MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

Institut Supérieur Polytechnique Privé

****

Projet de Fin d'Année

En vue de l'obtention du Diplôme National d'Ingénieur en Mécatronique

Elaboré par :

Tanny DAMET et Soufien GHRIBI

ETUDE ET CONCEPTION D'UN BRAS DESSINATEUR

Encadré par

Monsieur BEN ROMDHANE Karim

Année universitaire
2019 - 2020

Remerciements

Nous tenons particulièrement à exprimer toute notre reconnaissance à notre Encadrant Monsieur Karim Ben Romdhane qui s'est toujours montré à l'écoute et très disponible tout au long de nos recherches. Nous le remercions de nous avoir encadrés, orientés, aidés et conseillés.

Nous souhaitons également adresser nos remerciements les plus sincères aux personnes qui, par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé nos réflexions et nous ont apportés leur aide pour mener à bien ce projet et à l'élaboration du rapport.

À tous ces intervenants, nous présentons nos remerciements, notre respect et notre gratitude.

Table des matières

Remerciements i
Table des matières ii
Liste des figures iv
Liste des tableaux vi
Liste des abréviations et symboles vii
Introduction générale 1
CHAPITRE I. PRESENTATION DU PROJET 2
I.1 Introduction 2
I.2 Objectif 2
I.3 Définition et Début 3
I.4 Etude de l'existant 4
I.5 Conclusion 7
CHAPITRE II. ANALYSE FONCTIONNELLE 8
II.1 Introduction 8
II.2 Représentation fonctionnelle du système 8
II.3 Etude du besoin 9
II.4 Analyse fonctionnelle de besoin 10
II.5 Elément de cahier des charges 13
II.6 Analyse fonctionnelle technique 13
II.7 Conclusion 15
CHAPITRE III. ETUDE 16
III.1 Introduction 16
III.2 Choix de la technologie 16
III.3 Etude du budget 21
III.4 Modélisation géométrique 22
III.5 Traitement d'image 27
III.6 Conclusion 32
CHAPITRE IV. CONCEPTION ET SIMULATION 33
IV.1 Introduction 33
IV.2 Conception 33
IV.3 Simulation 36
IV.4 Conclusion 38
Conclusion générale 39
Bibliographie 40
Annexes 41

Annexe 1 Installation de OpenCV 2.4.10 sur un Raspberry Pi B 41
Annexe 2 Modélisation 43
Annexe 3 Code 46
Annexe 4 Dessins techniques des pièces du bras 49

Liste des figures

Figure 1 – Robographe 4
Figure 2 – Robot de gravure 4
Figure 3 – BeachBot 5
Figure 4 – Drones dessinateurs 5
Figure 5 – Robot E-David 6
Figure 6 – Line-us 6
Figure 7 – Paul le robot dessinateur 7
Figure 8 – Diagramme SADT : Description fonctionnelle générale 8
Figure 9 – Diagramme SADT niveau A0 9
Figure 10 - Bête à cornes 10
Figure 11 - Frontière d'étude 10
Figure 12 - Diagramme Pieuvre 11
Figure 13 – Diagramme de FAST 14
Figure 14 – Filament PLA 16
Figure 15 – Dimensionnement du bras 17
Figure 16 – Servomoteur S3003 19
Figure 17 – Servomoteur MG995 19
Figure 18 – Carte Raspberry Pi 4 et ses caractéristiques 20
Figure 19 – Système d'alimentation de la carte 21
Figure 20 – Représentation du système avec les axes 22
Figure 21 – Modélisation du système avec les axes 23
Figure 22 – Bras robotique en position initiale 26
Figure 23 – Bras robotique en mouvement 26
Figure 24 – Traitement d'image exemple 1 27
Figure 25 – Image en gris (à gauche) - Image floue (à droite) 28
Figure 26 – Image floue (à gauche) – Gradient intensifié (à droite) 29
Figure 27 – Gradient intensifié (à gauche) – Suppression non maximal (à droite) 29
Figure 28 – Suppression non maximal (à gauche) – Résultat du seuil (à droite) 30
Figure 29 – Résultat du seuil (à gauche) – Résultats du processus d'hystérésis (à droite) 31
Figure 30 – Traitement d'image exemple 21 31
Figure 31 – Traitement d'image exemple 22 31
Figure 32 – Traitement d'image exemple 3 32

Figure 33 – Vue de côté du robot 33
Figure 34 – Vue de dessus du robot..... 33
Figure 35 – Vue de dessous du robot..... 33
Figure 36 – Vue de derrière du robot..... 33
Figure 37 – Vue de face du robot 33
Figure 38 – Vue isométrique 33
Figure 39 – Interface de programmation en langage Simscape 1..... 37
Figure 40 – Interface de programmation en langage Simscape 2..... 37
Figure 41 – Interface de programmation en langage Simscape 3..... 38

Liste des tableaux

Tableau 1 : Caractérisation des fonctions de service 1
Tableau 2 : Caractéristiques des servomoteurs 1
Tableau 3 : Etude du budget 1
Tableau 4 : Paramètres géométriques du robot 2
Tableau 5 – Description des pièces 2

Liste des abréviations et symboles

ATILF	Analyse et traitement informatique de la langue française
SADT	Structured Analysis and Design Technics
FP	Fonctions Principales
FC	Fonctions Complémentaires et/ou Contraintes
F	Flexibilité
FAST	Function analysis system technique
FT	Fonctions techniques
PLA	Polylactic acid
RAM	Random Access Memory
USB	Universal Serial Bus
HDMI	High Definition Multimedia Interface
GO	Giga Octet
Dt	Dinar Tunisien
Ddls	Degré de liberté
DHM	Denavit Hatenberg Modifiée
RGB	Red Green Blue
CAO	Conception Assistée par Ordinateur
3D	Trois dimensions
STL	Stereolithography
µ	Accélération de la pesanteur
N	Newton
Kg	Kilogramme
N/Kg	Newton par Kilogramme
Cm	Centimètre
N.cm	Newton Centimètre
Kg.cm	Kilogramme Centimètre
g	gramme
mm	millimètre
V	Volt
A	Ampère
s/°	Seconde par degré
°C	Degré Celsius
°	Degré

Introduction générale

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots. Le robot étant *« un appareil effectuant, grâce à un système de commande automatique à base de micro-processeur, une tâche précise pour laquelle il a été conçu dans le domaine industriel, scientifique, militaire ou domestique »* selon ATILF.

Le besoin de se divertir et de s'amuser a amené à la fabrication de différents types de robots chacun dans un domaine particulier : Le dessin, la musique et la dance. Les robots artistiques ont été créés pour répondre au besoin de se divertir.

Dans le cadre de notre projet de fin de la quatrième année de Mécatronique, nous nous intéressons au développement d'un bras robotisé. Composé de trois articulations rotoïdes, il sera capable de réaliser des portraits imitant les gestes d'un bras humain. Pour assurer ces tâches convenablement, le bras va être intégré dans un système global qui va lui fournir toutes les ressources nécessaires à son fonctionnement.

Ce document développera les diverses étapes de notre mission :

Le premier chapitre est un chapitre introductif. Nous présenterons le projet dans sa globalité en énonçant son objectif et en mettant en avant l'étude de l'existant.

Le deuxième chapitre nous aidera à découvrir l'environnement de travail de notre système grâce à son analyse fonctionnelle. On détaillera clairement le besoin du projet ainsi que son fonctionnement.

Pour le troisième chapitre, nous réaliserons une étude théorique. Aussi, en se basant sur notre objectif du projet, on proposera notre solution pour le choix de la technologie.

Concernant le chapitre quatre, on s'intéressera à notre solution proposée, composée de deux parties, la première sera consacrée à la conception sur Solidworks et la deuxième partie qui adaptera la conception avec une simulation sur Matlab.

A terme, une conclusion clôturera ce rapport. On expliquera dans cette partie l'état d'avancement de notre système.

Chapitre 1 : Présentation du projet

I.1 Introduction

Ce chapitre vise à situer le projet dans son contexte général. Nous allons donc d'abord énoncer l'objectif du projet avant de rappeler la définition du robot en général ainsi qu'une brève historique. Par la suite nous présenterons les différents types de robots artistes dessinateurs qui existent déjà avant de conclure.

I.2 Objectif

Étant dans un projet universitaire limité en moyens et en temps, nous avons décidé de restreindre notre projet de fin d'année, en sélectionnant les solutions à développer parmi toutes les possibilités permises.

Ainsi, nous avons projeté dans un premier temps de réaliser un système constitué principalement d'un bras articulé à trois degrés de liberté capable de réaliser sur une feuille de papier, le portrait extrait d'une image importée sur son unité de calcul via Bluetooth imitant ainsi le dessin de l'homme.

Notre objectif n'est pas de reproduire parfaitement le portrait, mais plutôt de simuler l'acte de dessiner par des mouvements répétitifs du bras robotique afin de tracer des lignes sur le papier dessin.

Ajouté à ce cœur de projet, nous entreprenons l'étude d'une alimentation électrique adaptée, le choix d'une motorisation du bras simple à mettre en place ainsi que la réalisation d'un programme de traitement de l'image capturée suivi de la génération du portrait de manière la plus fidèle et rapide possible.

Ce qui sera l'aboutissement du projet, à même de démontrer tout l'intérêt de la robotique et du traitement d'image.

I.3 Définition et Début

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) conçu pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines. La conception de ces systèmes est l'objet d'une discipline scientifique, branche de l'automatisme nommé robotique.

Le terme robot apparaît pour la première fois dans la pièce de théâtre (science-fiction) R. U. R. (Rossum's Universal Robots), écrite en 1920 par l'auteur Karel Capek. Le mot a été créé par son frère Josef à partir du mot tchèque « robota » qui signifie « travail, besogne, corvée ».

Les premiers robots industriels apparaissent, malgré leur coût élevé, dans le début des années 1970. Ils sont destinés à exécuter certaines tâches répétitives, éprouvantes ou toxiques pour un opérateur humain : peinture ou soudage des carrosseries automobiles. Aujourd'hui, l'évolution de l'électronique et de l'informatique permet de développer des robots plus précis, plus rapides ou avec une meilleure autonomie. Industriels, militaires ou spécialistes chirurgicaux rivalisent d'inventivité pour mettre au point des robots assistants les aidant dans la réalisation de tâches délicates ou dangereuses. Dans le même temps apparaissent des robots à usages domestiques : aspirateur, tondeuses, etc.

L'usage du terme « robot » s'est galvaudé pour prendre des sens plus larges : automate distributeur, dispositif électromécanique de forme humaine ou animale, logiciel servant d'adversaire sur les plateformes de jeu, bot informatique.

Mais, à côté de ces robots « utiles », nous commençons à découvrir des robots « artistes » conçus principalement dans le but de se divertir : ils sont capables de dessiner, de peindre, d’écrire, de danser, de jouer la comédie ou même de pratiquer un instrument de musique.

I.4 Etude de l'existant

À l'origine, les robots ont été installés dans les industries, ce qui permet de faire des tâches répétitives avec une précision constante.

Par la suite, Les robots ont envahi les domaines de loisirs, dessin et peinture.

Voici donc quelques exemples de robots artistes dessinateurs :

- En premier lieu, nous avons le robographe : C'est un robot qui produit un texte ou un dessin en imitant les gestes d'un humain en utilisant la capture de mouvement de l'utilisateur. La gestuelle de sa main est alors captée et renvoyée au robographe qui restitue fidèlement le dessin dans l'ordre chronologique.

Figure 1 – Robographe

- Nous avons aussi le robot de gravure qui fonctionne comme une machine CNC, réalisant des gravures sur du bois, du fer ou même sur du plastique.

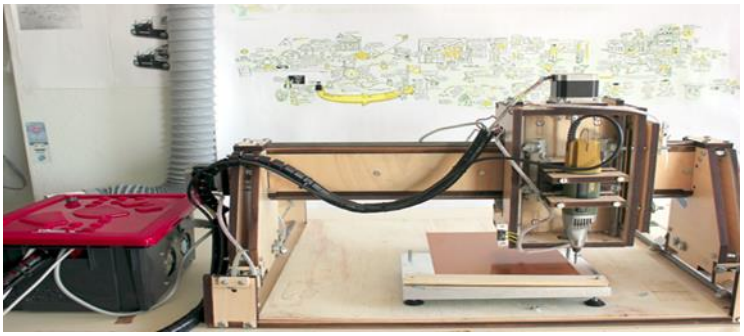


Figure 2 – Robot de gravure

- Nous avons ensuite le BeachBot qui est un robot autonome qui peut créer de l'art du sable à grande échelle. Le robot sera déployé sur une plage publique pour émerveiller les baigneurs qui passent. Non seulement l'image finale est importante, mais tout le processus de dessin offrira un spectacle magique exceptionnel.

Figure 3 – BeachBot

- Il existe aussi les drones dessinateurs. Le célèbre artiste KATSU présente une série de graffitis innovants car pour la première fois, l'artiste américain n'a pas réalisé ses peintures à la main mais à l'aide d'un drone.

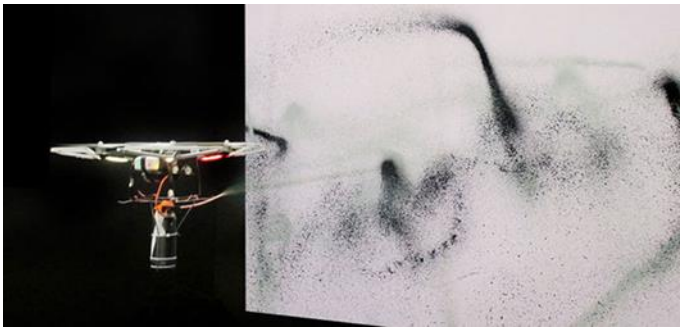


Figure 4 – Drones dessinateurs

- Le robot E-David se présente sous la forme d'un bras robotisé, capable de peindre des toiles qu'il est difficile de différencier d'œuvres réalisées de main humaine. À l'origine, il s'agissait d'un robot industriel, destiné à souder les carrosseries de voitures.

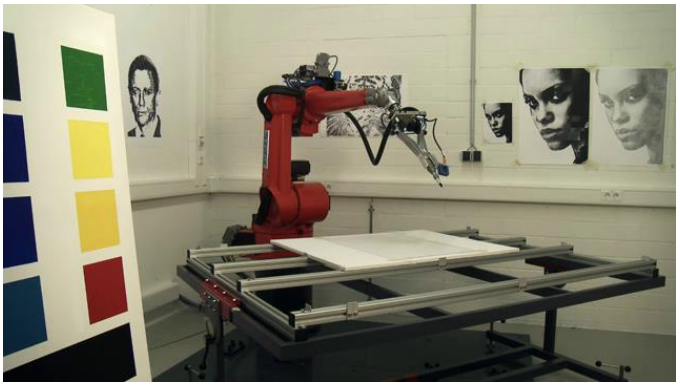


Figure 5 – Robot E-David

- Line-us est un petit bras robot qui dessine avec un stylo sur papier tout ce qu'on dessine à l'écran. Le petit robot original imite l'écriture sur une tablette ou un ordinateur en temps réel, copiant les mouvements des mains, à travers ses croquis, exactement dans le même ordre. Le petit robot peut également dessiner des dessins enregistrés.

