon de recherche apparaît. C'est dans ce pavillon que j'ai mené mes travaux. C'est en janvier 2000 que l'hôpital Marie Enfant devient le Centre de réadaptation Marie Enfant après avoir été intégré au CHU de Sainte-Justine. C'est aujourd'hui le seul centre de réadaptation pédiatrique de toute la province de Québec.

2.3 Pavillon de recherche

Le centre de recherche compte 178 chercheurs. Ces chercheurs viennent de différents viviers tels que les facultés, les départements universitaires, ou les départements hospitaliers. On rajoute à ça les 300 universitaires en doctorat ou en post-doctorat sous la tutelle des chercheurs.

3 Études préliminaires

3.1 Architecture générale du bras robot

Le bras robot doit être pensé pour reproduire le mouvement d'un joueur de ping-pong. Le mouvement le plus basique et simple à réaliser est celui du coup droit. La raquette se déplace dans un plan incliné d'environ 45° avec la surface de la table.



FIGURE 3 : Le coup droit correspond à un mouvement de la raquette dans un plan incliné par rapport à la surface de la table

L'objectif sera donc de créer un bras qui sera capable de réaliser ce mouvement, et de frapper les balles qui traversent ce plan incliné. La base de l'architecture du bras est présentée figure 4.

Les deux articulations seront actionnées par des moteurs. Les exigences vis-à-vis de ces moteurs sont discutées dans la section 3.2.

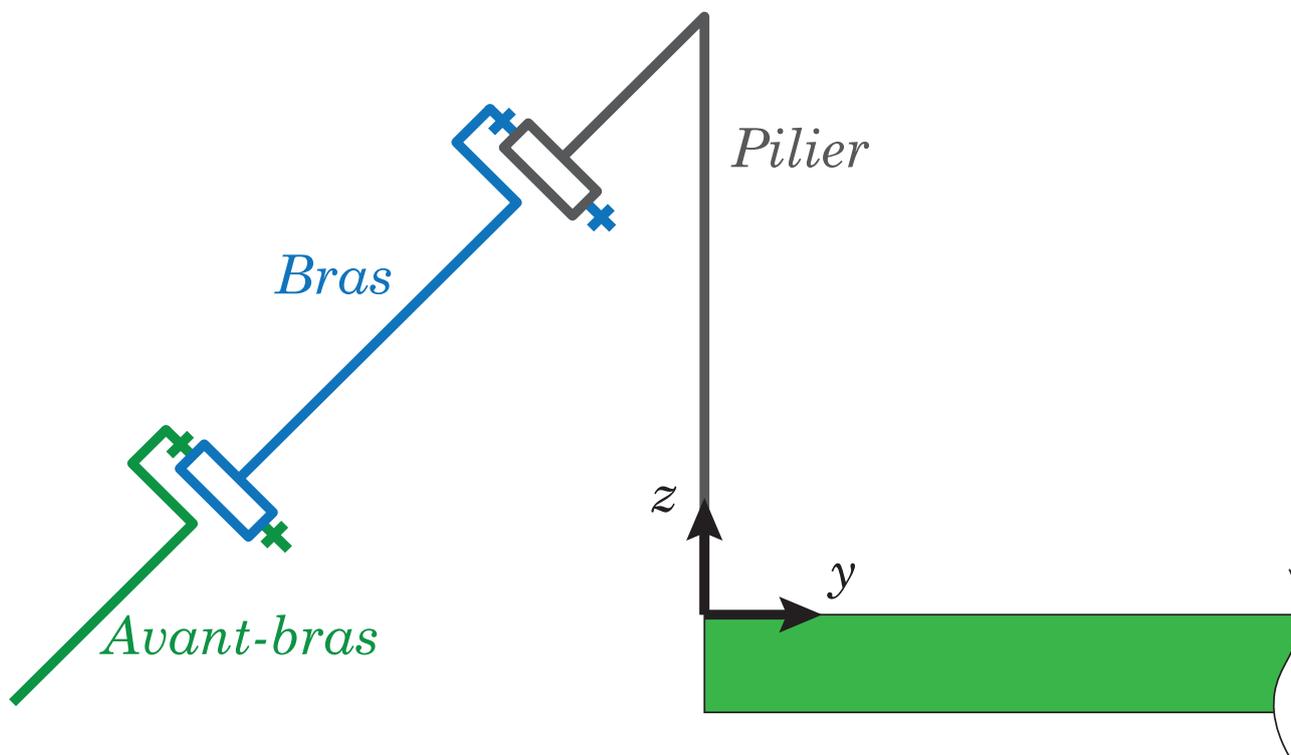


FIGURE 4 : Architecture du bras

Pour être capable de frapper la balle, le robot doit savoir la position de la balle. Pour cela, des caméras seront utilisées. Elles sont placées au-dessus du bras, et orientées vers la table. On discute des exigences de ces caméras dans la partie 3.3.

Enfin, toute l'architecture du bras sera réalisée en PLA imprimée à l'aide de l'imprimante MakerGear M2 et du logiciel Simplify 3D 4.0.1. Le bras doit être capable de couvrir la largeur de la table. Comme il est placé au centre de la table, la longueur cumulée du bras et de l'avant-bras doit valoir la moitié largeur de la table :

$$l_{bras} + l_{avant-bras} = \frac{l_{table}}{2} = 76.25 \text{ cm} \quad (1)$$

Pour limiter l'inertie du bras, on placera les moteurs au niveau du pilier. Les moteurs restant fixes, leur masse ne rentrera pas en compte dans les calculs de couple nécessaire pour mouvoir le bras.

L'ensemble des valeurs relatives au ping pong (masse de la balle, dimension de la table) sont tirées du règlement international [4].

3.2 Moteurs

3.2.1 Exigence de vitesse

Situation extrême : bras replié et toute la vitesse provoquée par un seul servo.

Analyse vidéo : <https://www.youtube.com/watch?v=BiWtLbgkch8>

On suppose que le mouvement du bras s'écrit sous la forme d'un sinus

$$h(t) = A \sin(\omega t) \quad (2)$$

Durée du mouvement : 20 images ce qui correspond à 0.8 sec.

$$\omega = \frac{2\pi}{T} \quad (3)$$

$$\omega = 7.85 \text{ s}^{-1} \quad (4)$$

L'amplitude verticale est d'environ 70 cm. Le plan étant incliné de 45° , on a

$$2A = \sqrt{2} * 40 \text{ cm} \quad (5)$$

$$A = 30 \text{ cm} \quad (6)$$

On détermine donc la vitesse max :

$$\frac{dh}{dt}(t)_{max} = A\omega = 2.2 \text{ ms}^{-1} \quad (7)$$

Pour atteindre cette vitesse, avec un bras de 40 cm, la vitesse angulaire nécessaire est de :

$$v = r\dot{\theta} \quad (8)$$

$$\dot{\theta} = \frac{v}{r} = \frac{2.2 \text{ ms}^{-1}}{0.4 \text{ m}} = 5.5 \text{ rads}^{-1} \quad (9)$$

Ce qui correspond à $0.18 \text{ s}/60^\circ$. Donc, il faut un moteur qui puisse atteindre une vitesse de $0.18 \text{ s}/60^\circ$

3.2.2 Exigence de couple

Pour déterminer le couple nécessaire dans les servomoteurs, on se place dans le cas d'un bras déplié à l'horizontal (moment du poids maximum). Pour obtenir une rigidité suffisante, et en prévision d'un éventuel remplacement de matériau, on remplira les impressions à $r = 40\%$. Dans cette configuration, le bras doit pouvoir taper une balle, soit avec l'épaule, soit avec le coude, et donc pouvoir recréer le mouvement décrit dans la partie précédente.

