

## **Treball final de grau**

**Estudi: Grau en Enginyeria Mecànica**

**Títol:**

**Disseny d'un braç robòtic controlat per un  
quant electrònic**

**Document 1: Memòria i Annexos**

**Alumne: Josep Rueda Collell**

**Tutor: Narcís Gascons Clarió**

**Departament: Enginyeria Mecànica i de la Construcció Industrial**

**Àrea: Enginyeria Mecànica**

**Convocatòria: Juny/2019**

**INDEX:**

1.	INTRODUCCIÓ.....	4
1.1.	Antecedents.....	4
1.1.1.	Peticionari.....	4
1.1.2.	Descripció de la situació .....	4
1.1.3.	Exposició del problema .....	4
1.2.	Objecte del problema .....	5
1.3.	Requeriments i abast .....	5
1.3.1.	Requeriments.....	5
1.3.2.	Abast del projecte.....	5
2.	SOLUCIÓ PROPOSADA.....	6
2.1.	Estudi de forma .....	7
2.1.1.	Braç perpendicular al vidre de la campana .....	7
2.1.2.	Braç paral·lel al vidre de la campana .....	8
2.1.3.	Conclusió.....	9
2.2.	Descripció del braç.....	10
2.2.1.	Descripció dels components del braç .....	11
2.3.	Descripció de la mà.....	14
2.3.1.	Descripció components de la mà.....	15
2.4.	Descripció del guant.....	17
2.4.1.	Descripció components del guant.....	17
2.5.	Hardware.....	19
2.6.	Consum.....	22
3.	DESCRIPCIÓ DEL PROGRAMA.....	25
3.1.	Funcionament del programa.....	25
3.2.	Codi.....	27
3.3.	Llibreries .....	27
3.4.	Funcions.....	29

4.	RESUM DEL PRESSUPOST .....	31
5.	CONCLUSIONS .....	32
6.	RELACIÓ DE DOCUMENTS.....	34
7.	GLOSSARI .....	35
	ANNEX A: DESCRIPCIONS TÈCNIQUES .....	36
A.1.	Peces de compra .....	37
A.1.1.	Servo-motors .....	37
A.1.1.1.	Servo-motor RDS5160 .....	37
A.1.1.2.	Servo-motor RDS3135 .....	38
A.1.1.3.	Servo-motor RDS3218 .....	38
A.1.1.4.	Servo-motor HS-785HB .....	39
A.1.2.	Hardware .....	39
A.1.2.1.	Arduino Uno .....	39
A.1.2.2.	Transformador DC-DC XH-M401 .....	40
A.1.2.3.	Giroscopi MPU6050 .....	40
A.1.2.4.	Multiplexor TCA9548A .....	41
A.1.3.	GHB 12V .....	41
	ANNEX B: CÀLCULS.....	42
B.1.	Càlcul de càrregues.....	43
B.1.1.	Càlcul de la força necessària per sostenir una proveta .....	43
B.1.2.	Determinació del pes de la pinça .....	44
B.1.4.	Càlcul de càrregues al canell.....	49
B.1.5.	Càlcul de càrregues al colze.....	50
B.1.6.	Càlcul de càrregues a l'espatlla .....	51
B.2.	Càlcul de càrregues dels elements .....	52
B.2.1.	Càlcul a l'extrem del braç.....	52
B.2.1.1.	Càlcul del punt a .....	53
B.2.1.2.	Càlcul al punt b .....	54

B.2.2.	Càlcul a l'espatlla .....	55
B.2.2.1.	Càlcul del punt a: .....	56
B.2.2.2.	Càlcul del punt b: .....	57
B.3.	Comprovació de la flexió.....	58
B.4.	Modelització matemàtica del braç: .....	61
B.5.	Càlcul matricial.....	63
ANNEX C:	PROGRAMA .....	68
C.1.	Codi Comentat .....	69
ANNEX D:	SIMULACIÓ INFORMÀTICA DEL BRAÇ .....	84
D.1.	Simulació del braç amb Matlab i Robotics Toolbox.....	85

## 1. INTRODUCCIÓ

### 1.1. Antecedents

#### 1.1.1. Peticionari

Sr. Marcel McCullough, científic dels laboratoris SCBD (Stem Cells Biology and Development) del Duve Institute, Bèlgica.