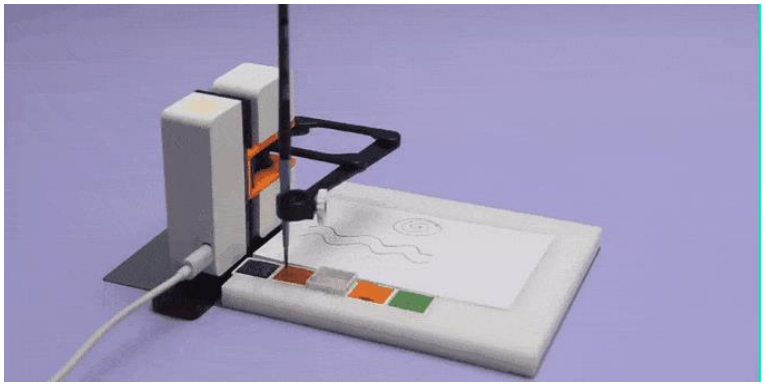


Figure 6 – Line-us

- Finalement nous avons Paul le robot dessinateur qui représente un peu l'objet d'inspiration de notre projet. Paul est un robot constitué d'un bras articulé avec une caméra numérique motorisée capable de réaliser ses propres dessins. C'est un robot portraitiste.

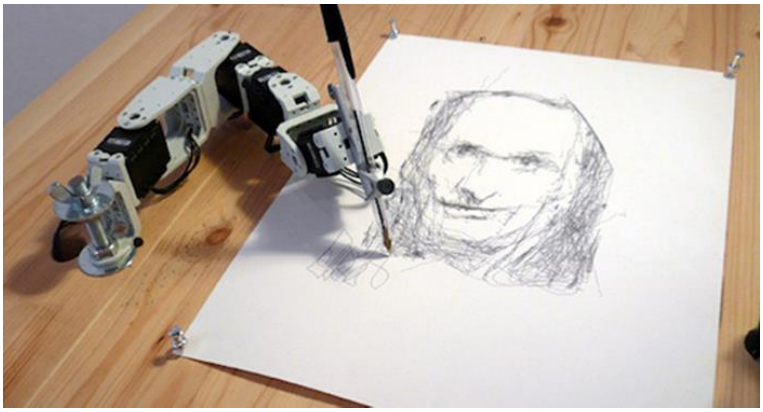


Figure 7 – Paul le robot dessinateur

I.5 Conclusion

Tout au long de ce chapitre, nous avons mis notre projet dans son cadre général, à savoir ses objectifs, une définition du robot ainsi que ses débuts, puis nous avons présenté l'étude de l'existant. Dans le prochain chapitre, nous allons nous intéresser à l'analyse fonctionnelle du projet.

Chapitre 2 : Analyse fonctionnelle

II.1 Introduction

Dans ce chapitre, nous nous intéresserons à l'analyse fonctionnelle du projet. Par différents schémas et tableaux, nous allons d'abord donner une représentation fonctionnelle du système, puis nous poursuivrons avec une étude du besoin ainsi que l'analyse fonctionnelle du besoin. Nous présenterons par la suite le cahier de charge et finirons avec l'analyse fonctionnelle technique.

II.2 Représentation fonctionnelle du système

Le diagramme SADT permet la modélisation fonctionnelle du système. C'est une méthode d'analyse structurée qui utilise la schématisation.

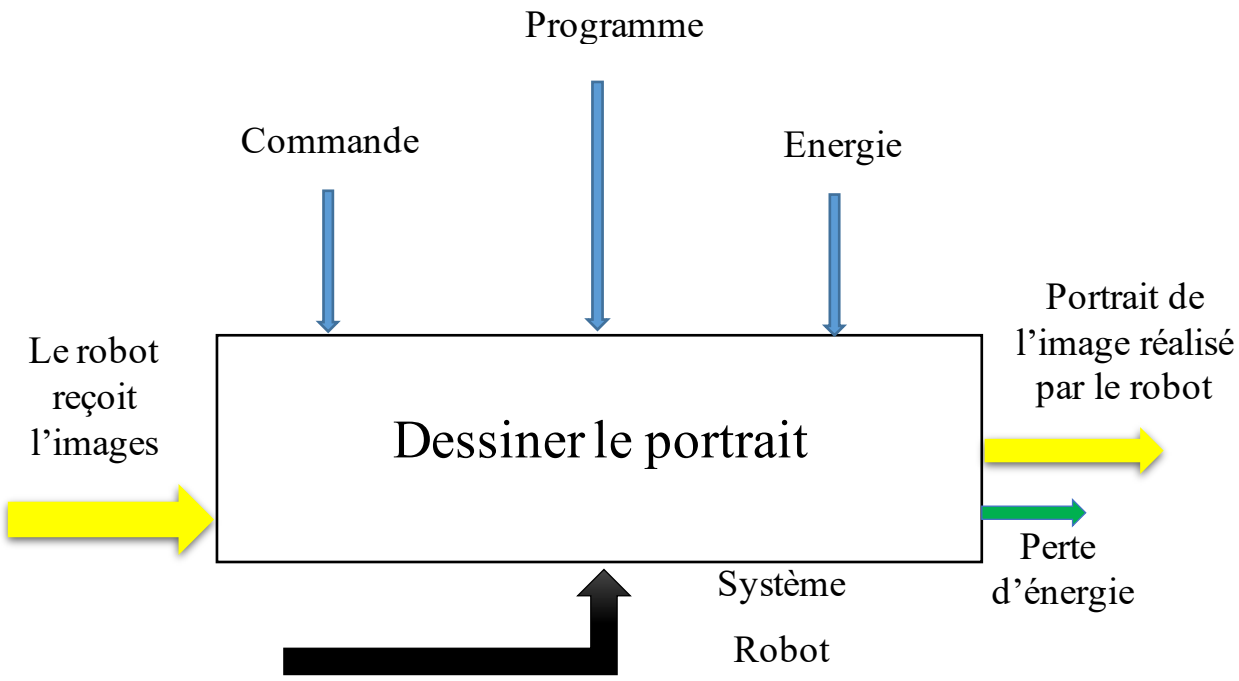


Figure 8 – Diagramme SADT : Description fonctionnelle générale

Dans le diagramme qui suit, nous essayerons de décortiquer les différentes fonctions essentielles à notre système afin de mieux l'étudier.

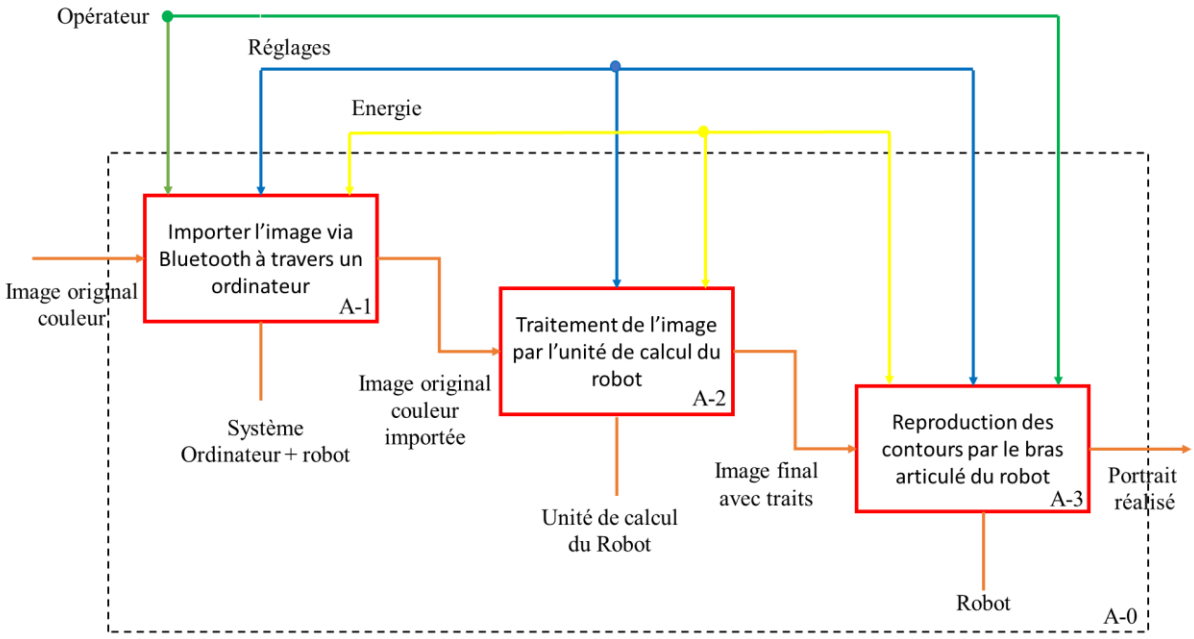


Figure 9 – Diagramme SADT niveau A0

II.3 Etude du besoin

L'analyse du besoin impose la démarche suivante :

Les robots articulés sont très utilisés dans le domaine de l'industrie. Cependant, de plus en plus, ils envahissent le domaine de loisirs, dessin et peinture.

Toutefois le besoin principal de notre projet est le divertissement. Nous ne cherchons pas nécessairement à reproduire un portrait parfait. C'est ce le côté imitation de la gestuelle humaine qui nous intéresse le plus.

Ainsi pour exprimer avec précision et délimiter notre domaine d'étude nous allons répondre à ces trois questions à travers le diagramme bête à cornes.

- A qui rend-t-elle ce service (A quoi ?)
  - ➔ A l'utilisateur
- Sur qui (sur quoi) agit-elle ?
  - ➔ Sur un papier de dessin
- Dans quel but ?
  - ➔ Réaliser des portraits imitant les gestes d'un bras humain

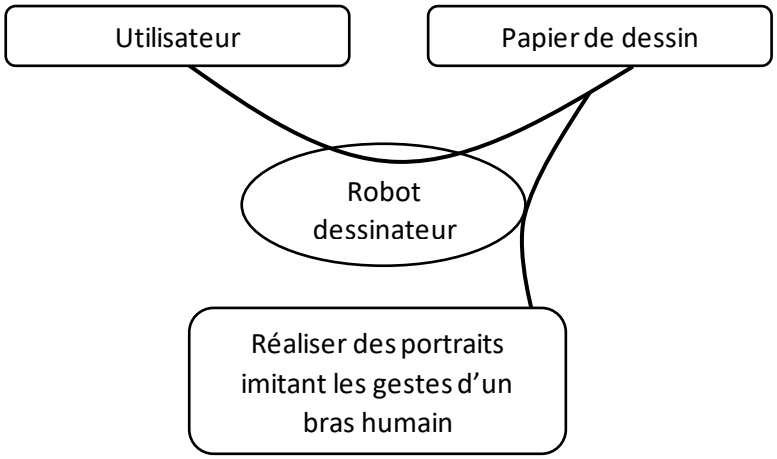


Figure 10 - Bête à cornes

II.4 Analyse fonctionnelle de besoin

❖ Frontière d'étude

Nous allons tout d'abord essayer de délimiter les frontières de notre système pour identifier les différents éléments en interaction avec lui.

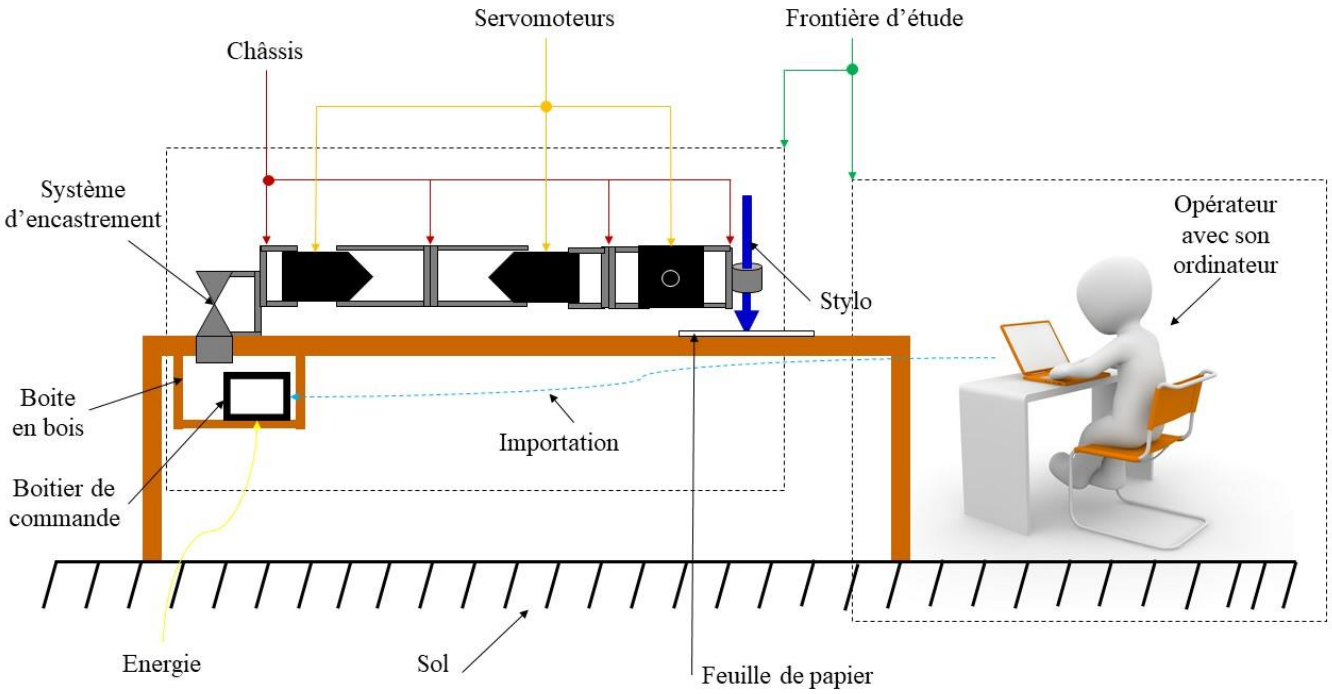


Figure 11 - Frontière d'étude

❖ Eléments d'environnement

- Opérateur
- Ordinateur
- Energie de commande
- Sol

- Bras robotisé (servomoteurs + châssis + Système d'encastrement)
- Normes
- Unités de travail (papier + stylo + table + boite de soutien en bois)
- Boitier de commande

❖ Identification des fonctions de service : Diagramme Pieuvre

Le Diagramme Pieuvre est un outil d'analyse du besoin représentant graphiquement les interactions d'un produit/service avec son environnement. Il met en évidence les relations entre les différents Eléments du Milieu Extérieur et le produit/service au travers de fonctions. :

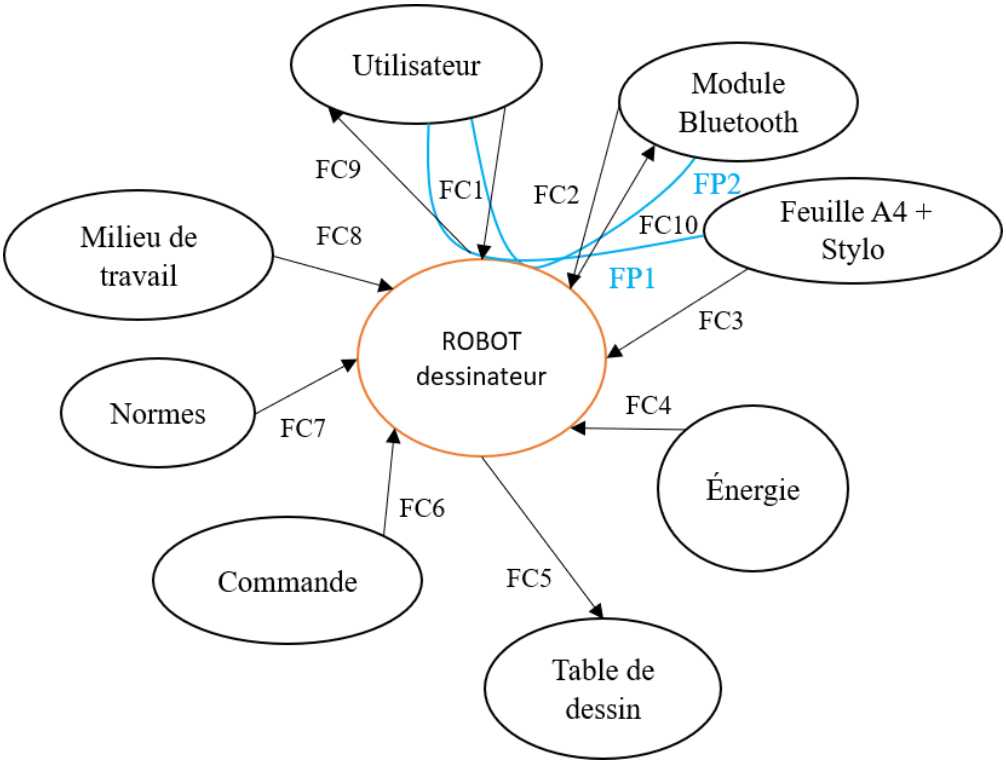


Figure 12 - Diagramme Pieuvre

- ❑ FP1 : Réaliser les portraits sur image sur une feuille avec un stylo
- ❑ FP2 : Importer l'image via un module Bluetooth
- ❑ FC1 : Intervient en cas de défaut
- ❑ FC2 : Le module Bluetooth doit être adapté au robot
- ❑ FC3 : Doit être adapté à la feuille et au stylo en dimension
- ❑ FC4 : S'adapter à la source d'énergie disponible
- ❑ FC5 : Assurer un environnement stable au robot