Couple dans l'épaule L'accélération voulue est d'environ 11 rads^{-1} (basé sur le calcul précédent : la vitesse max doit être atteinte le temps d'un mouvement de frappe qui dure une seconde).

$$\Sigma C = J\dot{\omega} \quad (10)$$

$$C_{servo} - C(P_{bras}) - C(P_{avant-bras}) - C(P_{raquette}) = \frac{\Sigma_{masse}\Sigma_{longueurs}^2}{3}\dot{\omega} \quad (11)$$

$$C_{servo} = C(P_{bras}) + C(P_{avant-bras}) + C(P_{raquette}) + \frac{\Sigma_{masse}\Sigma_{longueurs}^2}{3}\dot{\omega} \quad (12)$$

Calculs intermédiaires :

$$C(P_{bras}) = \frac{L_{bras}}{2}r\rho_{PLA}V_{bras}g = \frac{50 \text{ cm}}{2} * 0.4 * 1.25 \text{ g cm}^{-3} * 434 \text{ cm} * 9.81 \text{ N kg}^{-1} = 0.53 \text{ N m} \quad (13)$$

$$C(P_{avant-bras}) = (L_{bras} + \frac{L_{avant-bras}}{2})r\rho_{PLA}V_{avant-bras}g = (50 \text{ cm} + \frac{27.5 \text{ cm}}{2}) * 0.4 * 1.25 \text{ g cm}^{-3} * 195 \text{ cm} * 9.81 \text{ N kg}^{-1} \quad (14)$$

$$C(P_{avant-bras}) = 0.61 \text{ N m} \quad (15)$$

Ici, le centre de masse de la raquette est à l'extrémité de l'avant-bras :

$$C(P_{raquette}) = (L_{bras} + L_{avant-bras})m_{raquette}g = (50 \text{ cm} + 27.5 \text{ cm}) * 0.150 \text{ kg} * 9.81 \text{ N kg}^{-1} = 1.14 \text{ N m} \quad (16)$$

$$J\dot{\omega} = \frac{\Sigma_{masse}\Sigma_{longueurs}^2}{3}\dot{\omega} = \frac{(0.217 \text{ kg} + 0.097 \text{ kg} + 0.150 \text{ kg})(0.50 \text{ m} + 0.27 \text{ m} + 0.25 \text{ m}/2)^2}{3} * 11 \text{ rad s}^{-2} = 1.36 \text{ N m} \quad (17)$$

Donc pour l'épaule :

$$C_{servo} = 0.53 \text{ N m} + 0.61 \text{ N m} + 1.14 \text{ N m} + 1.36 \text{ N m} = 3.64 \text{ N m} \quad (18)$$

Couple dans le coude Même principe de calcul :

$$C_{servo} = C(P_{avant-bras}) + C(P_{raquette}) + \frac{\Sigma_{masse} \Sigma_{longueurs}^2}{3} \dot{\omega} \quad (19)$$

$$C_{servo} = 0.13 \text{ N m} + 0.40 \text{ N m} + 0.14 \text{ N m} = 0.67 \text{ N m} \quad (20)$$

3.3 Détection de la balle par caméras

Pour que le robot soit capable de frapper la balle, il faut qu'il soit capable de la localiser et de prévoir sa trajectoire. Pour cela, on utilise des caméras. La première étape est d'évaluer les performances que ces caméras doivent remplir. Pour cela, commençons pas rappeler quelques ordres de grandeur du problème.

3.3.1 FPS

L'objet à détecter est une balle de ping-pong lors d'un échange. En s'appuyant sur l'analyse de plusieurs vidéos, on obtient les valeurs suivantes : lors de l'échange le plus rapide du monde, la balle avait une vitesse de 8 m s^{-1} . Pour un échange classique, la vitesse de la balle est en moyenne de 3.5 m s^{-1} . C'est la vitesse qu'on prend pour nos calculs : $v = 3.5 \text{ m s}^{-1}$.

On note d la distance parcourue par la balle entre deux coups de raquette. On considère qu'elle correspond à la longueur de la table : $d = 2.74 \text{ m}$.

Pour avoir une mesure fiable de la position et de la vitesse, on exige qu'il y ait au moins 40 mesures entre deux coups de raquette. Ce nombre de mesures permet de limiter le flou de mouvement inhérent à ce moyen de mesure.

Ainsi le temps entre deux coups de raquette est :

$$t = \frac{d}{v} \quad (21)$$

Donc la nombre d'images par seconde vaut :

$$fps = \frac{40}{t} = \frac{20v}{d} = 51 \text{ images s}^{-1} \quad (22)$$

3.3.2 Une seule caméra ?

L'incertitude sur la mesure de la position doit être inférieure à 1 cm au filet, et dans toutes les directions.

La première façon qu'on peut imaginer pour mesurer la position de la balle utilise une seule caméra. La position de la balle sur l'image donne la position de la balle suivant x et z . La position suivant y serait déduite du diamètre apparent.

Il est clair que la plus grosse incertitude serait sur la mesure de la position suivant y . L'incertitude venant de la mesure du rayon, on commence par relier cette incertitude à la résolution de l'image. L'incertitude sur la mesure du rayon correspond à la largeur a d'un pixel projeté sur la plan du filet.

$$a = \frac{l}{W} \quad (23)$$

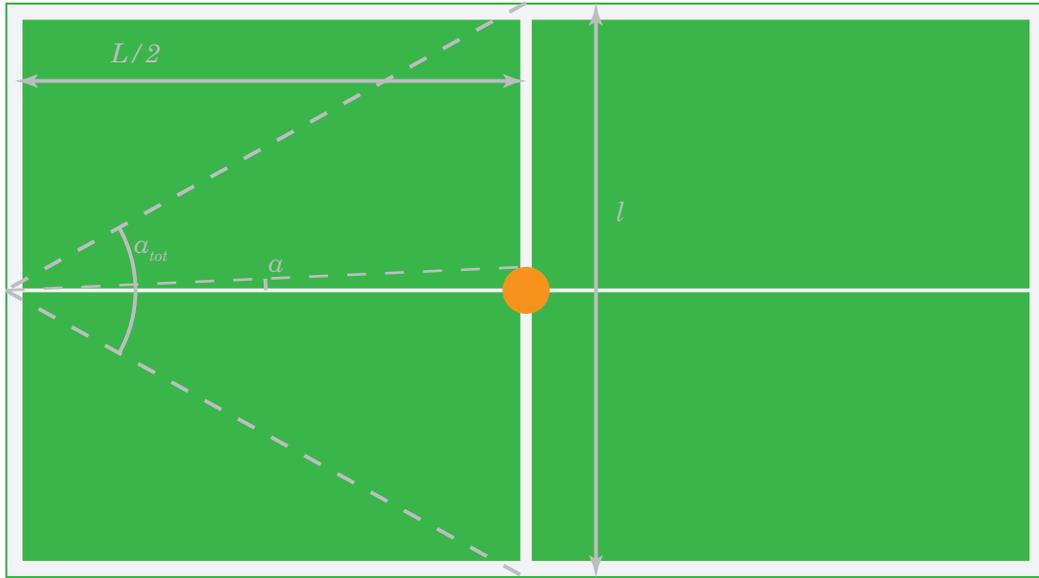


FIGURE 5 : Détection de la balle avec une caméra

L'angle total couvert par la caméra est de :

$$\tan \frac{\alpha_{tot}}{2} = \frac{l}{L} \quad (24)$$

$$\alpha_{tot} = 1.02 \text{ rad} \quad (25)$$

La relation entre le rayon apparent et la distance de la balle :

$$y_{balle} = \frac{r}{\tan \frac{nb_{pix}}{W} \alpha_{tot}} \quad (26)$$

En se servant de cette relation on peut représenter Δy_{balle} (erreur sur la distance) en fonction de la taille de l'image. Le résultat est représenté figure 6.