*Fig. 1:  
Logotip del Duve Institute*

#### 1.1.2. Descripció de la situació

L'institut de Duve és un institut enfocat a la recerca genètica i biotecnològica, el qual disposa de varis laboratoris dins del campus de la UCL (Universitat Catòlica de Louvain).

Aquest institut deu el seu renom a la política de centrar-se en la recerca amb finalitats mèdiques, així com incentivar la política de col·laborar amb altres empreses del mateix àmbit.



*Fig. 2:  
Sala de preservació criogènica.*

Per a la realització de les conseqüents investigacions, els seus laboratoris, estan equipats amb estacions d'anàlisi, sales blanques, cambres hiperbàriques, sales de preservació criogènica, estacions d'aïllament de proteïnes, entre d'altres instal·lacions.

#### 1.1.3. Exposició del problema

Habitualment, en aquest àmbit de recerca, s'han de fer processos en condicions potencialment perilloses, ja sigui perquè treballen amb elements a temperatures

massa fredes per ser aptes pel contacte humà o amb substàncies tòxiques o corrosives. Dur la conseqüent vestidura protectora, pot dificultar i, fins i tot, impossibilitar la realització de certes tasques.

## **1.2. Objecte del problema**

L'objecte del projecte és dissenyar un braç robòtic, capaç de copiar els moviments d'un guant, que duria un operari al voltant del braç.

## **1.3. Requeriments i abast**

### **1.3.1. Requeriments**

- El braç ha de cabre dins una campana de laboratori de 1400 mm d'ample, 1100 mm d'alt i 800 mm de profunditat.
- L'eix de la mà sempre ha d'anar paral·lel al terra.
- Cap articulació tindrà una acceleració superior a  $0,2 \text{ rad/s}^2$ .
- Cap articulació tindrà una velocitat superior a  $0,4 \text{ rad/s}$ .
- Les provetes tindran un diàmetre de 60 mm.

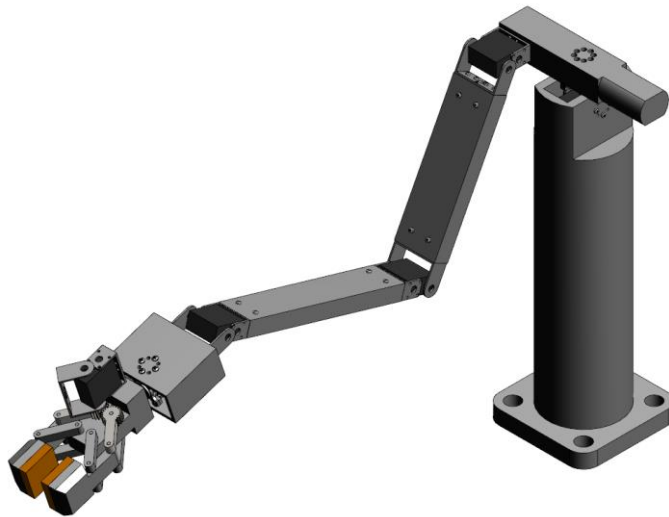
### **1.3.2. Abast del projecte**

La feina del projectista acaba un cop dissenyada la solució del problema i s'entrega el projecte al client. S'entregaran els documents necessaris per realitzar la construcció del braç robòtic:

- Plànols acotats complint la normativa vigent.
- Esquema electrònic del braç i del guant
- Codi per a la implementació en base Arduino.

## 2. SOLUCIÓ PROPOSADA

La idea proposada és construir un braç robòtic amb una estructura similar a la d'un braç humà, capaç de moure i decantar provetes o altres recipients de caràcter similar.



*Fig. 3:  
Visualització de la solució*

El braç ha de ser capaç de agafar i deixar una proveta o tub d'assaig, traslladar-lo en qualsevol dels 3 eixos.

Per a fer aquestes combinacions concretes de moviments, hem escollit un disseny el qual és incapaç de fer tots els moviments que podria fer un braç humà ja que implementar un braç que els pogués fer, tenint en compte com de reduïdes són les tasques que se'ns exigeixen, resulta redundant.