- FC6 : S'adapter aux commandes nécessaires à son fonctionnement
- FC7 : Respecter les normes de sécurité, d'esthétique, etc.
- FC8 : Limiter les débordements
- FC9 : Réaliser le portrait envoyé par l'utilisateur
- FC10 : Doit être doté d'un module Bluetooth

Fonction	Critères	Flexibilité
FP1 : Réaliser les portraits sur image sur une feuille A4 avec un stylo	Vitesse	F1
	Précision	F2
	Stabilité	F1
FP2 : Importer l'image via un module Bluetooth	Format	F1
	Rapidité	F3
FC1 : Intervient en cas de défaut	Surface d'accès Outillage standard	F0
FC2 : Doit être doté d'un module Bluetooth	Caractéristiques	F0
FC3 : Doit être adapté à la feuille A4 et au stylo en dimension	Dimensions	F1
FC4 : S'adapter à la source d'énergie disponible	Source électrique	F1
FC5 : Assurer un environnement stable au robot	Equilibre	F1
FC6 : S'adapter aux commandes nécessaires à son fonctionnement	Programmes	F0
FC7 : Respecter les normes de sécurité, d'esthétique, etc.	Esthétique Hygiène et Sécurité	F1
FC8 : Limiter les débordements	Singularité	F0

FC9 : Réaliser le portrait de l'utilisateur	Fidélité de reproduction	F2

Tableau 1 : caractérisation des fonctions de service

II.5 Eléments de cahier des charges

Le projet consiste à l'étude et la conception d'un bras articulé capable de réaliser des portraits de visage capturé au préalable sur une image importé dans son unité de calcul via un ordinateur portable imitant un bras humain. Le mécanisme, objet d'étude, doit être fiable et doit permettre de :

- Recevoir l'image par Bluetooth
- Traiter l'image afin d'obtenir une forme plus simple à réaliser
- Agencer les mouvements en sorte d'obtenir le résultat attendu
- S'adapter à la source d'énergie disponible
- Respecter les normes en vigueur

Caractéristiques et exigences :

Ce robot :

- Doit avoir des traits avec un bras humain
- Doit solide et facile à manipuler
- Ne doit pas trop coûter
- Ne doit pas trop consommer d'énergie

II.6 Analyse fonctionnelle technique

La méthode FAST aide à penser au problème de façon objective et permet de déterminer la portée du projet en illustrant les relations logiques entre les fonctions. La disposition des fonctions de façon logique dans un diagramme FAST permet aux participants de déterminer toutes les fonctions requises. Le diagramme FAST peut servir à vérifier si, et à illustrer comment, une solution proposée atteint les besoins du projet, ainsi qu'à déterminer les fonctions inutiles, dédoublées ou omises.

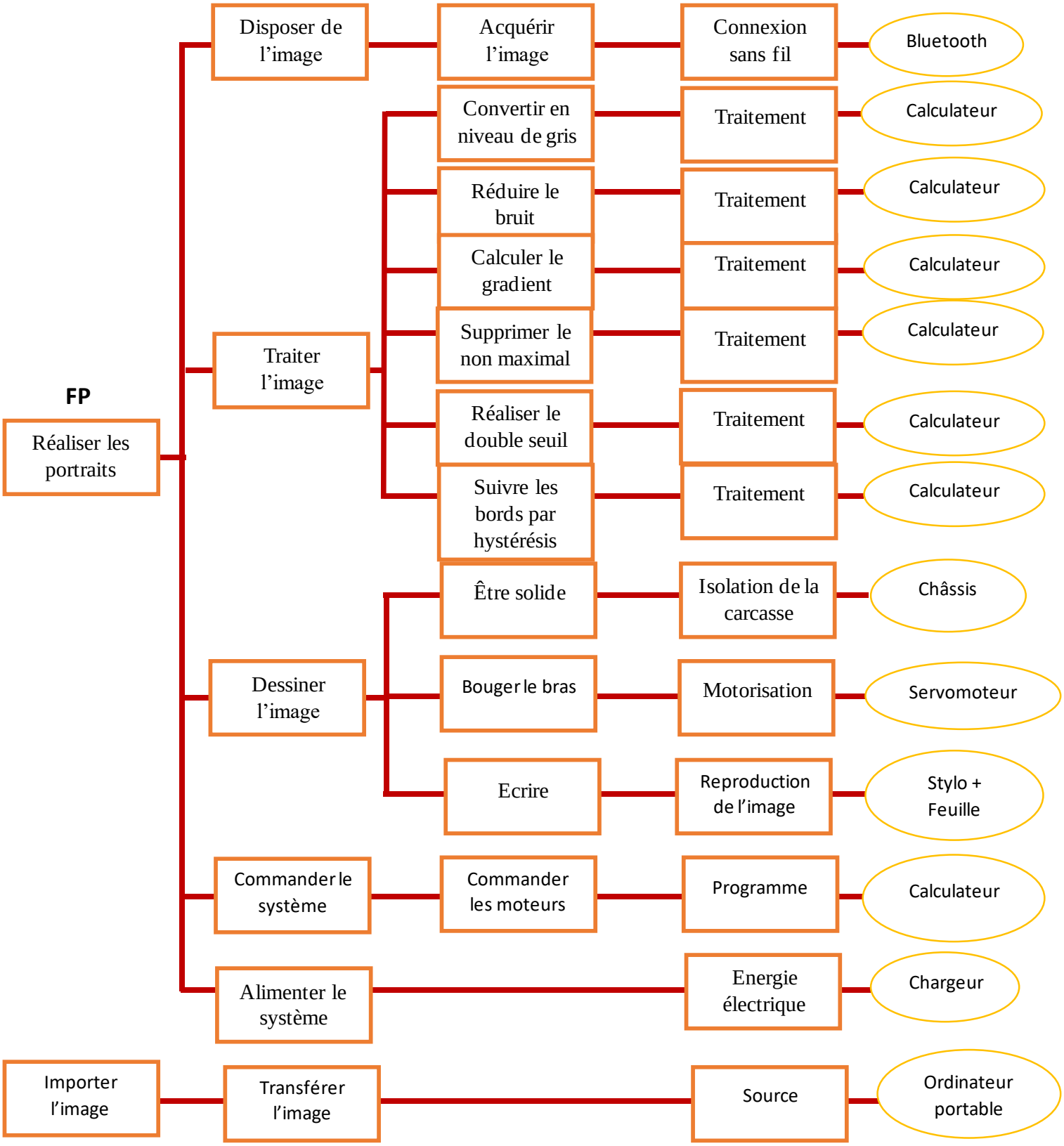


Figure 13 – Diagramme de FAST

II.7 Conclusion

En définitive, l'analyse fonctionnelle a permis d'étudier le besoin de notre projet ainsi que sa structure technique et fonctionnelle à travers l'exploitation des différents diagramme et tableaux. Le chapitre prochain sera consacré essentiellement à l'étude et le dimensionnement du robot.

Chapitre 3 : Étude

III.1 Introduction

Nous aborderons dans ce chapitre, l'étude du projet. Nous commencerons d'abord par choisir la technologie adéquate à notre travail, avec une présentation du bilan financier qu'il nous est permis d'établir. Nous poursuivrons ensuite avec une modélisation géométrique afin d'obtenir les différentes positions atteignables par le bras. Nous finirons par présenter la procédure qu'on utilisera pour traiter l'image.

III.2 Choix de la technologie

- Choix de la matière

Nous avons décidé de choisir le plastique plus spécifiquement le PLA dans la mesure où nous aurons à imprimer nos pièces à cause de la complexité de leurs formes ; le PLA étant le choix optimal qui nous était offert.

Figure 24 – Filament PLA

❖ Choix des servomoteurs

En ce qui concerne les servomoteurs, nous avons plusieurs options à notre disposition, cependant selon notre dimensionnement et la matière que nous avons choisie, nous avons opté pour le S3003 pour tenir le stylo et le MG995 pour les deux premières articulations ; d'abord parce qu'ils ont des couples nécessaires aux mouvements que nous aurons besoin de reproduire, ensuite parce que son coût pourrait rentrer dans notre budget et qu'ils consomment peu d'énergie.

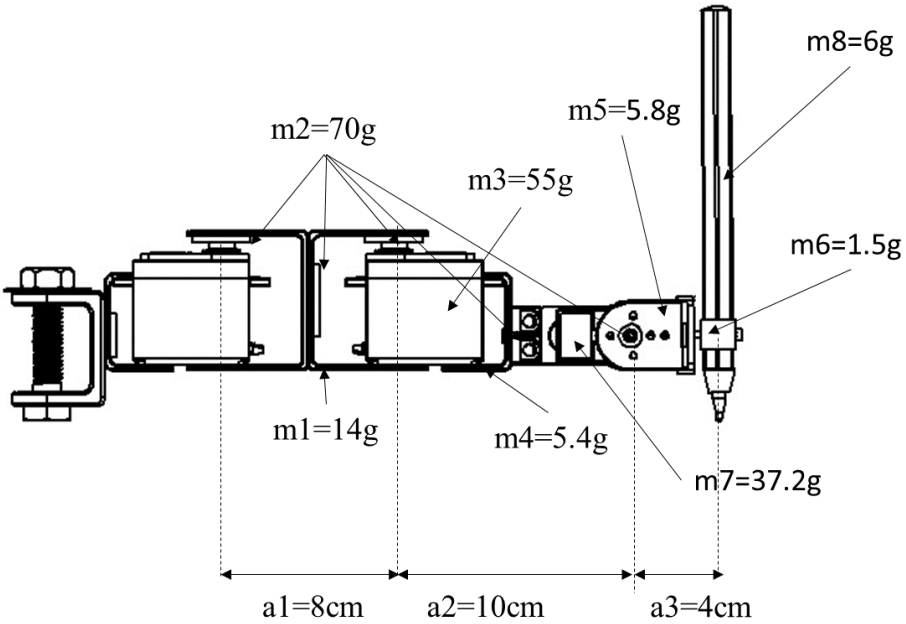


Figure 35 – Dimensionnement du bras

On a

$$\begin{cases} m1 = \text{masse de la première pièce de laison} \\ m2 = \text{masse des cinq disques métalliques} \\ m3 = \text{masse du servomoteur MG995} \\ m4 = \text{masse de la seconde pièce de laison} \\ m5 = \text{masse de la troisième pièce de laison} \\ m6 = \text{masse du support du stylo} \\ m7 = \text{masse du servomoteur S3003} \\ m8 = \text{masse du stylo} \\ Pi = \text{poids supporté par l'articulation i allant de la base} \\ Ci = \text{couple du servomoteur i allant de la base} \\ di = \text{distance entre l'axe de l'articulation i et l'effecteur allant de la base} \\ ai = \text{distance entre l'axe rotation i et i} + 1 \end{cases}$$

Pour la première articulation en choisissant le MG995 on aurait besoin d'un couple :

C1 = d1*P1

Avec P1 = (m1 + m2 + m3 + m4 + m5 + m6 + m7 + m8) μ, μ=9.8 N/Kg

P1 = 194.9*9.8 = 1.91 N

Et d1 = a1 + a2 + a3 = 22 cm

Donc C1 = 1.91*22 = 42.02 N.cm = 4.28 Kg.cm

Quant à la deuxième on aurait :

C2 = d2*P2

Avec P2 = $(\frac{m2}{5} * 3 + m3 + m4 + m5 + m6 + m7 + m8)$ μ

P2 = 152.9*9.8 = 1.5 N

Et d2 = a2 + a3 = 14 cm

Donc C2 = 1.5*14 = 21 N.cm = 2.14 Kg.cm

En ce qui concerne la dernière, on aurait :

C3 = d3*P3

Avec P3 = (m5 + m6 + m8) μ

P3 = 13.3*9.8 = 0.027 N

Et d3 = a3 = 4 cm

Donc C3 = 0.027*4 = 0.106 N.cm = 0.011 Kg.cm

Figure 46 – Servomoteur S3003

Figure 57 – Servomoteur MG995

Dans le tableau ci-dessous, nous avons récencé les différentes caractéristiques des servomoteurs qui nous ont été utiles dans nos choix.

Caractéristiques	S3003	MG995
Poids	37.2g	55g
Dimensions	40.4 x 19.8 x 36mm	40 x 20 x 36,5 mm
Couple (kg.cm)	3.2kg.cm sous 4.8V 4.1kg.cm sous 6V	11Kg.cm sous 4.8V - 13Kg.cm sous 6V
Vitesse de rotation (s/°)	0.23s/60° sous 4.8V 0.19s/60° sous 6V	0.16 s/60° sous 4.8V 0.13 s/60° sous 6V
Tension d'alimentation	4.8V - 6V	4.8V – 7.2V
Température de fonctionnement	0 ~ +55 °C	30 ~ +60 °C
Pignons	Nylon	Tout métal
Angle de rotation	180°	180°

Tableau 2 – Caractéristiques des servomoteurs

❖ Choix de l'unité de calcul

Selon les besoins de notre projet, on aurait besoin d'une carte de processeur de grande vitesse pouvant commander les servomoteurs, ayant un module Bluetooth pour recevoir l'image à traiter et une RAM suffisante pour assurer le traitement d'image. On a donc opté pour la carte Raspberry qui remplissait toutes ces conditions.

Raspberry Pi est un ordinateur entièrement fonctionnel, un système sur puce ; c'est-à-dire un système complet embarqué sur une seule puce ("circuit intégré"), qui fonctionne sur un système d'exploitation Linux spécialement conçu pour lui, nommé Raspbian. Comme un ordinateur, il a une mémoire, un processeur, des ports USB, une sortie audio, un pilote graphique pour la sortie HDMI et comme il fonctionne sous Linux, la plupart des applications logicielles Linux peuvent être installées dessus. Il a plusieurs modèles et révisions comme Raspberry Pi, Raspberry Pi 2, Raspberry Pi Model B + etc. Nous avons pris parti pour la version Raspberry Pi 4 2 GO.