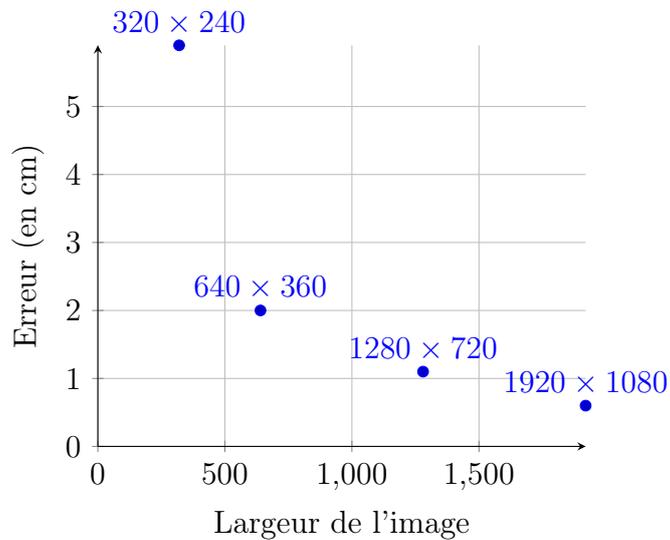


FIGURE 6 : Erreur sur y_b engendrée par une erreur d'un pixel sur l'estimation du rayon

Pour répondre à l'exigence de précision énoncée précédemment, il est nécessaire de traiter des images au minimum de 1920×1080 pixels. En se rappelant que le programme doit traiter au moins 51 images par seconde, on en déduit que le programme doit traiter 105 millions de pixels par seconde. Ce nombre est trop important pour un ordinateur classique.

L'utilisation d'une seule caméra n'est donc pas adaptée, nous utiliserons donc deux caméras et déduirons la position de la balle par triangulation.

3.3.3 Vision à deux caméras

On note (x_b, y_b, z_b) les coordonnées de la balle dans le repère de la table.

3.3.4 Détermination de x_b et y_b

Pour déterminer la position de la balle selon les axes x et y , il s'agit dans un premier temps de déterminer la position horizontale du centre de la balle sur chacune des images. Ces coordonnées sont noté p_1 et p_2 et sont en nombre de pixels.

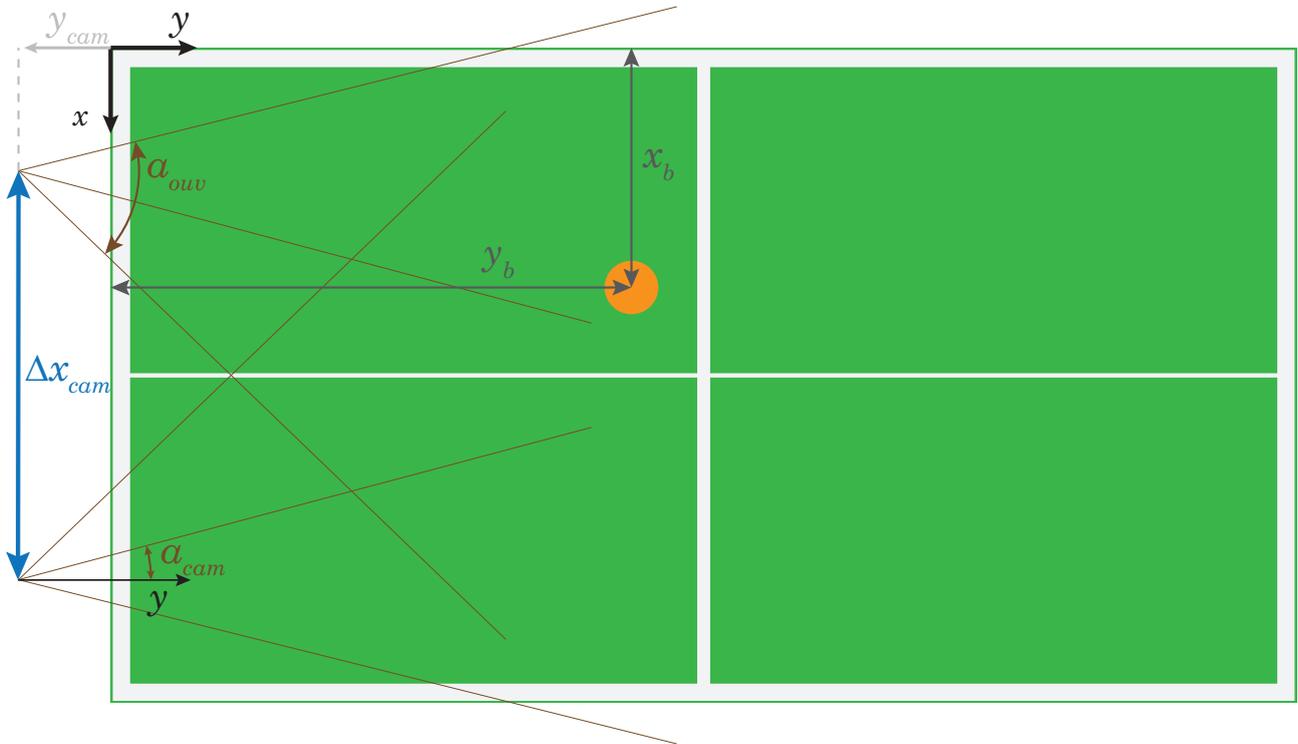


FIGURE 7 : Paramétrage pour la détermination de x_b et y_b

Ensuite, on convertit ces deux valeurs de pixel et en angle par rapport à l'axe de la caméra. On a donc :

$$\alpha_i = \alpha_{ouv} \left(\frac{p_i}{W} - \frac{1}{2} \right) \quad (27)$$

On pose :

$$\beta_1 = \frac{\pi}{2} + \alpha_1 - \alpha_{cam} \quad (28)$$

$$\beta_2 = \frac{\pi}{2} + \alpha_2 + \alpha_{cam} \quad (29)$$

En considérant l'angle que fait la caméra avec l'axe y (noté α_{cam}), on obtient deux droites sur lesquelles se trouvent la balle :

$$y_b - y_{cam} = \tan \beta_1 \left(x_b - \frac{l - \Delta x_{cam}}{2} \right) \quad (30)$$

$$y_b - y_{cam} = \tan \beta_2 \left(x_b - \frac{l + \Delta x_{cam}}{2} \right) \quad (31)$$

En inversant le système, l'équation déduit :

$$x_b = \frac{\tan \beta_1 (l - \Delta x_{cam}) - \tan \beta_2 (l + \Delta x_{cam})}{2(\tan \beta_1 - \tan \beta_2)} \quad (32)$$

$$y_b = \tan \beta_1 \left(x_b - \frac{l - \Delta x_{cam}}{2} \right) + y_{cam} \quad (33)$$