## 2.1. Estudi de forma

Tenint en compte que el braç haurà de treballar dins una campana d'un laboratori químic, he realitzat un estudi per veure quina és la forma més òptima per fer-lo servir.



Fig. 4:  
Model d'una campana de laboratori

Després de provar amb varies longituds pels diferents components que conformen el braç i varies localitzacions d'aquest dins la campana, hem acabat determinant dos possibles casos, col·locar el braç perpendicular al vidre de la campana o col·locar el braç paral·lel al vidre de la campana.

### 2.1.1. Braç perpendicular al vidre de la campana

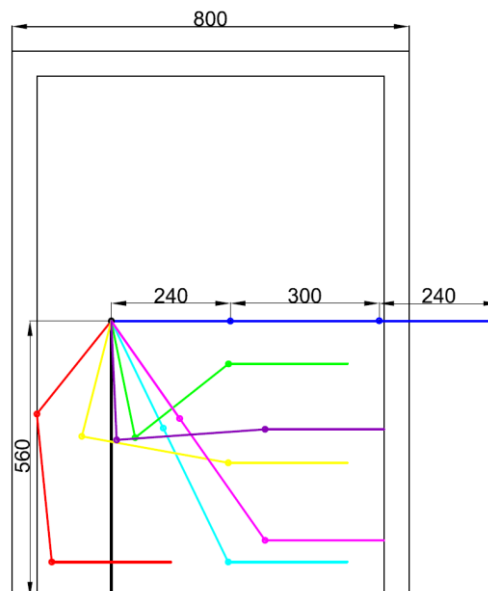


Fig. 5:  
Vista lateral del braç en el cas A per a varies posicions del braç

En aquest cas, estariem obligats a fer el tronc o esquena més llarg, a fi de poder aprofitar millor l'espai a dins de la campana.



La figura 5 ens presenta una vista lateral de la campana amb varies posicions dels tres components del braç: braç avantbraç i mà també en vista lateral.

Tal i com podem veure, amb aquesta distribució, podem controlar gran part de la base de la campana, però no podríem estirar el braç completament, ja que xocaríem amb l'interior d'aquesta.

Tanmateix, com que el robot copiarà els moviments que faci l'operari, aquesta distribució pot resultar en moviments incòmodes o fins i tot impossibles de realitzar per a un ésser humà, com per exemple, l'esquema representat amb color vermell on el canell hauria de quedar pràcticament perpendicular al braç.