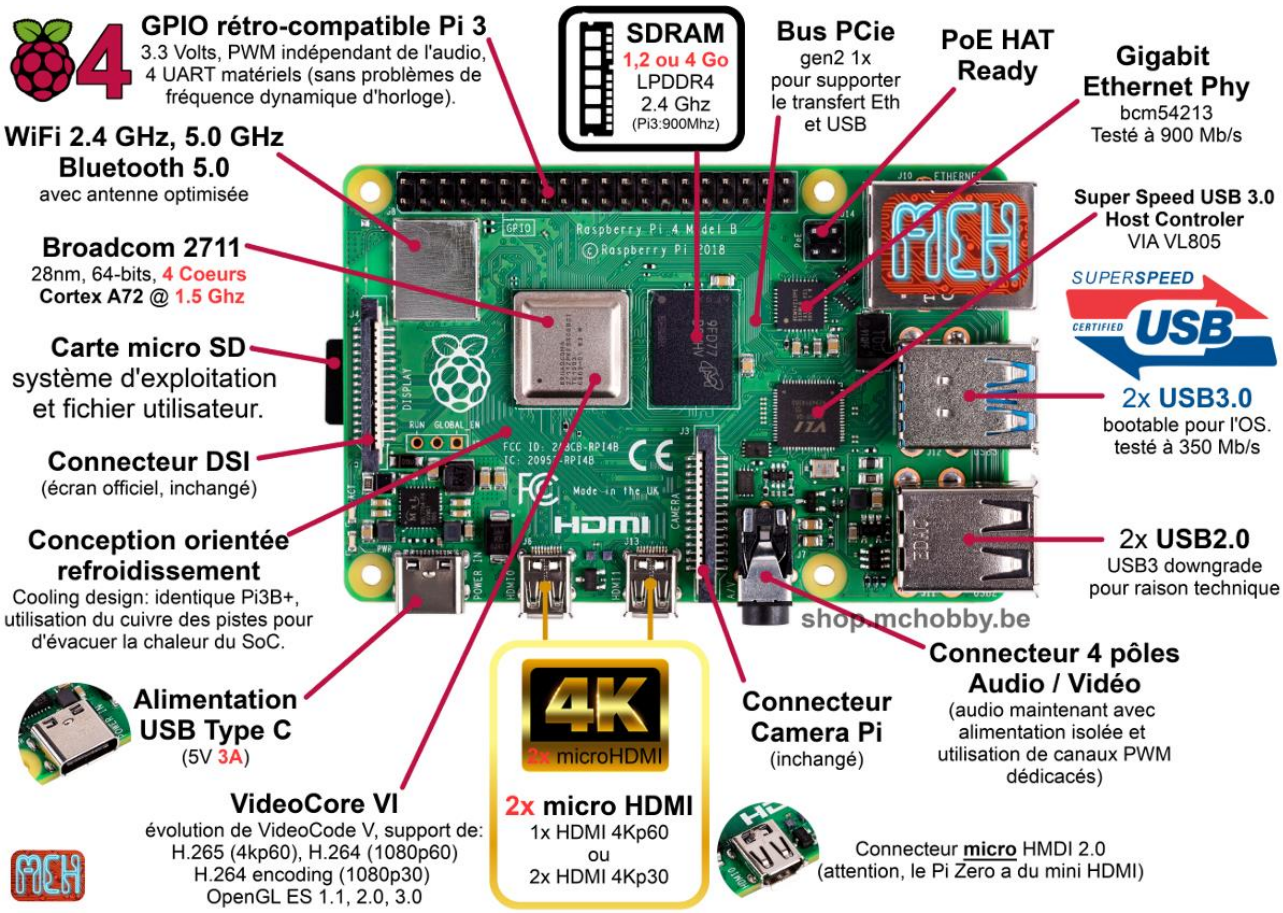


Figure 18 – Carte Raspberry Pi 4 et ses caractéristiques

❖ Choix de l'alimentation

Pour l'alimentation, nous avons choisi le chargeur type C Samsung car il délivre les 5V et 3A dont la carte aurait besoin pour fonctionner dans la mesure où c'était déjà à notre disposition, efficace et facile à utiliser

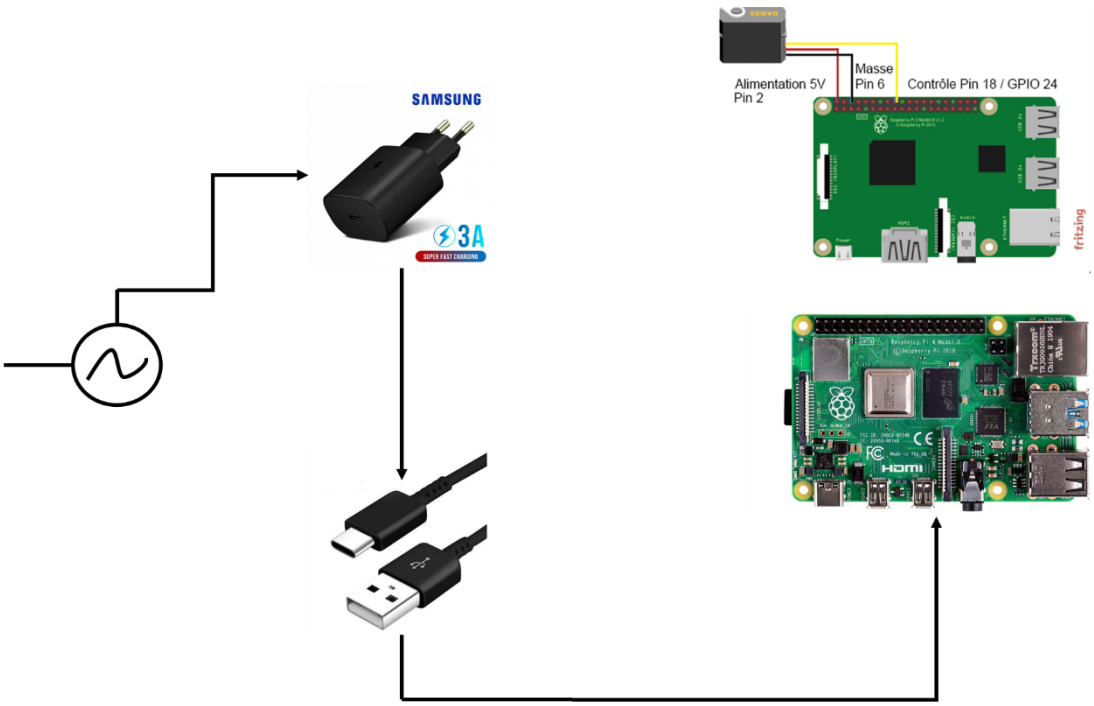


Figure 19 – Système d'alimentation

III.3 Etude du budget

N°	Matériel	Modèle	Quantité	Prix en dt	Total en dt
1	Carte de commande des servomoteurs, d'acquisition et traitement des images	Raspberry Pi 4 2GO	1	234	234
2	Protection carte	Boitier	1	80	80

3	Connexion	Adaptateur Micro HDMI male vers HDMI femelle	1	3	3
4	Stockage	Carte mémoire 32 GO	1	13.65	13.65
5	Affichage	Ecran	1	-	-
6	Alimentation	Chargeur Samsung type C	1	-	-
7	Motorisation	Servo MG996R	2	35	70
8	Motorisation	Servo S3003	1	-	-
9	Impression	!	!	!	!
10	Conception des boitiers et autres	!	!	!	!
-	Total	-	-	-	400,65

Tableau 3 – Etude du budget

III.4 Modélisation géométrique

Pour notre projet, nous avons choisi un robot à 3 ddls RRR ; c'est-à-dire un robot à trois articulations rotoïdes.

La convention, ou notation de Denavit-Hartenberg permet de caractériser la position relative de deux solides avec seulement quatre paramètres. Dans la suite, nous allons utiliser la convention modifiée DHM pour modéliser notre système.

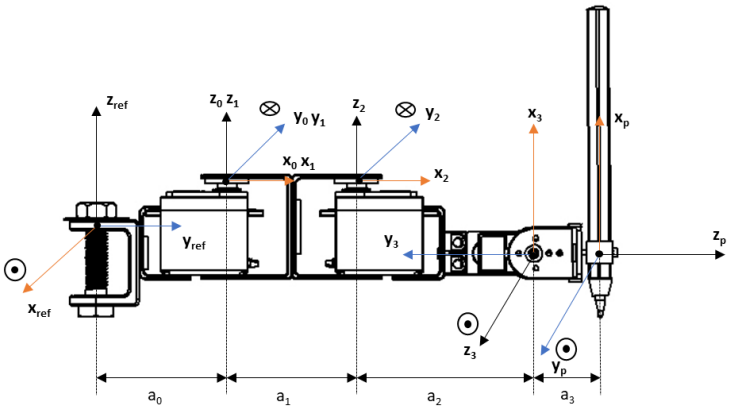


Figure 20 – Représentation du système avec les axes

❖ Modélisation géométrique directe

Dans cette partie essayerons de déterminer l'attitude du robot ; c'est-à-dire la position de son effecteur par rapport au repère de référence. Cela nous permettra de contrôler les mouvements des servomoteurs par la suite.

i = articulation

α_{i-1} = l'angle autour de l'axe X_{i-1} entre les axes Z_{i-1} et Z_i

a_{i-1} = la distance le long de l'axe X_{i-1} entre les axes Z_{i-1} et Z_i

θ_i = l'angle autour de l'axe Z_{i-1} entre les axes X_{i-1} et X_i

d_i = la distance le long de l'axe Z_{i-1} entre les axes X_{i-1} et X_i

${}_{i-1}^{i}H$ = matrice de transformation homogène

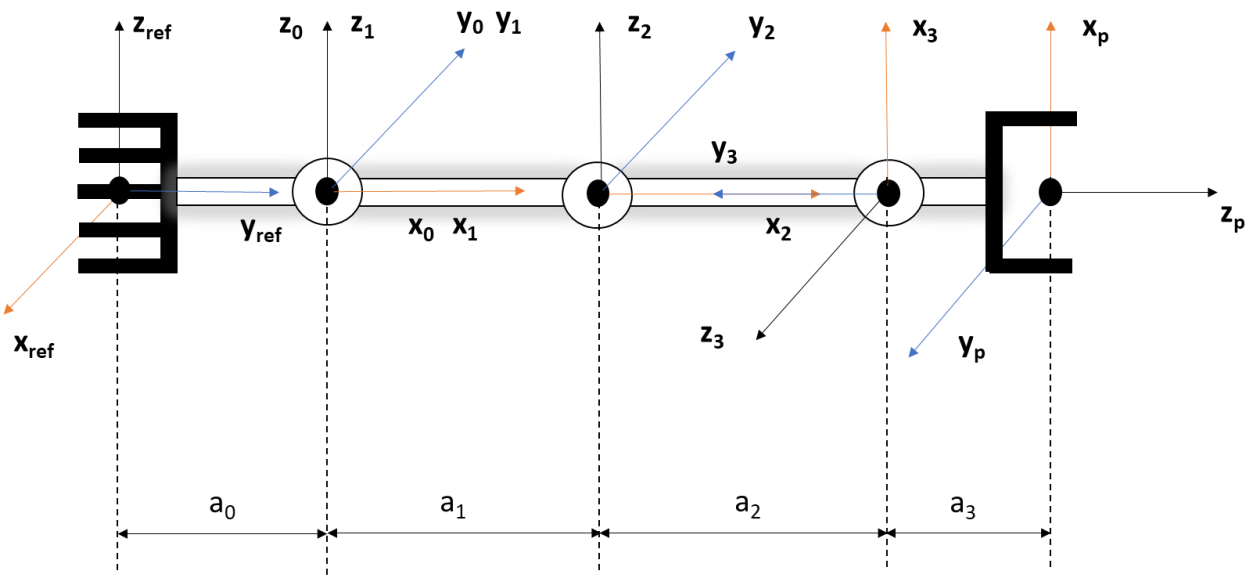


Figure 21 – Modélisation du système avec les axes

i-1 ⟶ i	i	α_{i-1}	a_{i-1}	Θ_i	d_i	${}_{i-1}^{i}H$
0 ⟹ 1	1	0	0	Θ_1	0	${}_{0}^{1}H$
1 ⟹ 2	2	0	a_1	Θ_2	0	${}_{1}^{2}H$
2 ⟹ 3	3	$\frac{\pi}{2}$	a_2	$\Theta_3 + \frac{\pi}{2}$	0	${}_{2}^{3}H$

Tableau 4 – Paramètres géométriques du robot

Apres calcul (que nous avons détaillé en annexe), nous retrouvons le système de coordonnées suivant :

$$\begin{cases} x = -a_1 sin\Theta_1 - a_2 sin(\Theta_1 + \Theta_2) - a_3 sin(\Theta_1 + \Theta_2) cos\Theta_3 \\ y = a_0 + a_1 cos\Theta_1 + a_2 cos(\Theta_1 + \Theta_2) + a_3 cos(\Theta_1 + \Theta_2) cos\Theta_3 \\ z = a_3 sin\Theta_3 \end{cases}$$

(X Y Z) = coordonnées cartésiennes de l'effecteur

- **Modélisation géométrique inverse**

Dans notre projet, les coordonnées cartésiennes sont censées être connues à l'avance grâce au programme du traitement d'image associant les pixels aux points matériels, nous allons donc essayer de retrouver les angles que les articulations devraient effectuer afin d'identifier comment les commander afin d'obtenir les positions qu'il faudrait avoir pour réaliser le dessin.

$$\begin{cases} x = -a_1 sin\Theta_1 - [a_2 + a_3 cos\Theta_3] sin(\Theta_1 + \Theta_2) \\ y = a_0 + a_1 cos\Theta_1 + [a_2 + a_3 cos\Theta_3] cos(\Theta_1 + \Theta_2) \\ z = a_3 sin\Theta_3 \end{cases}$$

Nous avons choisi de fixer l'angle Θ_3 à 30° quand le stylo est soulevé et 0° quand il est posé afin de faciliter la commande.

$$\Theta_2 = \arctan\left(\frac{\sqrt{1-A^2}}{A}\right)$$

$$A = \frac{x^2 + (y - a_0)^2 - a_1{}^2 - [a_2 + a_3 cos\Theta_3]^2}{2a_1[a_2 + a_3 cos\Theta_3]}$$

$$\boxed{\Theta_1 = -\Theta_2 + \lambda + \arctan\left(\frac{\sqrt{1-B^2}}{B}\right)}$$

$$B = \frac{a_1 sin\Theta_2}{\sqrt{x^2 + (y - a_0)^2}}$$

$$\lambda = \begin{cases} \arctan\left(\frac{y - a_0}{x}\right), & x \geq 0 \\ \pi + \arctan\left(\frac{y - a_0}{x}\right), & x < 0 \end{cases}$$

- **Singularité**

La singularité concerne l'ensemble des points non atteignables par le bras ; ce qui délimite le domaine de définition du robot.

$$\begin{cases} x = -a_1 sin\Theta_1 - [a_2 + a_3 cos\Theta_3] sin(\Theta_1 + \Theta_2) \\ y = a_0 + a_1 cos\Theta_1 + [a_2 + a_3 cos\Theta_3] cos(\Theta_1 + \Theta_2) \\ z = a_3 sin\Theta_3 \end{cases}$$

$$J = \begin{pmatrix} -a_1 cos\Theta_1 - [a_2 + a_3 cos\Theta_3] cos(\Theta_1 + \Theta_2) & -[a_2 + a_3 cos\Theta_3] cos(\Theta_1 + \Theta_2) & a_3 sin\Theta sin(\Theta_1 + \Theta_2) \\ -a_1 sin\Theta_1 - [a_2 + a_3 cos\Theta_3] sin(\Theta_1 + \Theta_2) & -[a_2 + a_3 cos\Theta_3] sin(\Theta_1 + \Theta_2) & -a_3 sin\Theta_3 \cos(\Theta_1 + \Theta_2) \\ 0 & 0 & a_3 \cos\Theta_3 \end{pmatrix}$$

$$|J| = a_3 a_1 [a_2 + a_3 cos\Theta_3] \cos\Theta_3 sin\Theta_2 = 0$$

$$\begin{cases} a_1 \neq 0 \\ a_2 \neq 0 \\ a_3 \neq 0 \end{cases} \Longrightarrow \begin{cases} [a_2 + a_3 cos\Theta_3] \neq 0 \\ \cos\Theta_3 = 0 \\ sin\Theta_2 = 0 \end{cases} \Longrightarrow \begin{cases} \Theta_3 = \frac{\pi}{2} + k\pi \\ \Theta_2 = k\pi \end{cases}$$

Or pour la première équation, il est déjà évident que $\Theta 3 = \frac{\pi}{2} + k\pi$ donc les seuls points non atteignables dans le domaine de définition c'est ceux où $\Theta 2 = k\pi$. Etant donné les angles de rotation des servomoteurs que nous avons choisis restreints à **90°** à droite et **90°** à gauche, l'étude ne s'arrêtera qu'à ce domaines

Dans les figures ci-dessous, nous avons essayé de délimiter l'espace opérationnel du robot. Nous avons donc considéré le robot en position initiale puis en mouvement selon ses dimensions. Pour cela, nous avons utiliser le logiciel Geogerba qui est un logiciel de mathématiques dynamiques, pour tous les niveaux d'éducation, intégrant en un seul outil d'utilisation aisée : géométrie, algèbre, feuilles de calcul, représentations graphiques, statistiques et calculs différentiels.