On vérifie par exemple que si la balle est détectée au centre des deux images ($\alpha_1 = \alpha_2 = 0$), on a bien une balle qui se situe au centre de la table suivant x :

$$x_b = \frac{\tan \left(\frac{\pi}{2} - \alpha_{cam} \right) (l - \Delta x_{cam}) - \tan \left(\frac{\pi}{2} + \alpha_{cam} \right) (l + \Delta x_{cam})}{2 \left(\tan \left(\frac{\pi}{2} - \alpha_{cam} \right) - \tan \left(\frac{\pi}{2} + \alpha_{cam} \right) \right)} \quad (34)$$

$$x_b = \frac{2l \tan \left(\frac{\pi}{2} - \alpha_{cam} \right)}{4 \tan \frac{\pi}{2} - \alpha_{cam}} \quad (35)$$

$$x_b = \frac{l}{2} \quad (36)$$

3.3.5 Détermination de z_b

Pour déterminer z_b , on utilise le paramétrage géométrique de la figure 8.

Sur l'image envoyée par la caméra¹, le centre de la balle est repéré par sa position verticale p . Ainsi, l'angle γ est défini de la façon suivante :

$$\gamma = \gamma_{ouv} \left(\frac{p}{H} - \frac{1}{2} \right) \quad (37)$$

Par exemple, sur la figure 8, γ est négatif.

$$\tan (\gamma_{cam} + \gamma) = \frac{z_b - z_{cam}}{y_b - y_{cam}} \quad (38)$$

$$z_b = (y_b - y_{cam}) \tan (\gamma_{cam} + \gamma) + z_{cam} \quad (39)$$

¹en fait on fait la moyenne sur les deux caméras

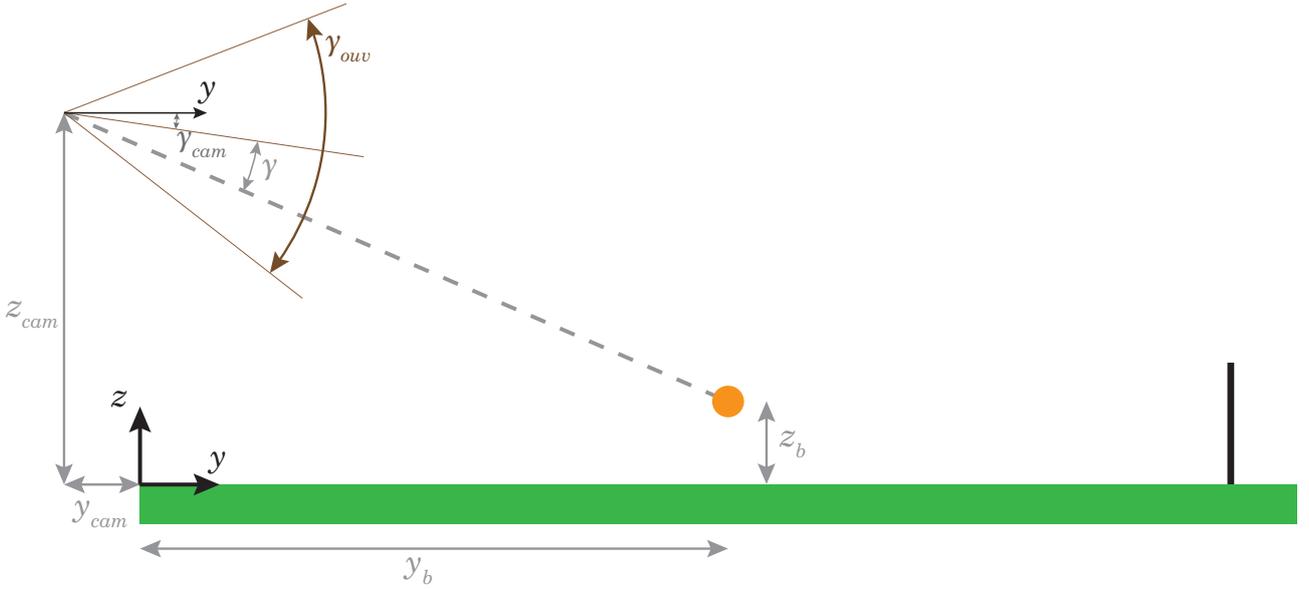


FIGURE 8 : Paramétrage pour la détermination de z_b

Exemple Sur le schéma précédent, on a $\gamma_{cam} = -10^\circ$, $\gamma = -15^\circ$, $y_b = 0.7$ m, $y_{cam} = -0.1$ m, $z_{cam} = 0.47$ m

$$z_b = (0.7 \text{ m} - (-0.1 \text{ m})) \tan(-10^\circ - 15^\circ) + 0.47 \text{ m} = 0.1 \text{ m} \quad (40)$$

3.3.6 Résolution pour la vision à deux caméras

L'objectif est d'évaluer la taille de l'image qui est nécessaire pour que l'erreur sur la mesure de la position de la balle reste inférieure à 1 cm au filet (engendrée par une erreur d'un pixel). L'erreur est la plus grande quand la balle est au plus loin. Ici, la balle est au plus loin quand elle est au filet, dans le coin opposé à la caméra. Cette distance est majorée par la longueur de la diagonale de la demi table² : $d = \sqrt{(1.37 \text{ m})^2 + (1.525 \text{ m})^2} = 2.05$ m.

A cette distance, une erreur de pixel engendre un δx qui vérifie :

$$\tan \frac{\alpha_{ouv}}{W} = \frac{\delta x}{d} \quad (41)$$

Ici on a $\tan \alpha \simeq \alpha$.

$$\frac{\alpha_{ouv}}{W} = \frac{\delta x}{d} \quad (42)$$

$$W = \frac{\alpha_{ouv} d}{\delta x} \quad (43)$$

En prenant $\alpha_{ouv} = 60^\circ$ (valeur classique), on obtient :

$$W = \frac{\frac{\pi}{3} \times 2.05 \text{ m}}{0.01 \text{ m}} = 214 \text{ pixels} \quad (44)$$

²On néglige ici Δx_{cam} , y_{cam} et z_{cam} qui sont de l'ordre d'une dizaine de centimètres

On en déduit que toutes les résolutions avec un largeur de plus de 214 pixels conviennent. Par exemple, 320×240 convient bien. On pourrait mener les mêmes calculs sur H , mais le résultat précédent nous donne un minoration suffisamment faible pour ne pas avoir à la préciser.

En utilisant un résolution de 320×240 à 50 images par seconde, la vitesse de traitement doit être de $2 \times 50 \text{ images/seconde} \times 320 \text{ pixels} \times 240 \text{ pixels} = 7.6 \text{ Mpixels/sec}$. Cette vitesse de traitement raisonnable.

3.3.7 Conclusion sur les performances requises

Le robot disposera de deux caméras et utilisera un algorithme simple de stéréovision basé sur les équations 32, 33 et 39 pour déterminer les trois coordonnées du centre de masse de la balle. Ces deux caméras devront fournir au moins 50 images par seconde. Les images devront avoir une résolution minimum de 320×240 .