#### 2.1.2. Braç paral·lel al vidre de la campana

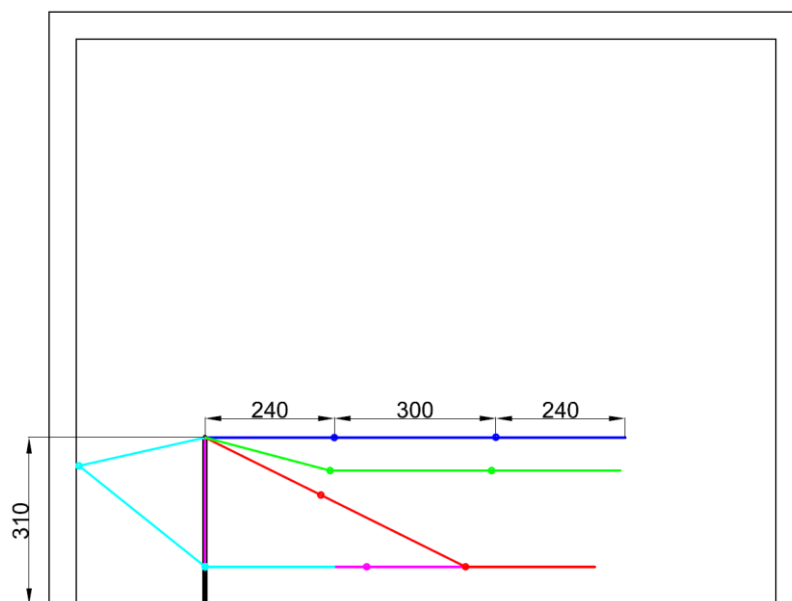


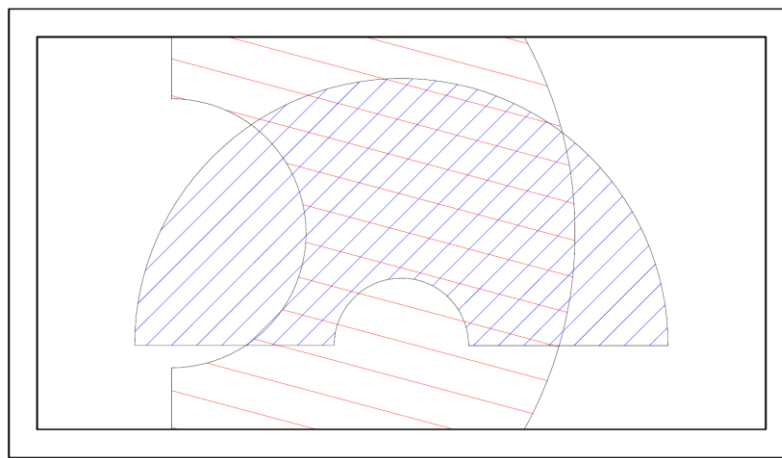
Fig. 6:  
Vista frontal del braç en el cas B per a varies posicions del braç

La figura 6 ens presenta una vista frontal de la campana amb varies posicions dels tres components del braç en vista lateral.

Tal i com podem veure a la figura, amb aquesta distribució, podríem controlar molt més bé la profunditat de la campana a cost de menys llibertat per controlar la base.

Aquesta distribució no permet que l'operari baixi massa el braç ja que podria xocar amb el terra amb molta més facilitat que al primer cas, per tant, és recomanable, alhora que pràctic, que l'operari s'assegui prop d'una taula la qual limiti els moviments del braç per sota d'aquesta.

### 2.1.3. Conclusió



*Fig. 7:  
Vista superior de la campana en una comparativa de  
rangs entre el cas A i el cas B*

En conclusió, tal i com podem veure a la figura 7 ens presenta els rangs dels braços a nivell del terra de la campana on el blau representaria el cas del braç perpendicular al vidre de la campana i el vermell representaria el cas del braç paral·lel a aquesta.

El problema que ens presenta el cas B és que tan si treball arran de terra com si treballa a certa alçada, quan gira corre el risc de xocar amb els límits de la campana.

Tanmateix el rang del cas del braç paral·lel és superior al del cas del braç perpendicular.

Tenint en compte el rang de treball superior del cas del braç paral·lel i la facilitat de control del braç per part de l'operari, ens hem decantat pel segon cas.

La màquina es dissenyarà partint de la suposició que l'operari, per tal de controlar el braç, es col·locarà, de costat, davant de la campana i girarà el cap per observar el moviment del braç robòtic el qual copiarà en paral·lel els moviments del guant de l'operari.

## 2.2. Descripció del braç

Tenint en compte l'estudi de forma de l'aparat anterior, el braç no ha de poder fer tots els moviments que podria acomplir un braç humà, dit això hem limitat el moviment del braç a tan sols cinc graus de llibertat més l'accionament de la pinça.

Li hem donat una forma similar a la d'un braç humà, però hem optat per reduir la longitud del braç i augmentar la de l'avantbraç, d'aquesta manera, els servomotors que s'encarregaran de controlar les articulacions seran de valors més semblants.

L'estructura anirà muntada sobre la peça "base braç" la qual estarà cargolada al terra mitjançant 4 cargols i faria la funció d'esquena, fent el paral·lelisme amb la fisiologia d'un ésser humà.

Un servomotor el qual ens hi referirem com a "servomotor esquena" estarà col·locat a l'extrem superior del la "base braç". Aquest mitjançant l'ajuda d'un "servo-block" aguantarà tota la resta de l'estructura.

El "servo-block" és una peça a mode d'esquelet per a servomotors que absorbeix el tallant i el moment



Fig. 8:  
Servo-block

que altrament, es carregarien a l'eix d'un servomotor i que el podrien malmetre.

Una peça anomenada "espatlla" estarà col·locada sobre el "servo-block". El "servomotor esquena", anteriorment esmentat, farà girar aquesta peça en un eix perpendicular al terra.