E centre du repère de la première articulation

F centre du repère de la deuxième articulation

G centre du repère de la troisième articulation

H position de l'effecteur

Les domaine en bleu représente l'ensemble des positions atteignables par la rotation du premier servomoteur lié à la base.

Les domaine en rouge représente l'ensemble des positions atteignables par la rotation du second servomoteur lié à la base.

Le domaine en restangle représente la feuille de papier A4. Il est à noter que les graduations ont été faites en centimètres.

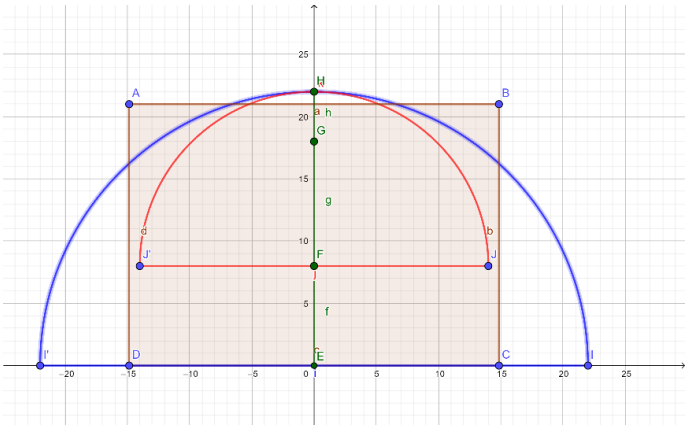


Figure 22 – Bras robotique en position initiale

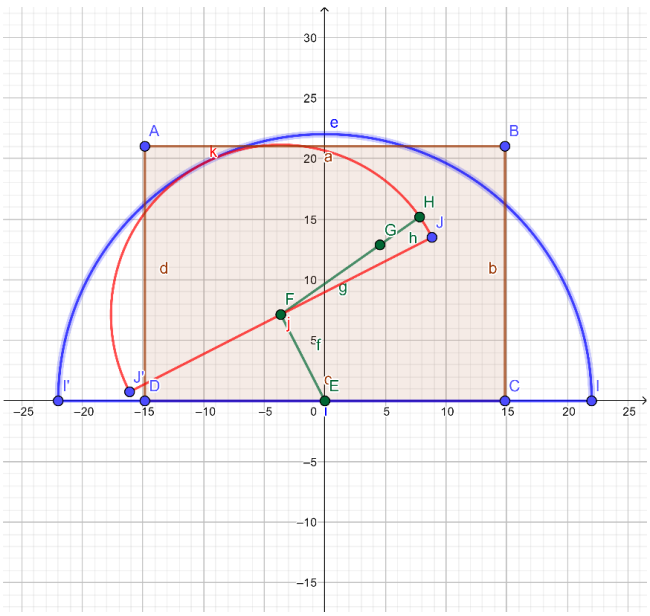


Figure 23 – Bras robotique en mouvement

III.5 Traitement d'image

Dans cette partie, nous exploiterons les différents procédés qu'on aura à utiliser pour passer d'une image en couleur RGB en traits réalisables par le robot. Pour se faire, nous avons décidé d'utiliser la méthode du filtre de Canny.

Le détecteur de bords Canny est un opérateur de détection de bords qui utilise un algorithme à plusieurs étapes pour détecter une large gamme de bords dans les images. Il a été développé par John F. Canny en 1986. Canny a également produit une théorie informatique de la détection des bords expliquant pourquoi la technique fonctionne.

L'algorithme de détection des contours Canny se compose de 5 étapes :

- Réduction de bruit
- Calcul de gradient
- Suppression non maximale
- Double seuil
- Suivi des bords par hystérésis

Après avoir appliqué ces étapes, vous pourrez obtenir le résultat suivant :

Figure 24 – Traitement d'image exemple 1

Une chose importante à mentionner, c'est que l'algorithme est basé sur des images en niveaux de gris. Par conséquent, la condition préalable est de convertir l'image en niveaux de gris avant de suivre les étapes mentionnées ci-dessus.

- Réduction de bruit

Étant donné que les mathématiques impliquées derrière la scène sont principalement basées sur des dérivés, les résultats de détection des contours sont très sensibles au bruit de l'image. Une façon de se débarrasser du bruit sur l'image est d'appliquer un flou gaussien pour la lisser.

Figure 25 – Image en gris (à gauche) - Image floue (à droite)

- Calcul de gradient

L'étape de calcul du gradient détecte l'intensité et la direction du bord en calculant le gradient de l'image à l'aide d'opérateurs de détection de bord.

Les bords correspondent à un changement d'intensité des pixels. Pour le détecter, le plus simple est d'appliquer des filtres qui mettent en évidence ce changement d'intensité dans les deux sens : horizontal (x) et vertical (y)

Figure 26 – Image floue (à gauche) – Gradient intensifié (à droite)

- Suppression non maximale

Idéalement, l'image finale devrait avoir des bords fins. Ainsi, nous devons effectuer une suppression non maximale pour affiner les bords.

Le principe est simple : l'algorithme passe par tous les points de la matrice d'intensité du gradient et trouve les pixels avec la valeur maximale dans les directions des bords.

Le résultat est la même image avec des bords plus fins. On peut cependant encore remarquer quelques variations concernant l'intensité des contours : certains pixels semblent plus lumineux que d'autres, et nous allons essayer de combler cette lacune avec les deux dernières étapes.

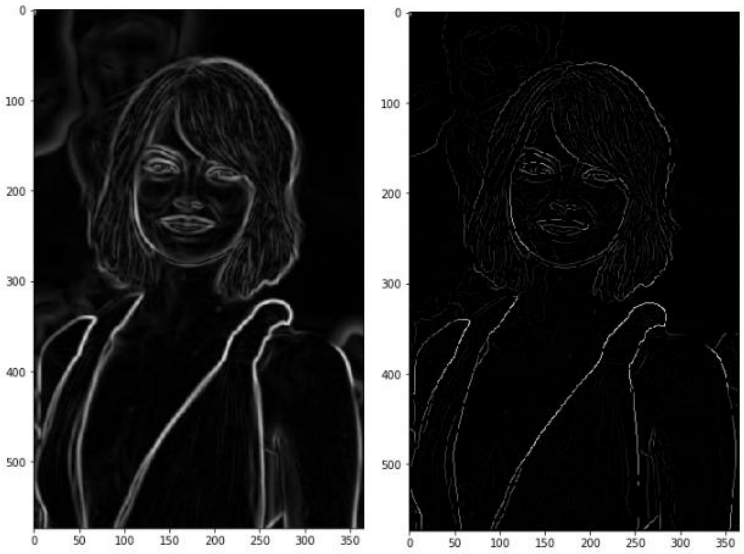


Figure 27 – Gradient intensifié (à gauche) – Suppression non maximal (à droite)

- Double seuil

L'étape du double seuil vise à identifier 3 types de pixels : forts, faibles et non pertinents : les pixels forts sont des pixels qui ont une intensité si élevée que nous sommes sûrs qu'ils contribuent au bord final. Les pixels faibles sont des pixels dont la valeur d'intensité n'est pas suffisante pour être considérée comme forte, mais pas suffisamment petite pour être considérée comme non pertinente pour la détection des contours. Les autres pixels sont considérés comme non pertinents pour le bord.

Vous pouvez maintenant voir ce que le double seuil contient : Le seuil haut est utilisé pour identifier les pixels forts (intensité supérieure au seuil haut). Le seuil bas est utilisé pour identifier les pixels non pertinents (intensité inférieure au seuil bas)

Tous les pixels ayant une intensité entre les deux seuils sont signalés comme faibles et le mécanisme d'hystérésis (étape suivante) nous aidera à identifier ceux qui pourraient être considérés comme forts et ceux qui sont considérés comme non pertinents.

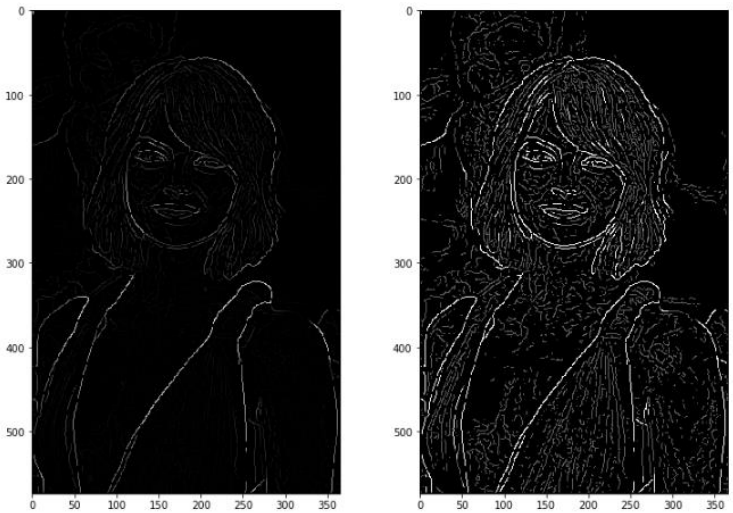


Figure 28 – Suppression non maximal (à gauche) – Résultat du seuil (à droite)

- Suivi des bords par hystérésis

Sur la base des résultats de seuil, l'hystérésis consiste à transformer des pixels faibles en pixels forts, si et seulement si au moins l'un des pixels autour de celui en cours de traitement est fort, comme décrit ci-dessous :

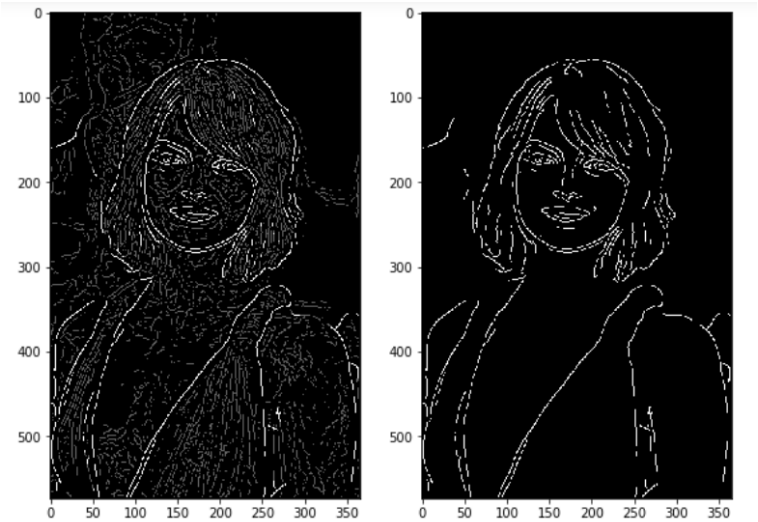


Figure 29 – Résultat du seuil (à gauche) – Résultats du processus d'hystérésis (à droite)

Voici donc quelques autres exemples de traitement que nous avons pu faire :

Figure 30 – Traitement d'image exemple 21

Figure 31 – Traitement d'image exemple 22

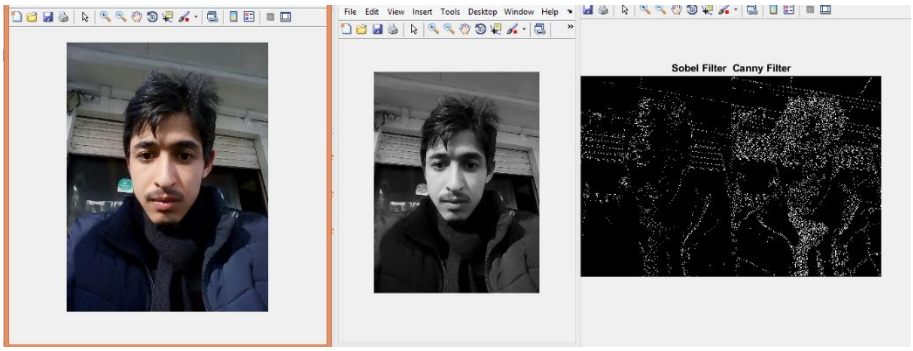


Figure 32 – Traitement d'image exemple 3

Il est à noter qu'après ce traitement, un processus d'extraction des lignes de ces contours suivi d'un filtrage pour ne garder que l'essentiel sera lancé afin de retrouver les traits assimilables à des segments de droites géométriques que le robot percevra sous forme de deux points à lier grâce à un programme traité et exécuté par son unité de calcul. Evidemment le résultat final dépend essentiellement de la qualité de l'image.

L'environnement de travail qu'on utilisera pour notre traitement c'est l'Opencv sur Raspberry Pi 4. L'Opencv est un logiciel informatique (Vision par ordinateur open source). C'est une bibliothèque open source qui comprend plusieurs centaines d'algorithmes de vision par ordinateur, il contient à une structure modulaire, ce qui signifie que le package comprend plusieurs bibliothèques partagées ou statiques. Parmi ces modules on a :

- Un module de traitement d'image qui inclut le filtrage d'images linéaire et non linéaire, les transformations d'images géométriques (redimensionnement, déformation affine et perspective, re mappage générique basé sur table), la conversion de l'espace colorimétrique, les histogrammes, etc.
- Détection d'objets et d'instances des classes prédéfinies (par exemple, visages, yeux, tasses, personnes, voitures, etc.)

III.6 Conclusion

A travers cette étude du projet, nous avons pu déterminer la technologie applicable à notre projet. Nous avons de plus cherché à avoir toutes les positions possibles et réalisables avant de présenter le procédé de traitement d'image qui corresponde le plus à notre travail et qui nous permettra de lier la partie purement mécanique à la partie traitement d'image.

Chapitre 3 : Conception et Simulation

IV.1 Introduction

Dans cette dernière partie de notre projet, à défaut de pouvoir le réaliser à cause de diverses raisons, nous nous arrêterons à l'étude de la conception et de la simulation du bras dessinateur.

IV.2 Conception

L'outil utilisé pour la conception de projet est SolidWorks. Ce logiciel CAO est une application de conception mécanique 3D paramétrique qui permet aux concepteurs d'esquisser rapidement des idées, d'expérimenter des fonctions et des cotes afin de produire des modèles et des mises en plan précises.