4 Réalisation du prototype

4.1 Mécanique du robot

La réalisation mécanique du robot a été faite sous Catia V5. Les fichiers sont disponibles dans le dossier Stage_CRME/Modèle 3D V3. Les autres versions sont des versions plus anciennes issues du travail préliminaire.

La taille d'impression étant limitée, la plupart des pièces a dû être découpée pour pouvoir être imprimée. La fixation des pièces entre elles a été faite avec des écrous.

Le remplissage des pièces a été systématiquement fait en **honeycomb**. Le taux de remplissage est variable en fonction des contraintes dans les différentes pièces. Il s'échelonne entre 30% et 75%. On reste ainsi cohérent avec les calculs de masse déterminant le couple nécessaire pour les moteurs.

Les figures 9 et 10 montrent le bras, sur une table de ping-pong.

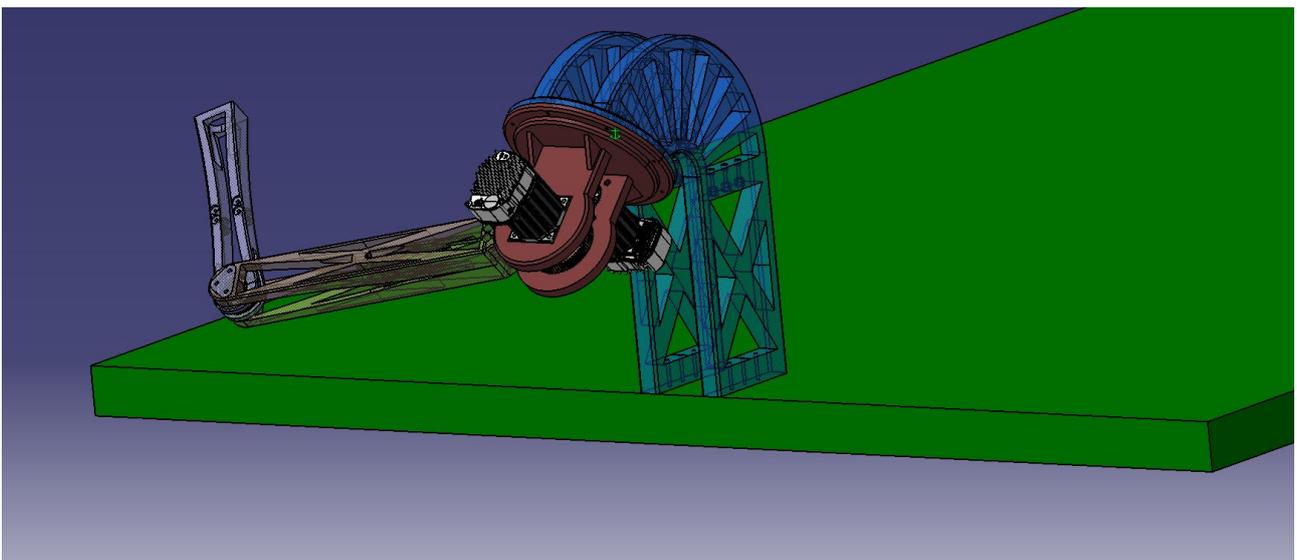


FIGURE 9 : Modèle Catia du robot sur une table de ping-pong (vue joueur)

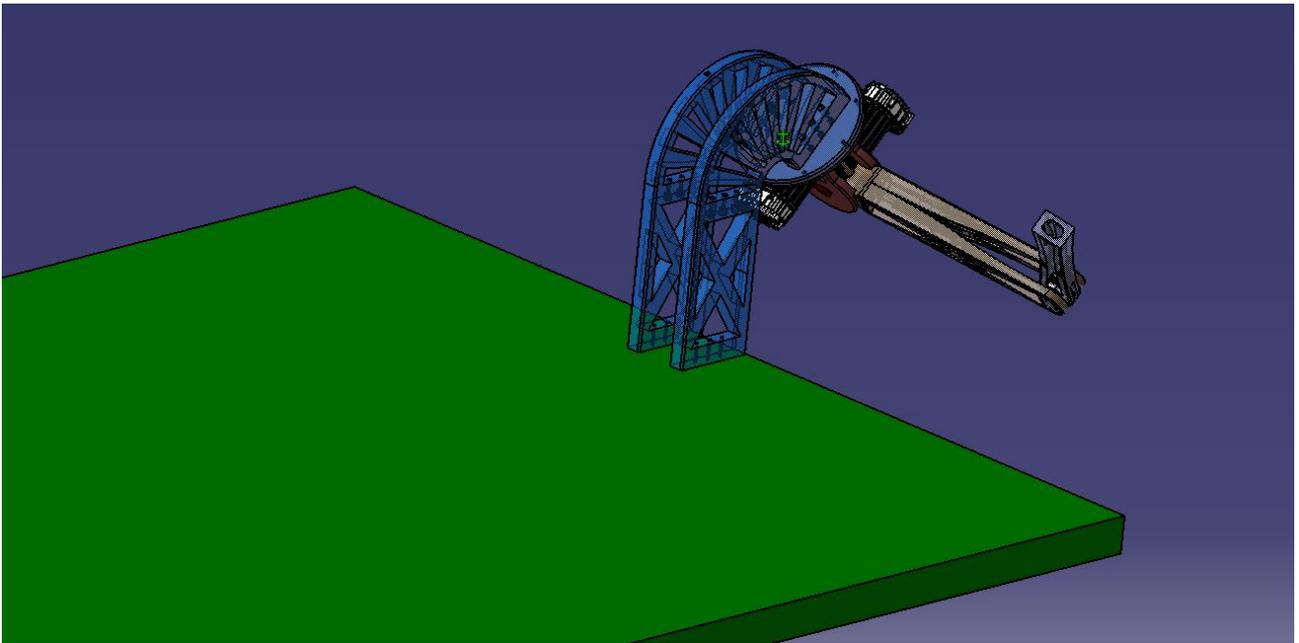


FIGURE 10 : Modèle Catia du robot sur une table de ping-pong (vue adverse)

La réalisation du modèle 3D, l'impression et l'assemblage des pièces sont des étapes quasiment abouties. L'état actuel du bras est présenté figure 11 et 12.



FIGURE 11 : Photo du bras robot joueur de ping-pong prise le 09/08/2018



FIGURE 12 : Photo du bras robot joueur de ping-pong prise le 09/08/2018

Il manque encore deux points importants : la fixation de la courroie, qui est actuellement ouverte. Il faut donc trouver un moyen de la fermer pour que l'action du moteur du coude soit transmise.

Il faut également imprimer la partie inférieure des piliers qui sera elle-même fixée à la table. On obtiendrait ainsi le modèle présenté dans la partie 4.1.

4.2 Commande des moteurs

Le schéma électronique pour la commande des moteurs est présenté figure 13.