Aquesta peça "espatlla" en un costat hi tindrà el "servomotor espatlla" del qual hi penjarà la resta del braç mentre que l'altre costat s'hi podrà col·locar un contrapès per contrarestar part del moment que ha de suportar el "servo-block".

Aquest "servomotor espatlla" tindrà connectat el braç, aquest estarà connectat, al "servomotor colze", aquest a l'avantbraç i aquest al "servomotor canell" del qual en penjarà la pinça o mà.

Tots 3 servomotors ("servomotor espatlla", "servomotor colze" i "servomotor canell") treballen sobre el mateix pla, el qual és perpendicular al terra i l'orientació del qual ve determinada pel gir del "servomotor esquena".

A l'Annex B.3 Modelització matemàtica del braç, s'entra més en detall sobre el moviment del braç.

#### 2.2.1. Descripció dels components del braç

Hem calculat les necessitats mecàniques dels servomotors. Tal i com s'ha especificat al ANNEX B.1. Càlcul de càrregues i consegüentment hem escollit els servomotors en acord amb aquestes necessitats.

Els servomotors, posteriorment descrits, no utilitzaran unitats del sistema internacional sinó que expressaran el parell que poden subministrar en kg·cm ja que són les unitats que s'utilitzen a nivell comercial.

Al "servomotor esquena" només se li oposaran les carregues inercials del braç. Degut al poc pes dels components, la inèrcia és pràcticament negligible, conseqüentment hem agafat el servomotor HS-785HB de la casa *Hitec* o similars el qual està preparat per al nostre "servo-block". Aquest servomotor pot arribar a subministrar un parell de 13.2 kg·cm.

El "servomotor espatlla", tal i com indica el nom, estaria situat a l'espatlla i haurà d'aguantar el pes i el moment de la resta de l'estructura del braç, per tant serà el servomotor de més potència. Hem escollit el RDS5160 de la casa *Robot Digital Servos* o similars, el qual pot subministrar un parell de 60 kg·cm.

El "servomotor colze" estaria situat al colze. Hem escollit el RDS3218 de la casa *Robot Digital Servos* o similars, el qual pot subministrar un parell de 35 kg·cm.

El "servomotor canell" estaria situat al canell i és el que aguanta la pinça. Hem escollit el RDS3135 de la casa *Robot Digital Servos* o similars, el qual pot subministrar un parell de 20 kg·cm.

El "servomotor espatlla", "servomotor colze" i "servomotor canell" escollits prèviament, venen equipats amb 2 orelles o "brackets" a fi de facilitar la unió entre 2 peces. Aquest servomotors estan especialment dissenyats per a ser emprats com a articulacions per a robots és per això que reben el sobrenom de "Robot Servo". Hem escollit específicament aquest tipus ja que faciliten molt tan el muntatge del braç com el control d'aquest.

Podem trobar els aspectes tècnics dels anteriors servomotors al ANNEX A.1 Peces de compra.

La peça base braç tindrà allotjat dins seu a un servo-block amb el conseqüent servomotor. Aquest servo-block anirà collat a les parets de la peça base braç mitjançant cargols M3,5 x 8 mm. A l'extrem mòbil del servo-block hi col·locarem la peça espatlla i realitzarem la unió mitjançant cargols M3,5 x 12 mm. Podem

veure més detalls sobre la unió al PLÀNOL 8 SUBCONJUNT ESPATLLA-ESQUENA del Document 2: Plànols.

La peça espatlla anirà cargolada amb cargols auto-roscants a el "servomotor espatlla" i aquest utilitzant també cargols auto-roscants estarà unit al component braç. Utilitzant aquesta classe de cargols, no necessitem roscar forats a la peça ni tampoc necessitem al·locar-hi forats per a les femelles.

El component braç estarà compost per un perfil rectangular d'alumini de 54,76 mm per 18,78 mm i 2,04 mm de gruix i tindrà una longitud de 228,4 mm.

Al seu interior hi tindrà allotjat a cada extrem un prisma de PLA. Aquest prisma anirà cargolat al perfil d'alumini i a les orelles o "brackets" dels servomotors mitjançant Cargols Auto-roscants tal i com s'indica al PLÀNOL 11 SUBCONJUNT BRAÇ - SERVO - AVANTBRAÇ i al PLANOL 12 SUBCONJUNT ESPATLLA - SERVO - BRAÇ del Document 2: Plànols.