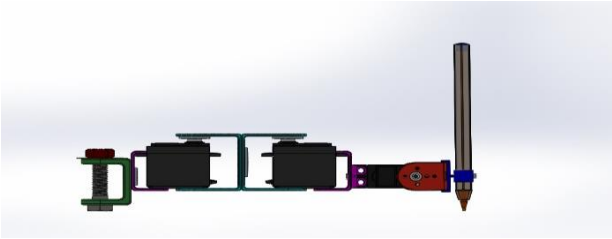


Figure 33 – Vue de côté du Robot

Figure 34 – Vue de dessus du Robot

Figure 35 – Vue de dessous du Robot

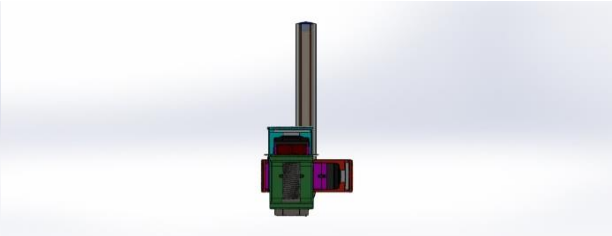


Figure 36 – Vue de derrière du Robot

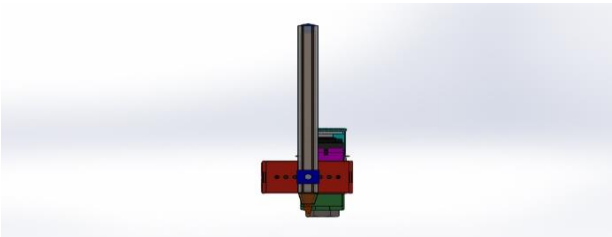


Figure 37 – Vue de face du Robot

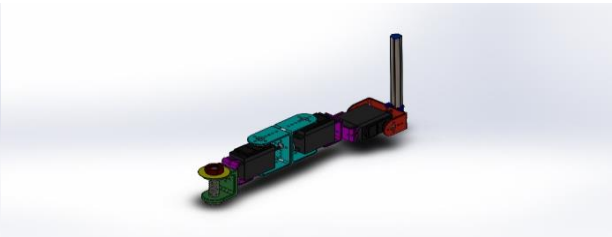
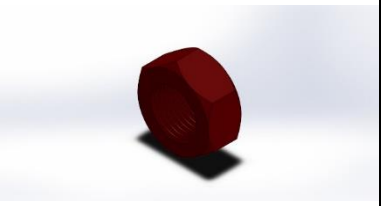
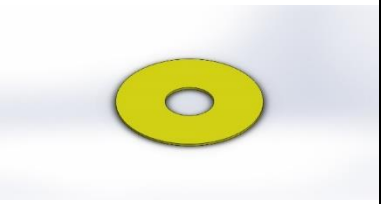
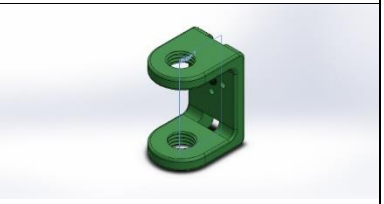
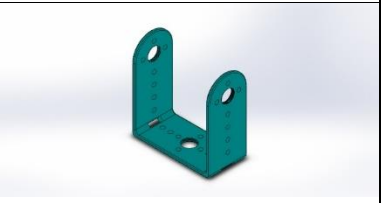
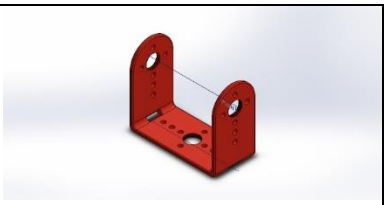
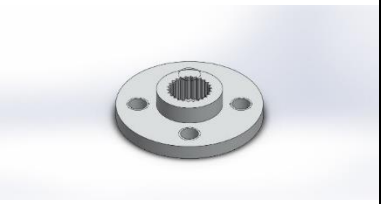
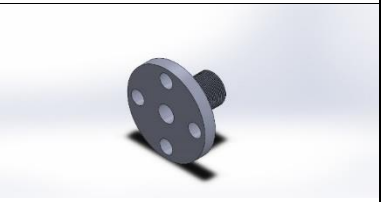
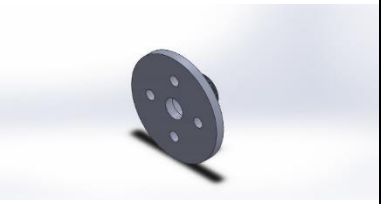
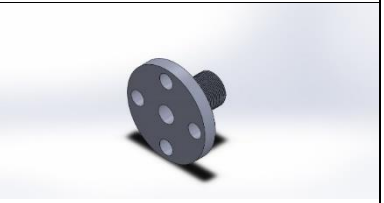
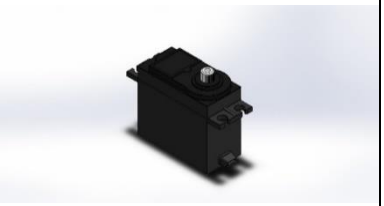


Figure 38 – Vue isométrique du Robot

Tableau ci-dessous décrit brièvement le rôle de chaque pièce dans notre système tout en donnant leurs masses approximatives en fonction de la matière choisie dans le chapitre précédent.

Numéro	Nom	Masse	Description	Image
1	Vis de blocage	33.07g	Le système Vis-Ecrou-Rondelle sert à fixer le robot à la table et à le maintenir en équilibre au repos et lorsqu'il est en mouvement. Dans la réalisation, contrairement aux autres pièces, il sera acheté et adapté au projet.	
2	Écrou	7.86g		
3	Rondelle	2.47g		
4	Support de la vis de blocage	11.77g	Les supports permettent de lier les différents servomoteurs grâce aux vis et écrous additionnels. Ils sont conçus de telle sorte à correspondre aux dimensions des servomoteurs.	
5	Support 1 du Servomoteur	2.55g		
6	Support 2 du servomoteur	8.12g		

7	Support 3 du servomoteur	6.98g		
8	Disque du servomoteur	5.61g	Lié au pignon du servomoteur, il transmet le mouvement de rotation au support qui entrainant ainsi les pièces suivantes	
9	Disque 1 de support	0.39g	Ces disques permettent la liaison entre deux supports successifs selon leurs forment	
10	Disque 2 de support	1.48g		
11	Disque 3 de support	0.52g		
12	Servomoteur	55g (MG995) 37.2g (S3003)	Les servomoteurs créent le mouvement de rotation c'est bien pour cela que le choix de leur dimensionnement est primordial	

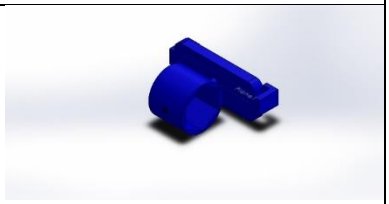
			dans la conception	
13	Support stylo	1.33g	Ce support maintient le stylo stable grâce à la vis de serrage du stylo. Il est adaptable aux différentes formes classiques de stylo. Ce couple Support-Vis représente l'effecteur du robot	
14	Vis de support Stylo	0.31g		
15	Support stylo	5.09g	Stylo permet de réaliser le dessin.	

Tableau 5 – Description des pièces

IV.3 Simulation

En ce qui concerne la simulation, nous nous sommes intéressés au logiciel Matlab, plus précisément la plateforme de simulation multi-domaine et de modélisation de systèmes dynamiques Simulink. Elle fournit un environnement graphique et un ensemble de bibliothèques contenant des blocs de modélisation qui permettent le design précis, la simulation, l'implémentation et le contrôle de systèmes de communications et de traitement du signal.

Grace au langage intégré dans Simulink Simscape, nous avons pu créer un modèle au moyen de connexions physiques qui s'intègrent directement dans des schéma-blocs. En effet, en assemblant des composants de l'assemblage importés de SolidWorks sous format STL dans des schéma-blocs avec leurs caractéristiques, nous avons pu modéliser notre robot en un système plus simple à commander.

A travers l'études faite dans les chapitres précédents et les schéma-blocs de Simscape, nous avons pu amener notre bras à réaliser un extrait des commandes qui lui seront attribuées ; c'est-à-dire aller d'un point P_1 à un autre P_2 dessinant ainsi la trajectoire d'un segment de droite en contrôlant les mouvements des servomoteurs.

Les figures suivantes montrent comment le programme en langage Simulink se présente :

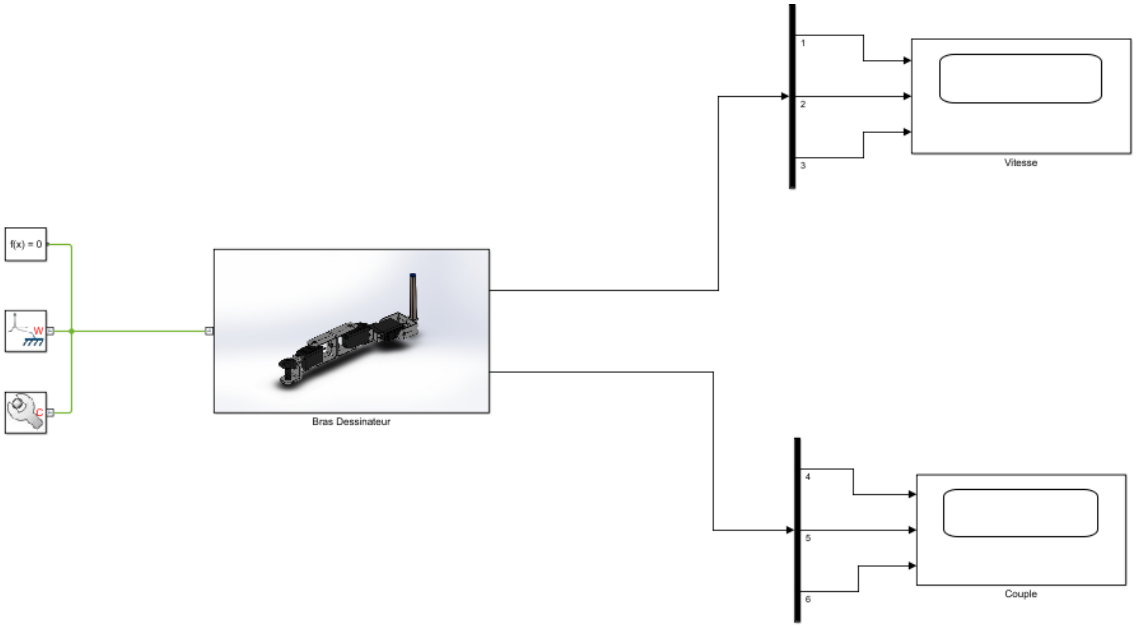


Figure 39 – interface de programmation en langage Simscape 1

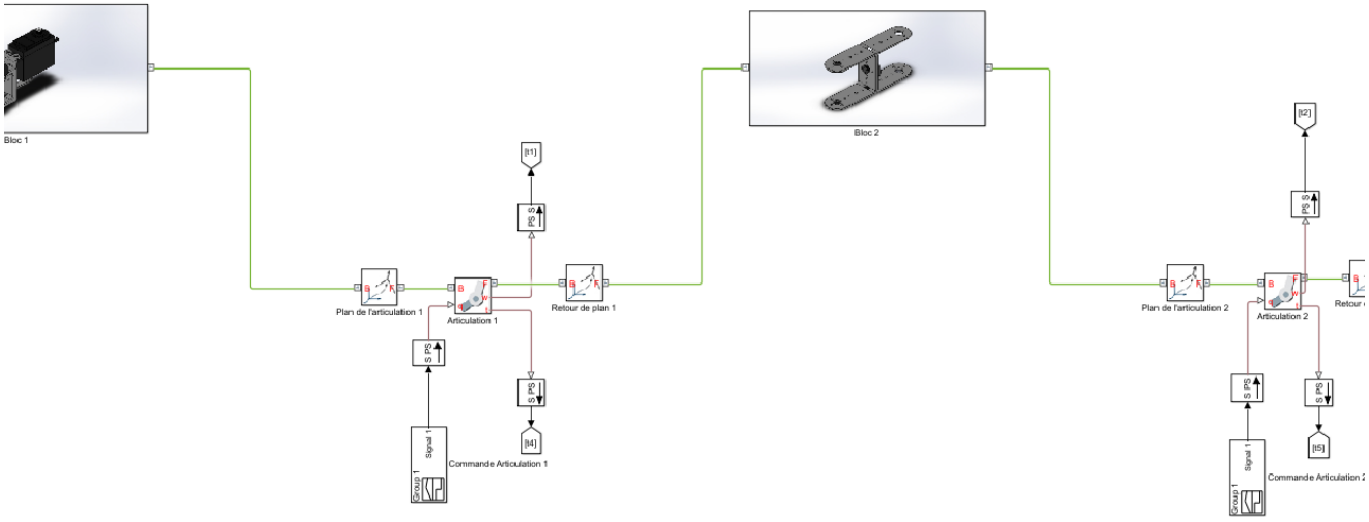


Figure 40 – interface de programmation en langage Simscape 2

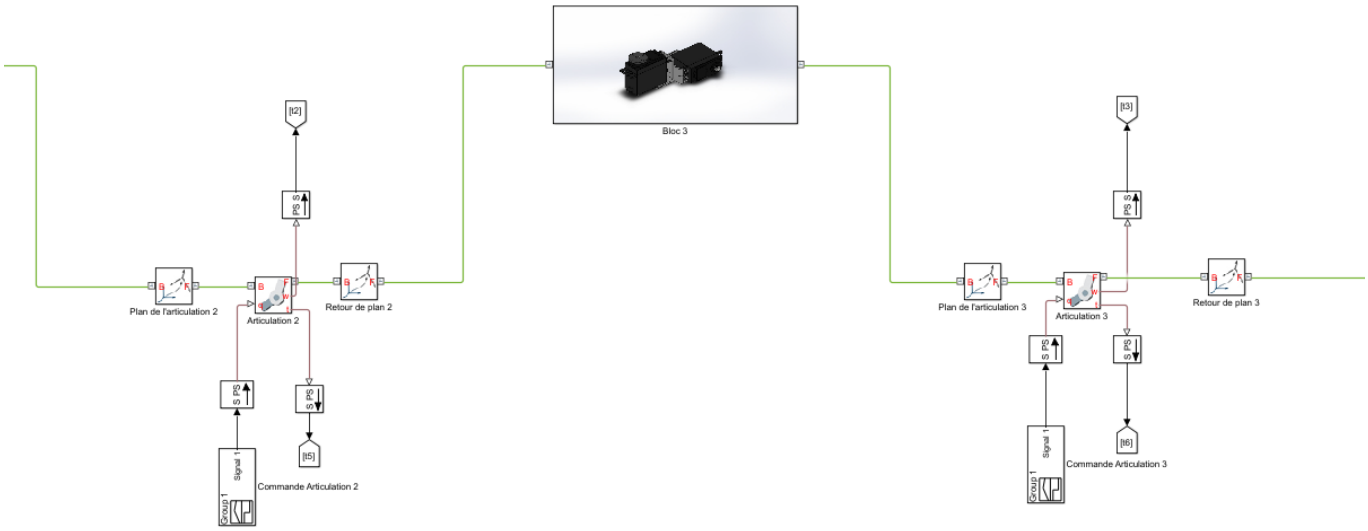


Figure 41 – interface de programmation en langage Simscape 3

IV.4 Conclusion

Nous avons consacré cette partie essentiellement à l'étude de conception et de simulation de notre robot. En se servant des logiciels SolidWorks et Matlab, nous avons pu avoir un aperçu de comment fonctionnerait notre robot en réalité et ainsi identifier les failles.

Conclusion générale

Le domaine de la robotique prend de plus en plus de l'ampleur aujourd'hui ; les idées commencent à diverger. Outre le facteur industriel, la robotique se tournent vers d'autres divers horizons dont l'art et le divertissement. Allant d'un robot pianiste, au robot chef cuisinier, les robots artistes dessinateur commencent quant à eux, à gagner le marché.