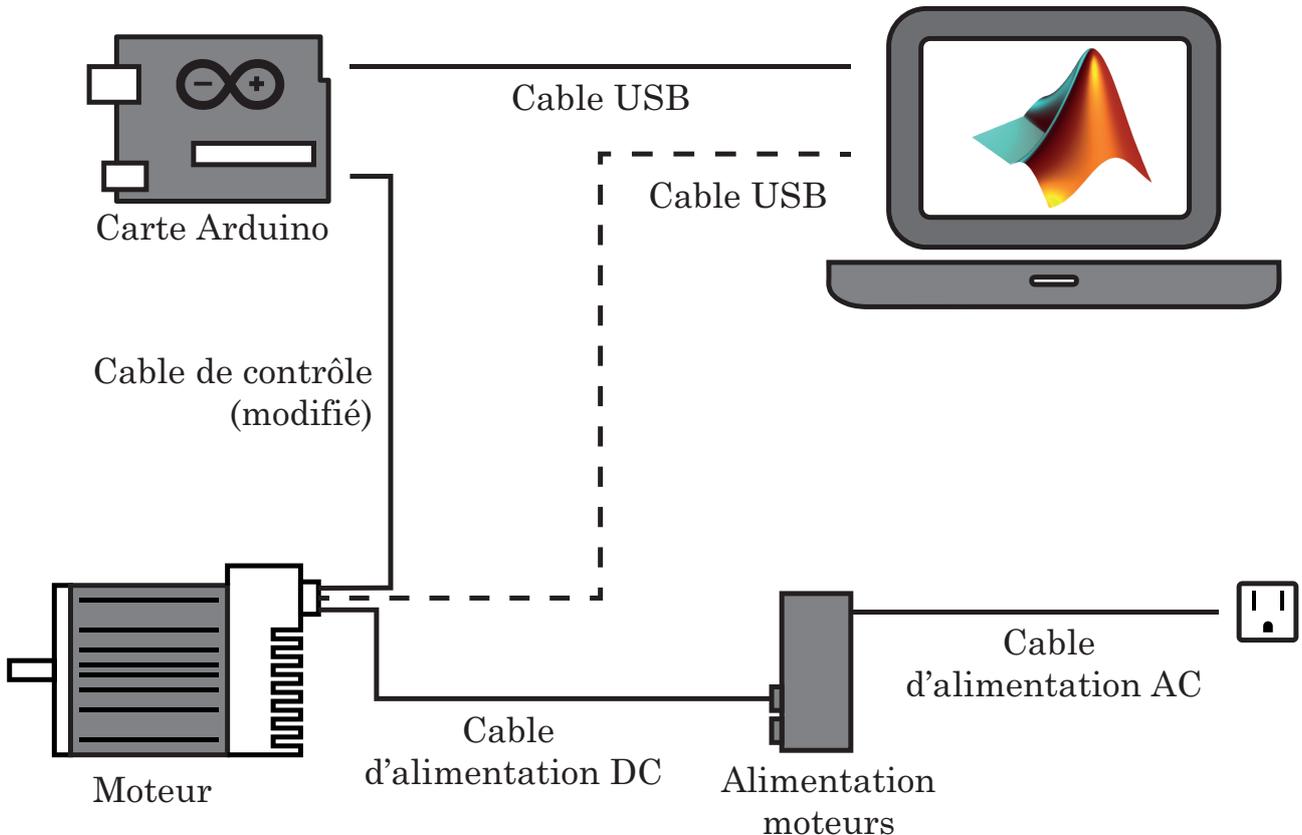


FIGURE 13 : Schéma électronique pour la commande des moteurs

Le moteur est commandé par la carte Arduino : elle envoie des commandes de rotation. Ces informations dépendent de la position de la balle mesurée par les caméras (voir section 3.3).

Il est possible de rajouter un câble USB entre le moteur et l'ordinateur. Ce câble permet, à l'aide du logiciel MSP V1.7 d'avoir un retour des différentes valeurs de couple, de vitesse, de position... sur un graphique dont un exemple est présenté sur la figure 14.

Les moteurs sont commandables avec la carte programmable Arduino. Un exemple simple se trouve dans le dossier `Stage_CRME/Commande moteur/Exemple`

Pour que le programme s'exécute, il faut d'abord installer les trois bibliothèques `PulseClearpath.h`, `StepController.h` et `StepClearpath.h`. Dans l'IDE Arduino : **Croquis - Inclure une bibliothèque - Ajouter la bibliothèque .ZIP**. Puis sélectionner successivement les différents dossiers du même nom que les bibliothèques.

Les différents pins pour le câble de contrôle sont désignés dans l'introduction du code.

Attention, le contrôle du moteur est très sommaire. Un phénomène d'instabilité a été observé lors des premiers tests. Il s'agit d'être vigilant une fois le moteur branché pour éviter tout risque pour l'utilisateur et pour le matériel. Je recommande de desserrer le pignon du moteur pour permettre au moteur de glisser.

La programmation de la commande moteur est fonctionnelle sur le logiciel Arduino en téléversant le code sur la carte. Cependant, pour que le robot puisse fonctionner avec les caméras, il faut que la carte soit contrôlée avec Matlab. Pour que la carte soit utilisable avec Matlab, il faut au préalable installer l'Add-on `MATLAB Support Package for Arduino Hardware` (dis-



FIGURE 14 : Logiciel MSP V1.7 qui permet d'avoir un retour sur les grandeurs du moteur

ponible via Home - Add-ons - Search for Add-ons). Une fois l'Add-on installée, la carte est commandable à partir de Matlab : on peut envoyer directement des instructions à la carte.

Pour contrôler les moteurs, une bibliothèque est nécessaire, comme pour le code sur l'IDE d'Arduino. Cependant, cette bibliothèque n'est pas disponible. En m'appuyant sur la page "Custom Arduino Libraries" disponible sur le site de Matlab, j'ai commencé à créer cette bibliothèque. Ce sont les fichiers disponibles dans le dossier Stage_CRME/Commande moteur/+arduinoioaddons/+Clearpath_commande. Il s'agira donc de continuer la construction des fichiers Clearpath_commande.m et Clearpath_commande.h pour pouvoir utiliser les 3 bibliothèques citées précédemment dans Matlab. Le code principal pour pouvoir définir la nouvelle bibliothèque, définir la carte, définir le moteur est le fichier Stage_CRME/Commande moteur/Commande_arduino.m.

4.3 Algorithme de détection de la balle

Pour développer l'algorithme, nous nous sommes appuyés sur le principe couramment utilisé pour la reconnaissance d'objet sur image :

1. Acquisition d'une image
2. Isolation de la partie rouge
3. Soustraction de la luminosité
4. Application d'un filtre médian
5. Binarisation de l'image

6. Détection de forme ronde

L'algorithme, codé sous Matlab, a la forme suivante. L'intégralité du code est disponible en annexe A. Il utilise l'Add-on *Image Acquisition Toolbox by MathWorks*. L'installation de cet Add-on est requise pour pouvoir utiliser les codes présentés.

```
1 image_rgb = step(entree_video); % Acquisition d'une image
2 img_rouge = imsubtract(image_rgb(:,:,1), rgb2gray(image_rgb)); % On extrait la partie rouge
   de l'image en supprimant la contribution de la luminosité
3 img_rouge = medfilt2(img_rouge, [3 3]); % On filtre l'image en moyennant les carré de 3x3
4 bw_image = im2bw(img_rouge, seuil_rouge); % On convertit l'image en noir et blanc
5 [centers,radii] = imfindcircles(bw_image,[6 18]); % On cherche les cercles dont le rayon
   est compris dans l'intervalle
6
7 if (~isempty(centers)) % Si une balle est detectee, on recupere les coordonnees
8     x= centers(1); y = centers(2);
9     theta = x*coef_pixel-angle_max_x;
10    phi = -y*coef_pixel+angle_max_y;
11    rayon = int16(radii(1));
```

Listing 1: Traitement de l'image pour la localisation de la balle

L'algorithme complet est disponible dans le dossier `Stage_CRME/Vision/detection_balle.m`. En exécutant le programme, Matlab va ouvrir la caméra définie à la ligne 34 :

```
1 entree_video = imaq.VideoDevice('winvideo', 1, 'YUY2_320x240', ... % Video en entree
2     'ReturnedColorSpace', 'rgb');
```

Listing 2: Définition de l'entrée vidéo

Par défaut, il s'agit de la caméra frontale du PC qui exécute le programme (s'il en est équipé). Si une erreur survient à ce moment, il s'agit de vérifier que les paramètres de la caméra du PC sont bien ceux que vous demandez dans la fonction `VideoDevice`. Pour cela, il faut exécuter la commande suivante.

```
1 >>> imaqtool
```

Listing 3: Ouverture de la fenêtre de paramétrage

La fenêtre qui s'ouvre permet de sélectionner le format et la caméra souhaitée. Il s'agit uniquement de copier coller le contenu de la sous-fenêtre `Session log` dans `imaq.videoDevice()`.