El component avantbraç estarà compost per un perfil rectangular d'alumini de 45 mm per 20 mm i 1,2 mm de gruix i tindrà una longitud de 240 mm.

De la mateixa manera que amb el perfil rectangular del braç, utilitzarem prismes de PLA i cargols auto-roscants per a fer la unió entre l'avantbraç i el "servomotor colze" i l'avantbraç i el "servomotor canell" tal i com s'indica al PLÀNOL 11 SUBCONJUNT BRAÇ - SERVO - AVANTBRAÇ i al PLANOL 10 SUBCONJUNT AVANTBRAÇ - SERVO - BASE CANELL del Document 2: Plànols.

La peça espatlla i la peça base braç són peces de fabricació les qual degut a la seva complexitat de mecanització es conformaran mitjançant tecnologies d'impressió 3D i conseqüentment utilitzarem PLA com a material de fabricació.

Hem escollit perfils d'alumini buits ja que és un material molt lleuger alhora que resistent. Tal i com podem veure a ANNEX B.1 Càlcul de càrregues, el parell que transmet el pes propi dels components braç i avantbraç no és excessiu, mentre

que el factor de seguretat calculat a l'ANNEX B.2. Càlcul de càrregues dels elements, ens indica un factor de seguretat pràcticament excessiu. Cal recordar que hem escollit la mida dels perfils a fi de poder-hi allotjar a dintre un prisma prou gros com per poder realitzar les unions amb orelles o "brackets" dels servomotors. Si no estiguéssim lligats a això, hauríem pogut escollir perfils molt més petits.

### 2.3. Descripció de la mà

La peça "base canell", tal i com hem dit anteriorment, es connecta al "servomotor canell". Aquesta, allotjarà al "servomotor mà" el qual controlarà la rotació de la pinça sobre un eix paral·lel respecte el terra, amb l'ajuda d'un "servo-block" que absorbirà les forces tallants i moments que normalment descarregarien contra l'eix del servomotor.

Al extrem del "servo-block", hi connectaríem la peça "suport mà" sobre la qual hi construiríem els mecanismes de la pròpia pinça.

Aquesta pinça, estarà constituïda per dos superfícies paral·leles recobertes de goma, que actuarien a mode de pinça. Cada una d'aquestes superfícies paral·leles les quals anomenarem "suport goma" està articulada per dos quadrats articulats que treballen en paral·lel.

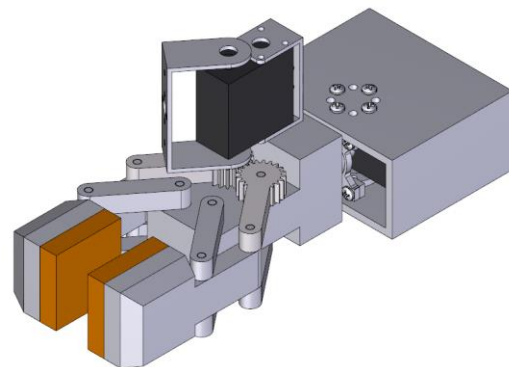


Fig. 9:  
Visualització 3D de la mà o pinça

Una biela de cada quadrat articulats estarà dentada, a mode d'engrenatge, a fi que totes dues superfícies de la pinça puguin treballar de forma sincronitzada en simetria.

Un servomotor el qual anomenarem "servomotor pinça" i estarà col·locat sobre la mà i tindrà una orella o "bracket" sobre la biela dentada i l'altre sobre la peça "suport mà" i s'utilitzarà per accionar el mecanisme de la pinça.

### 2.3.1. Descripció components de la mà

El "servomotor mà" estaria situat dins la peça "base canell" i és el que s'ocupa de la rotació de la mà. De la mateixa manera que el "servomotor esquena", aquest també va dins d'un "servo-block". Tenint en compte que l'únic parell a vèncer seria el vinculat al propi pes de la proveta en decantar-la i les inèrcies del conjunt pinça i càrrega, optem per menysprear-ho, per tant tornarem a escollir el servomotor específic per al "servo-block", el HS-785HB de la casa *Hitec* o similars, el qual pot subministrar un parell de 13.2 kg·cm.