Etant donné la grande portée qu'elle nous offre de par sa complexité ; à savoir la manipulation des configurations robotiques, l'exploitation du système embarqué et des algorithmes de traitement d'image. Nous avons donc tracé un objectif, dans ce travail, qui est l'étude et la conception d'un bras dessinateur afin de relever ce défi et tirer parti de ces différents domaines.

Pour ce faire, nous avons établi un plan d'étude et de conception aussi simple et compréhensible possible afin de cerner l'essentiel pour atteindre notre objectif qui est de faire dessiner un bras robotique à trois degrés de liberté des portraits d'une image reçue dans son calculer sur une feuille de papier.

Dans un premier temps, nous avons d'abord essayer de fixer notre objectif et connaitre un plus sur le sujet en consultant d'autres projets semblables. Nous avons ensuite établi une analyse fonctionnelle pour avoir une idée claire du système qu'on voudrait réaliser ainsi que son fonctionnement. Puis nous avons fait l'étude de dimensionnement afin de déterminer la technologie applicable à notre projet et nous avons fini avec la conception et la simulation.

Il est à noter que nous avons rencontré plusieurs difficultés lors de l'étude de dimensionnement ; cependant le plus dur a été la partie traitement d'image et l'implémentation du programme le liant aux mouvements des servomoteurs à cause du peu de connaissance que nous avions dans ces domaines.

Toutefois ce projet pourrait être amélioré de diverses manières que nous n'avons pas inclus dans notre travail faute de moyens et de temps et il serait intéressant d'analyser les différentes possibilités d'ouverture pour le développer davantage.

Bibliographie

https://fr.wikipedia.org/

https://ch.mathworks.com/help/simulink

https://www.raspberrypi.org/blog/

Rapport Projet : ***« Robot Artiste Dessin de portrait sur papier dessin »***

De Mohamed Amine Mobarek et Rami Ferjani

https://ch.mathworks.com/help/images/edge-detection.html

Annexe

Annexe 1 : Installation de OpenCV 2.4.10 sur un Raspberry Pi B

1) mettre à jour

   Assurez-vous que Raspbian est à jour:

```
Sudo apt-get update

Sudo apt-get upgrade
```

2) installer les dépendances

```
Sudo apt-get –y
```

   Puis faire:

```
sudo apt-get –y install
```

3) Installez OpenCV

   Télécharger OpenCV de http://opencv.org/downloads.html

```
wget http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.10/opencv-2.4.10.zip/download opencv-2.4.10.zip
```

   Ou obtenir le dernier instantané de la git:

```
git clone https://github.com/Itseez/opencv.git
```

Enter dans le dossier opencv-2.4.10

```
cd opencv-2.4.10
```

Créer un nouveau dossier release

```
mkdir release
```

Enter dans le dossier release

cd release

En fin

```
ccmake ../
```

Décompressez et se préparer pour la construction

```
unzip opencv-2.4.10.zip
```

Annexe 2 : Modélisation

– Calcul de l'attitude

Avec ${}_{i-1}^{i}\mathbf{H} = \begin{pmatrix} \boldsymbol{cos\theta} & \boldsymbol{-cos\alpha\ sin\theta} & \boldsymbol{sin\alpha\ sin\theta} & \boldsymbol{acos\theta} \\ \boldsymbol{sin\theta} & \boldsymbol{cos\alpha\ cos\theta} & \boldsymbol{-sin\alpha\ cos\theta} & \boldsymbol{asin\theta} \\ \mathbf{0} & \boldsymbol{sin\alpha} & \boldsymbol{cos\alpha} & \boldsymbol{d} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$

$${}_{ref}^{0}H = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & a_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}_{3}^{p}H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}_{0}^{1}H = \begin{pmatrix} cos\Theta_1 & -sin\Theta_1 & 0 & 0 \\ sin\Theta_1 & cos\Theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}_{1}^{2}H = \begin{pmatrix} cos\Theta_2 & -sin\Theta_2 & 0 & a_1 \\ sin\Theta_2 & cos\Theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}_{2}^{3}H = \begin{pmatrix} -sin\Theta_3 & -cos\Theta_3 & 0 & a_2 \\ 0 & 0 & -1 & 0 \\ cos\Theta_3 & -sin\Theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}_{ref}^{1}H = \begin{pmatrix} -sin\Theta_1 & -cos\Theta_1 & 0 & 0 \\ cos\Theta_1 & -sin\Theta_1 & 0 & a_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}_{ref}^{2}H = \begin{pmatrix} -sin(\Theta_1+\Theta_2) & -cos(\Theta_1+\Theta_2) & 0 & -a_1 sin\Theta_1 \\ cos(\Theta_1+\Theta_2) & -sin(\Theta_1+\Theta_2) & 0 & a_1 cos\Theta_1 + a_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}_{ref}^{3}H = \begin{pmatrix} sin(\Theta_1+\Theta_2)sin\Theta_3 & -sin(\Theta_1+\Theta_2)cos\Theta_3 & cos(\Theta_1+\Theta_2) & -a_1 sin\Theta_1 + a_2 sin(\Theta_1+\Theta_2) \\ -cos(\Theta_1+\Theta_2)sin\Theta_3 & -cos(\Theta_1+\Theta_2)cos\Theta_3 & sin(\Theta_1+\Theta_2) & a_0 + a_1 cos\Theta_1 + a_2 cos(\Theta_1+\Theta_2) \\ cos\Theta_3 & sin\Theta_3 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}_{ref}^{p}H = \begin{pmatrix} sin(\Theta_1+\Theta_2)sin\Theta_3 & cos(\Theta_1+\Theta_2) & -sin(\Theta_1+\Theta_2)cos\Theta_3 & -a_1 sin\Theta_1 - a_2 sin(\Theta_1+\Theta_2) - a_3 sin(\Theta_1+\Theta_2)cos\Theta_3 \\ -cos(\Theta_1+\Theta_2)sin\Theta_3 & sin(\Theta_1+\Theta_2) & cos(\Theta_1+\Theta_2)cos\Theta_3 & a_0 + a_1 cos\Theta_1 + a_2 cos(\Theta_1+\Theta_2) + a_3 cos(\Theta_1+\Theta_2)cos\Theta_3 \\ cos\Theta_3 & 0 & -sin\Theta_3 & a_3 sin\Theta_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{cases} x = -a_1 sin\Theta_1 - a_2 sin(\Theta_1+\Theta_2) - a_3 sin(\Theta_1+\Theta_2)cos\Theta_3 \\ y = a_0 + a_1 cos\Theta_1 + a_2 cos(\Theta_1+\Theta_2) + a_3 cos(\Theta_1+\Theta_2)cos\Theta_3 \\ z = a_3 sin\Theta_3 \end{cases}$$

– **Modélisation inverse**

$$\begin{cases} x = -a_1 sin\Theta_1 - [a_2 + a_3 cos\Theta_3] sin(\Theta_1 + \Theta_2) \\ y = a_0 + a_1 cos\Theta_1 + [a_2 + a_3 cos\Theta_3] cos(\Theta_1 + \Theta_2) \\ z = a_3 sin\Theta_3 \end{cases}$$

$$\frac{z}{a_3} = sin\Theta_3 \quad \Longrightarrow \quad \sqrt{1 - (\frac{z}{a_3})^2} = cos\Theta_3 \quad \Longrightarrow \quad \boxed{\Theta_3 = \arctan(\frac{\frac{z}{a_3}}{\sqrt{1-(\frac{z}{a_3})^2}})}$$

$$x^2 = {a_1}^2 (sin\Theta_1)^2 + [a_2 + a_3 cos\Theta_3]^2 [sin(\Theta_1 + \Theta_2)]^2 + 2a_1[a_2 + a_3 cos\Theta_3] sin\Theta_1 sin(\Theta_1 + \Theta_2)$$

$$(y - a_0)^2 = {a_1}^2 (cos\Theta_1)^2 + [a_2 + a_3 cos\Theta_3]^2 [\cos(\Theta_1 + \Theta_2)]^2 + 2a_1[a_2 + a_3 cos\Theta_3] \cos\Theta_1 \cos(\Theta_1 + \Theta_2)$$

$$x^2 + (y - a_0)^2 = {a_1}^2 + [a_2 + a_3 cos\Theta_3]^2 + 2a_1[a_2 + a_3 cos\Theta_3] \cos\Theta_2$$

$$x^2 + (y - a_0)^2 = {a_1}^2 + [a_2 + a_3 cos\Theta_3]^2 + 2a_1[a_2 + a_3 cos\Theta_3]$$

$$\cos\Theta_2 = \frac{x^2 + (y - a_0)^2 - {a_1}^2 - [a_2 + a_3 cos\Theta_3]^2}{2a_1[a_2 + a_3 cos\Theta_3]} = A \quad \Longrightarrow$$

$$\sin\Theta_2 = \sqrt{1 - A^2} \quad \Longrightarrow$$

$$\boxed{\Theta_2 = \arctan(\frac{\sqrt{1-A^2}}{A})}$$

$$\begin{cases} x\cos(\Theta_1+\Theta_2) = -a_1\sin\Theta_1\cos(\Theta_1+\Theta_2) - [a_2 + a_3\cos\Theta_3]\sin(\Theta_1+\Theta_2)\cos(\Theta_1+\Theta_2) \\ (y-a_0)\sin(\Theta_1+\Theta_2) = a_1\cos\Theta_1\sin(\Theta_1+\Theta_2) + [a_2 + a_3\cos\Theta_3]\sin(\Theta_1+\Theta_2)\cos(\Theta_1+\Theta_2) \\ z = a_3\sin\Theta_3 \end{cases}$$

$$x\cos(\Theta_1+\Theta_2) + (y-a_0)\sin(\Theta_1+\Theta_2) = a_1\sin\Theta_2 \quad \Longrightarrow$$

$$\sqrt{x^2+(y-a_0)^2}\cos(\Theta_1+\Theta_2-\lambda) = a_1\sin\Theta_2 \qquad \text{Avec} \qquad \lambda = \begin{cases} \arctan(\dfrac{y-a_0}{x}), & x \geq 0 \\ \pi + \arctan(\dfrac{y-a_0}{x}), & x < 0 \end{cases}$$

$$\cos(\Theta_1+\Theta_2-\lambda) = \frac{a_1\sin\Theta_2}{\sqrt{x^2+(y-a_0)^2}} = B \quad \Longrightarrow$$

$$\sin(\Theta_1+\Theta_2-\lambda) = \sqrt{1-B^2} \quad \Longrightarrow \quad \boxed{\Theta_1 = -\Theta_2 + \lambda + \arctan(\frac{\sqrt{1-B^2}}{B})}$$

Annexe 3 : Code

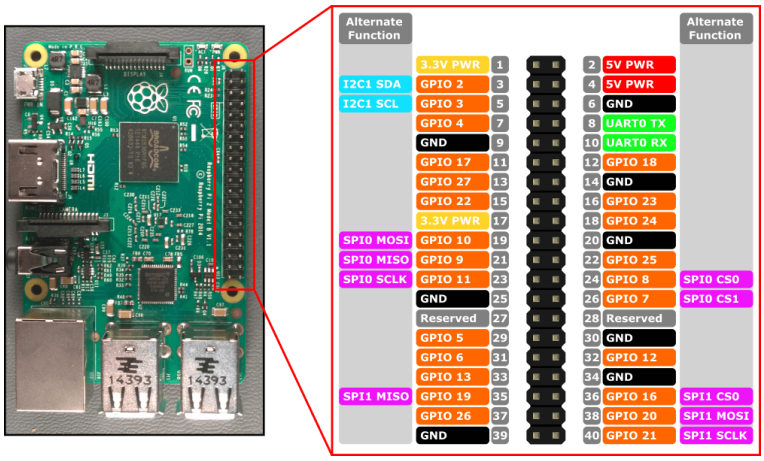
- Détection de contour

Pour se faire il faudra rentrer ce code sous l'invite de commande :

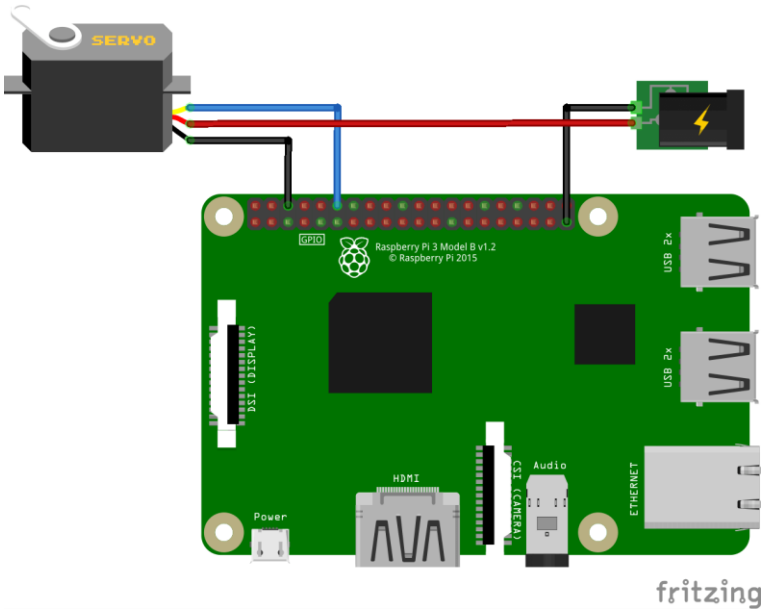
```
I = imread('image');
imshow(I)
x = rgb2gray(I);
figure
imshow(x)
BW1 = edge(x,'sobel');
BW2 = edge(x,'canny');
figure;
imshowpair(BW1,BW2,'montage')
title('Sobel Filter  Canny Filter');
```