L'algorithme a été testé avec la caméra Tecknet HD720P Webcam. Les tests ont été satisfaisants. La figure 16 montre un exemple d'image sur laquelle la balle a été lancée et détectée par la caméra. Comme prévu par les calculs de la partie 3.3.1, pour une caméra en 30 FPS, il y a un flou de mouvement qui fait que la balle n'est pas détectée assez précisément. Il faut donc remplacer la caméra Tecknet par une nouvelle (deux nouvelles en fait, pour faire de la stéréovision) qui répond aux exigences énoncées précédemment.

La nouvelle caméra envisagée était la PS3 Eye, dont les caractéristiques sont adaptées au problème (voir le fichier `PS3_eye.pdf`). Cependant, la PS3 Eye a été initialement créée pour fonctionner avec une PS3 et non comme une Webcam. Après un long travail sur la compatibilité, j'ai uniquement réussi à avoir un aperçu de la caméra sur le logiciel `CL Eye Test`. Windows détecte le périphérique comme étant un "Périphérique d'acquisition d'image" (au même titre qu'un scanner). Or, le programme Matlab fonctionne avec des Webcam, détectés comme étant des "appareils photo". Ainsi, Matlab n'arrive pas à travailler avec la PS3 Eye.

Il y a deux axes possibles de travail pour la partie vision :

- continuer à utiliser la PS3 Eye et travaillant sur la recherche du pilote qui permet de la détecter comme une entrée vidéo utilisable dans Matlab

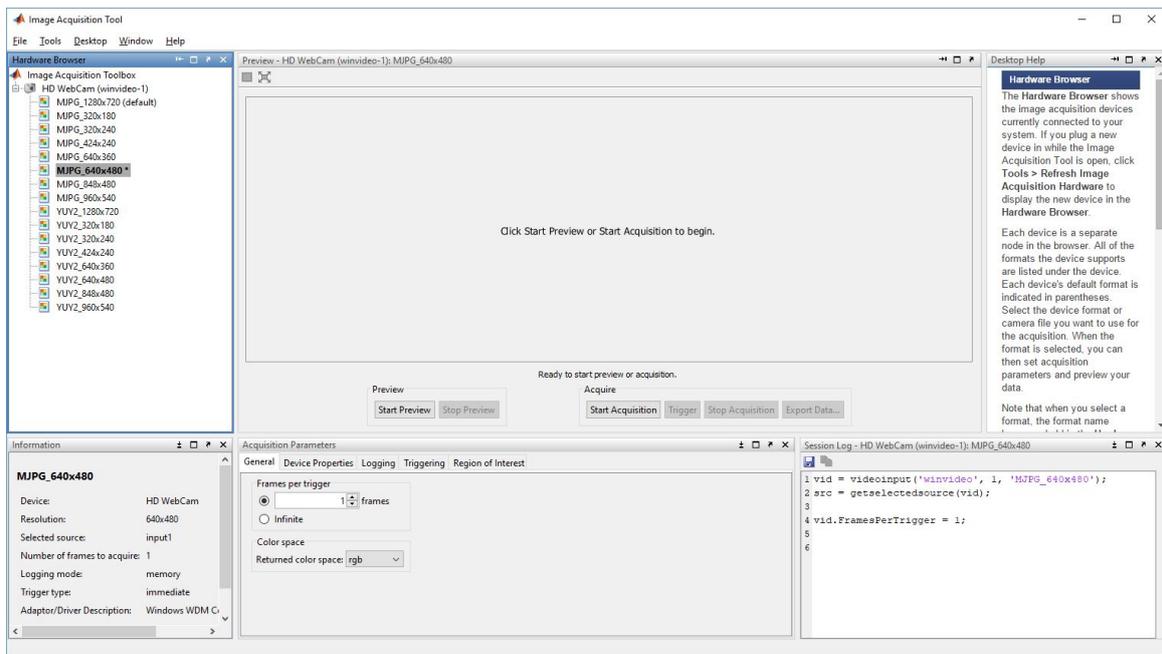


FIGURE 15 : Fenêtre de paramétrage de la caméra



FIGURE 16 : Balle détectée par l'algorithme de vision

- chercher une nouvelle caméra avec des caractéristiques similaires, avec laquelle il est possible de travailler sur Matlab

Ensuite, à partir des deux coordonnées d'image, et des calculs menés dans la partie 3.3, il faudra compléter l'algorithme pour que les valeurs de sortie correspondent aux coordonnées dans le repère de la table.

5 Conclusion du projet

Les trois axes développés dans ce rapport correspondent aux aspects qui doivent être intégrés au bras robot à terme. La réalisation du modèle 3D est sur le point d'être terminée. Il reste une solution technique à trouver pour la courroie, et pour la fixation à la table. Pour la fixation à la table, on peut s'inspirer de ce qui a été développé dans la version 1 du modèle (voir dossier). Ensuite, les actionneurs sont choisis et les bases de leur programmation ont été posées. Mais

il reste encore un travail sur l'intégration de la bibliothèque Arduino sur Matlab. Ensuite, il sera question d'intégrer les résultats de l'approche théorique à la commande de ces moteurs. Concernant les caméras, un algorithme fonctionnel est disponible. Il reste encore à trouver une caméra qui convienne pour fonctionner avec Matlab. Il faudra également adapter ce code pour qu'il puisse gérer deux entrées vidéo, et envoyer les résultats du traitement d'images à l'algorithme de commande des moteurs.

6 Le stage de recherche dans la construction de mon projet professionnel

6.1 Choix du stage

Le stage d'application était pour moi l'occasion d'acquérir une première expérience dans le domaine de la recherche. C'était mon objectif principal. Ensuite, j'ai choisi ce stage également pour avoir une première expérience à l'étranger. J'ai donc cherché dans la base de données des stages réalisés par les élèves centraliens et j'ai trouvé un contact au sein du CHU de Sainte Justine, à Montréal, au Canada. Ce contact m'a permis d'obtenir ce stage, qui conciliait mes différentes attentes.

En effet, je m'intéresse particulièrement au domaine de l'électronique, de la programmation et de la mécanique. La fabrication d'un bras robot était donc un sujet de stage très approprié. J'ai pu utiliser les différents logiciels que je manipulais en cours pour répondre aux problématiques du sujet. Par exemple : Matlab, Catia et Arduino. J'ai également pu en découvrir d'autres et me former dessus : Simplify 3D (équivalent de Cura), et le logiciel intégré des moteurs utilisés : ClearPath-MSP.