- Exemple de pilotage d'un servo avec Rapberry

Description des broches



Schema de connexion

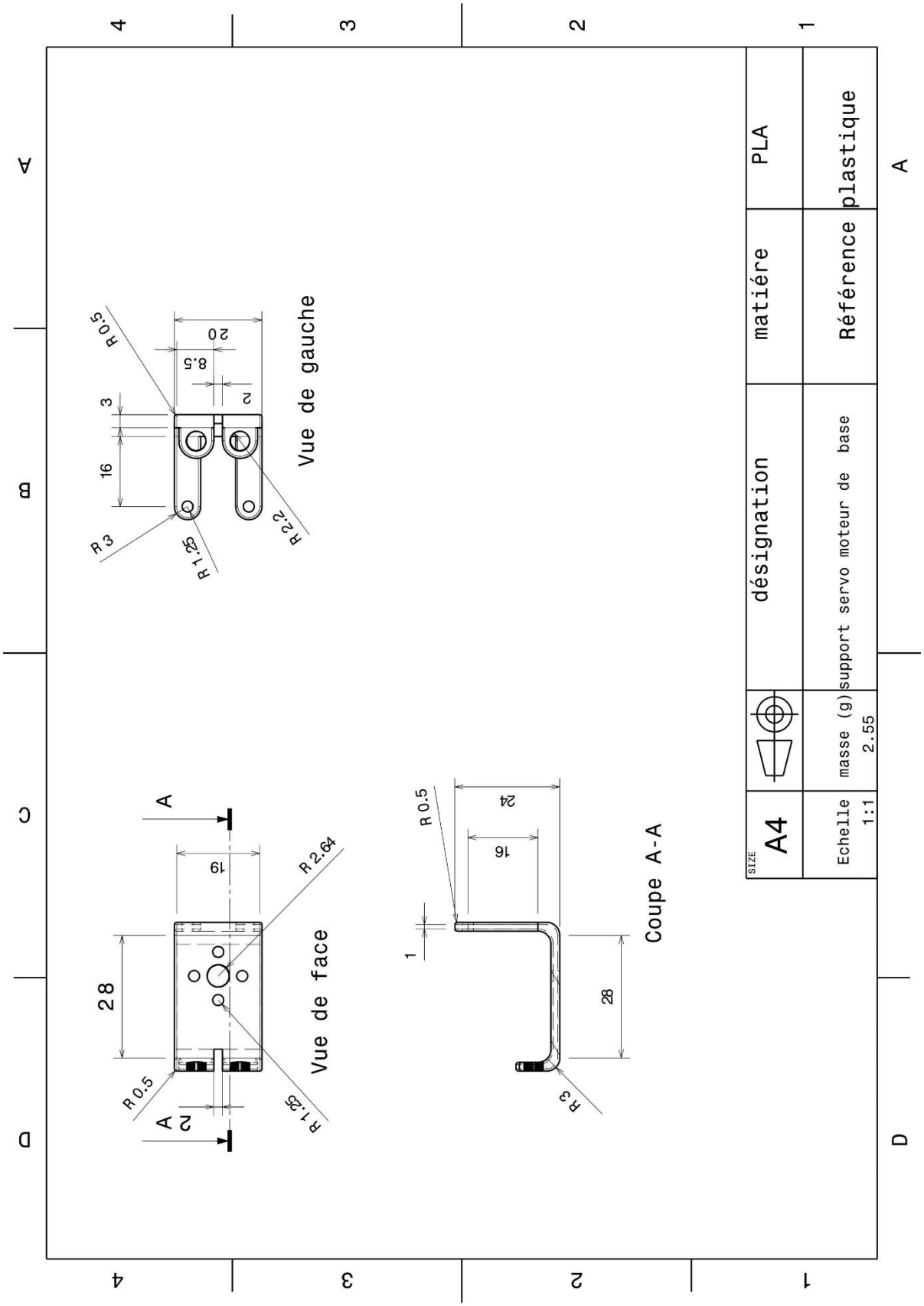


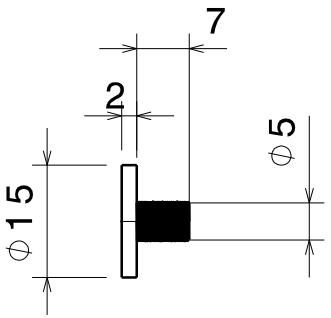
Code de base

```
#!/usr/bin/env  python
#-*-coding: latin-1-*-

import RPi.GPIO as GPIO
import time

def AngleToDuty(ang):
  return float(pos)/10.+5.

#Setup servoPin as PWM output of frequancy 100Hz
servoPin=12
GPIO.setmode(GPIO.BOARD)
GPIO.setup(servoPin,GPIO.OUT)
pwm=GPIO.PWM(servoPin,100)

#setup sweep parameters
depart =0
arrivee=180
DELAY=0.1
```

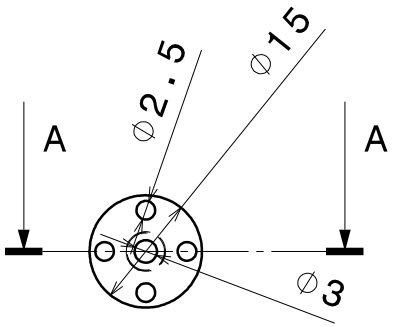
```
incStep=5
pos=depart

pwm.start(AngleToDuty(pos)) #star pwm

nbRun=3
i=0
while i<nbRun:
  print("--------------------------run  {}".format(i))
  for pos in range(depart,arrivee,incStep):
    duty=AngleToDuty(pos)
    pwm.ChangeDutyCycle(duty)
    time.sleep(DELAY)
    print("position: {}° -> duty cycle : {}%".format(pos,duty))

  for pos in range(arrivee,depart,-incStep):
    duty=AngleToDuty(pos)
    pwm.ChangeDutyCycle(duty)
    time.sleep(DELAY)
    print("position: {}° -> duty cycle : {}%".format(pos,duty))

  i=i+1

pwm.stop() #stop sending value to output
GPIO.cleanup() #release channel
```

Annexe 4 : Dessins techniques des pièces du bras

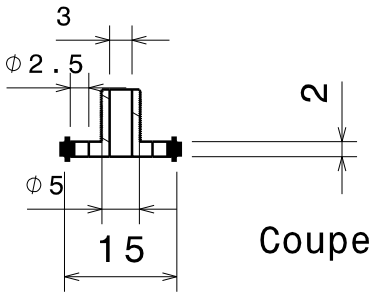




Vue de face

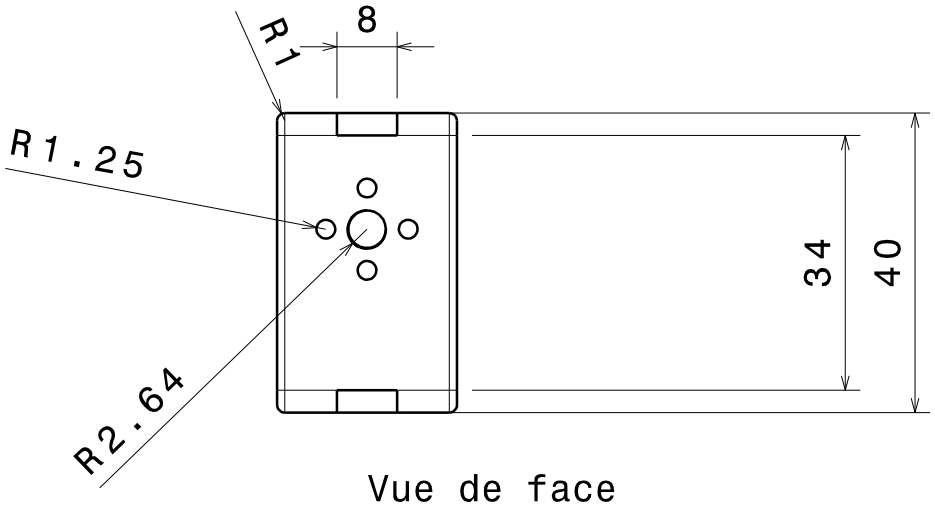


Vue de gauche

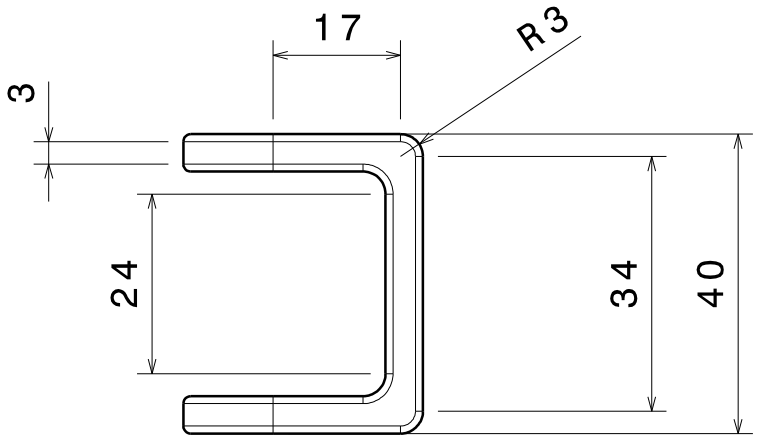


Coupe A-A

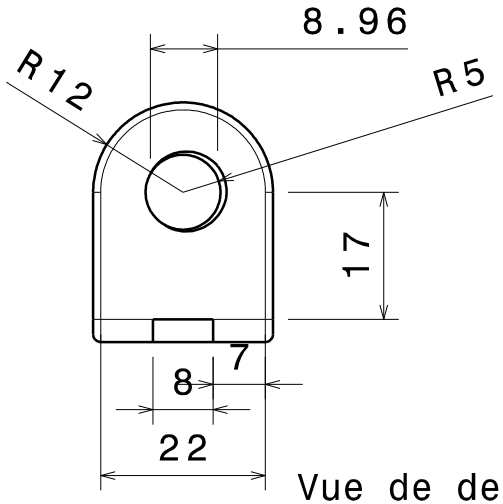
SIZE A4		désignation	matiére	PLA
Echelle	1:1	Disque support 3	Référence	plastique



Vue de face

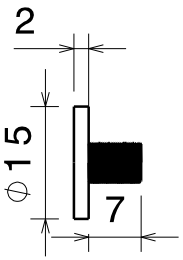


Vue de gauche

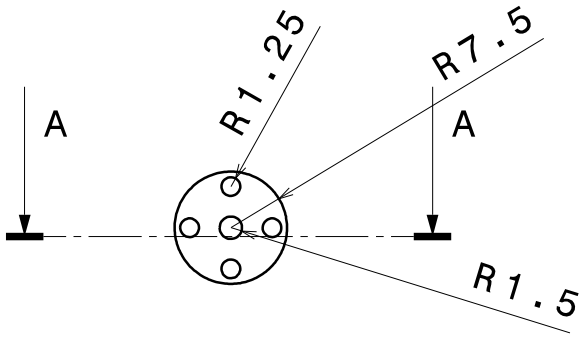


Vue de dessus

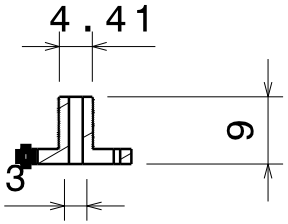
SIZE A4			désignation	matiére	PLA
Echelle	nb	1	suport de vis de blocage	Référence	plastique



Vue de face

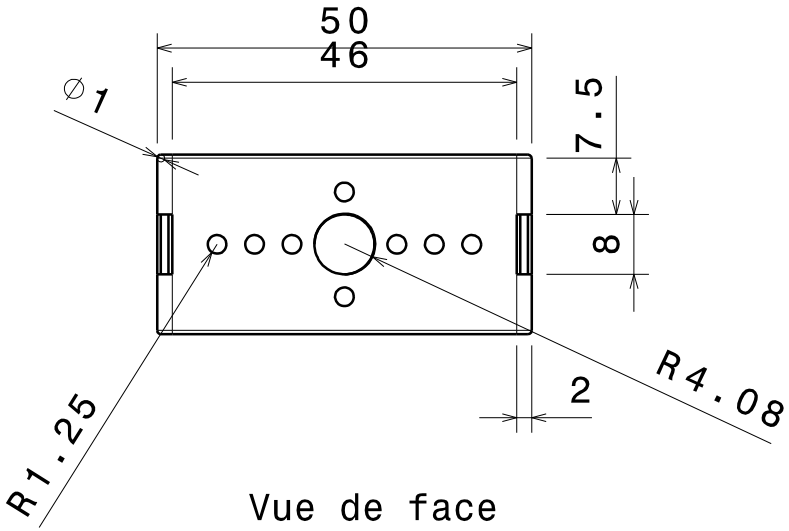


Vue de gauche

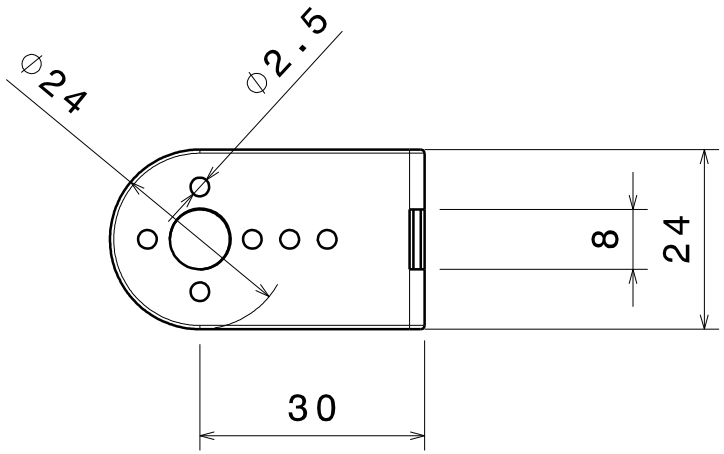


Coupe A-A

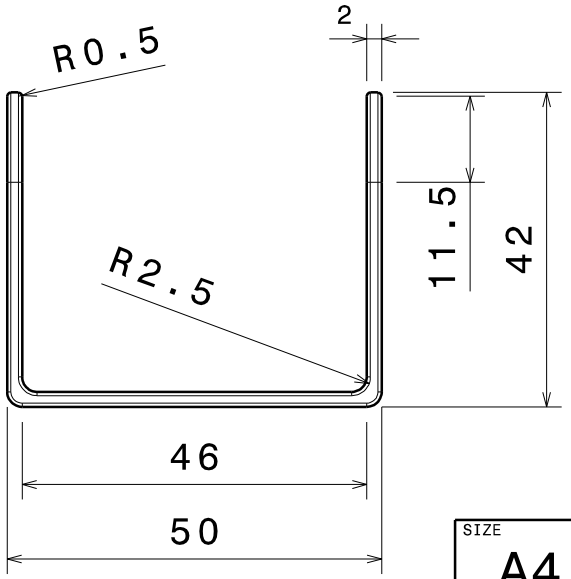
SIZE			désignation	matiére	alliage d'aluminium
A4					
1:1	nb	1	disuqe 1 du support	Référence	aluminum



Vue de face

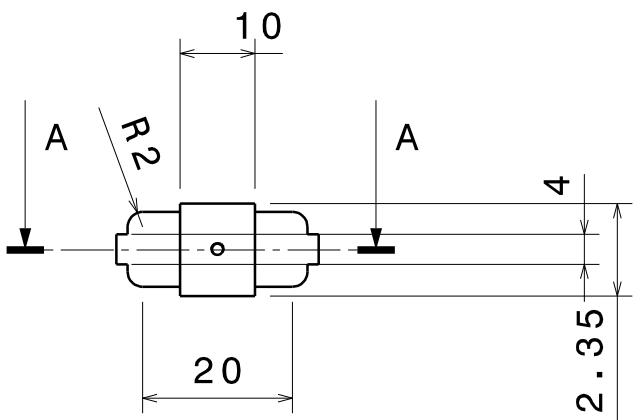


Vue de gauche

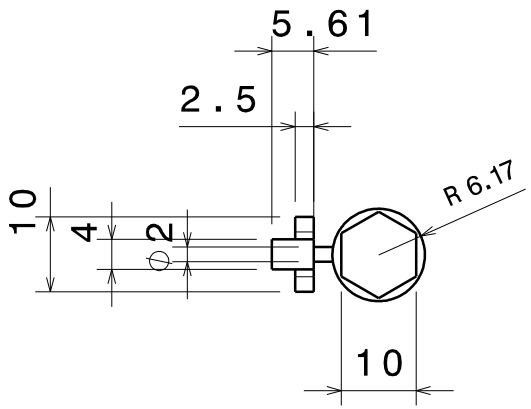


Vue de dessus

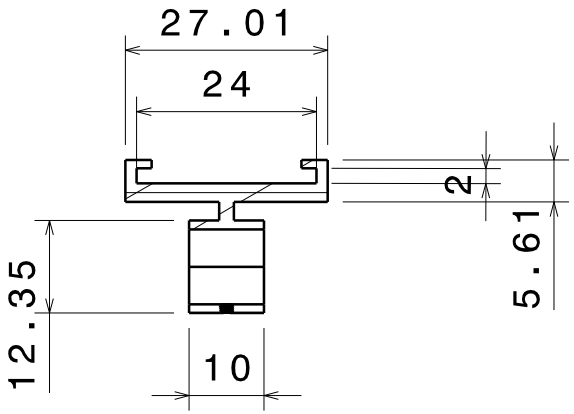
SIZE A4		désignation	matiére	PLA
Echelle 1:1	nb	2	suport servo 3	Référence	plastique



Vue de face



Vue de gauche



Coupe A-A

SIZE A4		désignation	matiére	PLA
Echelle 1:1		support stylo	Référence	plastique