J'ai d'abord contacté de façon spontanée et par mail monsieur Maxime Raison. Après quelques échanges, il m'a fait savoir qu'un sujet de stage attendait son candidat. Comme le sujet m'intéressait beaucoup et que les modalités aussi, j'ai décidé de confirmer ma candidature. Le stage n'étant pas rémunéré et ayant déjà un projet de césure, j'ai dû fixer la durée de mon stage à 14 semaines.

6.2 Aspects marquants

La grande particularité de mon stage a été que j'ai travaillé la plupart du temps en autonomie et seul. J'avais donc une grande liberté sur l'orientation que je donnais au projet. j'avais à disposition une imprimante 3D et un ordinateur de bureau. J'avais également accès à une salle avec une table de ping-pong pour réaliser mes essais. A posteriori, je pense que la principale qualité requise pour ce stage était l'autonomie et la prise d'initiative. Je pense avoir répondu à ces attentes, et mon tuteur m'a finalement félicité pour le travail que j'ai réalisé.

Un autre aspect marquant du stage était le contact direct que je pouvais avoir avec le personnel de l'hôpital et les patients. En effet, le pavillon de recherche étant intégré au centre, je croisais quotidiennement les patients, les infirmiers, les médecins, et en règle générale, l'ensemble du personnel du centre. L'intérêt est de pouvoir être proche des problématiques soulevées par le projet. J'avais sous les yeux les potentiels futurs bénéficiaires du travail que je réalisais.

6.3 Valorisation

Cette expérience m'a apporté une certaine maturité en me plongeant dans un monde auquel je n'étais pas familier. J'ai pu rencontrer des familles, des enfants traités au CRME et travailler ensuite sur ce qui pourrait devenir un bras robotisé pour ces derniers. D'un point de vue humain c'est une expérience que je jugeais nécessaire d'avoir avant la fin de mes études.

Me posant encore la question à propos d'un doctorat, j'ai pu me glisser dans la peau d'un chercheur et découvrir les aspects d'une telle activité. D'abord, la liberté dont jouissent les chercheurs en université est appréciable. Il est ensuite appréciable de pouvoir côtoyer des chercheurs pointus dans leur domaine. Les échanges sont enrichissants. Néanmoins, certains me décrivaient

certains mauvais travers qu'il peut y avoir dans la course aux publications.

D'un point de vue technique, j'ai pu travailler sur un projet d'envergure, à plein temps. Je me suis beaucoup perfectionné en modélisation 3D ainsi qu'en impression 3D. J'ai affiné mes connaissances en mécanique grâce à tous les aspects que couvrait ce sujet.

7 Conclusion du stage

A différents égards, ce stage m'a beaucoup avancé dans la construction de mon projet professionnel. Envisageant un doctorat, je me devais d'avoir une expérience dans la recherche. J'ai également profité de ce stage pour voyager, et me confronter à une autre culture, à un autre pays. La discrétion et l'altruisme que j'avais cultivés durant mes précédentes travaux dans le social (accompagnant vacance pour personnes en situation de handicap mental, cours particuliers bénévole à des jeunes en difficulté) m'ont beaucoup servi. Je me sens à l'aise dans ce genre de travail et c'est donc logiquement que je fais actuellement mon année de césure au sein de la Brigade de Sapeurs-Pompiers de Paris.

A Code Matlab pour la détection de la balle

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Nom du prgramme : Traquer une balle de ping pong %
3 % Auteur : Quentin Gallouedec %
4 % Description : Detecter et suivre une balle de ping pong orange %
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 %% Initialisation
8
9 seuil_rouge = 0.25 ; % Seuil rouge pour la mise en noir et blanc
10
11
12 %% Parametre camera
13
14 nb_pixels_x = 320 ; % Nombre de pixels sur la largeur de l'image
15 nb_pixels_y = 240 ; % Nombre de pixels sur la hauteur de l'image
16 angle_max_x = 0.42 ; % Ouverture = [-angle_max ; +angle_max] en radians (notice ou mesure)
17
18 coef_pixel = 2*angle_max_x/nb_pixels_x ; % Correspond a la l'angle auquel correspond une
19 pixel
20 angle_max_y = nb_pixels_y/2*coef_pixel ;
21
22 %% Definition des fonctions
23
24 initialisation();
25
26 %% Boucle de travail
27
28 nb_images = 0; % Initialisation du nombre d'images
29 nb_images_max = 10;
30
31 while(nb_images < nb_images_max)
32     image_rgb = step(entree_video); % Acquisition d'une image
33     img_rouge = imsubtract(image_rgb(:, :, 1), rgb2gray(image_rgb)); % On extrait la partie
34     rouge de l'image
35     img_rouge = medfilt2(img_rouge, [3 3]); % On filtre l'image en moyennant les carre de 3
36     x3
37     bw_image = im2bw(img_rouge, seuil_rouge); % On convertit l'image en noir et blanc
38     [centers, radii] = imfindcircles(bw_image, [6 18]);
39
40     if (~isempty(centers)) % Si une balle est detectee, on affiche les coordonnees
41         x= centers(1); y = centers(2);
42         theta = x*coef_pixel-angle_max_x;
43         phi = -y*coef_pixel+angle_max_y;
44         rayon = int16(radii(1));
45
46         image_rgb = step(dessine_cercle, image_rgb, [x,y,rayon]); % On affiche un cercle
47         autour de la balle
48         image_rgb = step(textes_centres, image_rgb, [int16(theta),int16(phi),rayon], [10
49         10]); % On affiche ses coordonnees
50         image_rgb = step(marqueur_balle, image_rgb, [x y]); % On dessine une croix au
51         centre
52     end
53     nb_images = nb_images + 1; % On passe a l'image suivante
54     step(sortie_video, image_rgb); % On envoie l'image en sortie
55     step(enregistrer_video, image_rgb); % On enregistre la video
56
57 end
58 %% Effacement des donnees video
59
60 release(sortie_video); % On libere les ports d'entree et de sortie
61 release(entree_video);
62 release(enregistrer_video);
63
64 clc;
```

Listing 4: Algorithme Matlab de détection de balle

B Logiciels utilisés et version

- Arduino 1.16.2
- Matlab 2018a
- MATLAB Support Package for Arduino Hardware 18.1.1
- Image Acquisition Toolbox
- ClearPath-MSP V1.7.12

Références

- [1] Forpheus, technologie to create the future. <https://www.omron.com/innovation/forpheus.html>. Accessed : 2019-01-07.
- [2] Kariane Bourassa. Un bras qui change des vies. *Le Journal de Montréal*, 08 2016. <https://www.journaldemontreal.com/2016/08/06/un-bras-qui-change-des-vies>, Accessed : 2018-08-01.
- [3] Tim Fryer. Photo essay [automation robotics]. *Engineering & Technology*, 12(4) :40–41, 2017.
- [4] The International Table Tennis Federation, Chemin de la Roche 11, 1020 Renens/Lausanne, Switzerland. *Handbook*, 2018.
- [5] BK Manou, J Bombo, AD Alloh, A Coulibaly, J Kouakou, and B Nandjui. Amputation de membres inférieurs et appareillage : expérience du centre d'appareillage «vivre debout» à abidjan. *Journal de Réadaptation Médicale : Pratique et Formation en Médecine Physique et de Réadaptation*, 29(4) :152–157, 2009.