El "servomotor pinça" estaria situat sobre la pinça i seria l'encarregat d'obrir-la i tancar-la. Hem escollit el RDS3135 de la casa *Robot Digital Servos* o similars, el qual pot subministrar un parell de fins a 20 kg·cm.

Podem trobar els aspectes tècnics d'aquests servomotor al ANNEX A.1 Peces de compra.

Totes les càrregues i pesos que estiguin aplicats a la pinça o a la mà són els més desfavorables pel servomotor de l'espatlla, ja que al ser el punt més allunyat del servomotor, el moment a suportar incrementa de forma dràstica per a masses relativament petites. Dit això, per a realitzar el muntatge del mecanisme, enlloc d'utilitzar unions mecàniques utilitzarem adhesiu.

Les varilles passaran a través de la peça "suport mà" i aplicarem adhesiu a les bieles perquè es mantingui subjecte.

El servomotor, tal i com havíem dit anteriorment, tindrà una orella o "bracket" a la "peça mà" i una altra a una de les bieles dentades.



Aquestes bieles dentades tindran un diàmetre primitiu de 20 mm i un mòdul de 0,75 el qual equivaldria a 15 dents, si fos un engranatge complet. Hem escollit un nombre de dents imparell a fi que els 2 engranatges puguin encaixar entre ells sense que quedin desfasats mig pas.

Tenint en compte que no podem posar cada orella o "bracket" del servomotor a sobre de cada biela perquè el punt d'encreuament de l'eix de cada biela varia en funció de l'angle d'obertura de la pinça, hem col·locat el servomotor sobre una de les bieles dentades de tal manera que l'eix de rotació del servomotor estigui alineat amb l'eix de rotació de la biela.

Per a unir el servomotor amb la pinça, també utilitzarem adhesiu, no només per la manca de pes que això implica, sinó també perquè les orelles o "brackets" del servomotor no tenen forats o encaixos amb els que realitzar unions amb peces mecàniques.

## 2.4. Descripció del quant

El quant estaria conformat per una tela que recobriria tot el braç on hi hauria adherits els sensors.

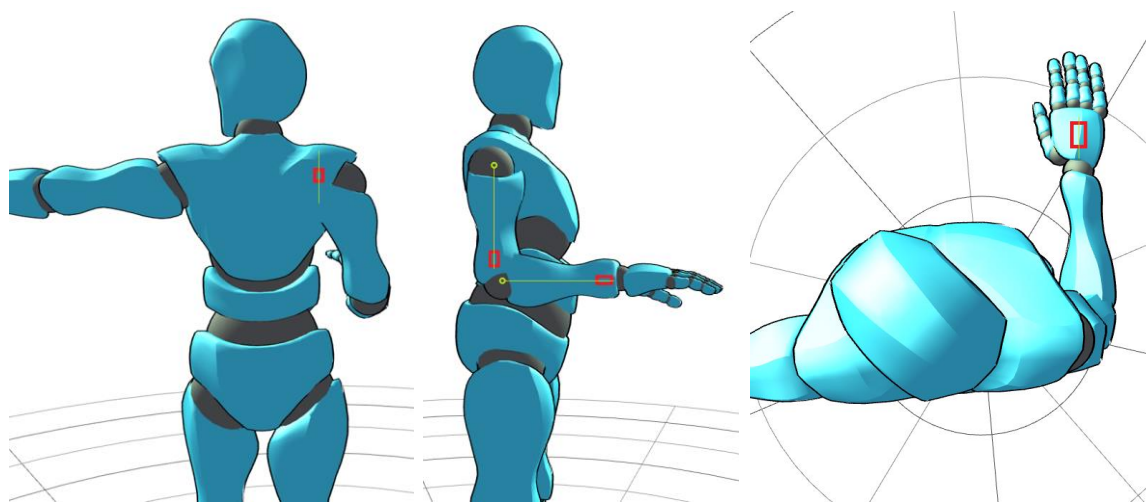
Els sensors estarà adherits amb unes abraçadores de tipus "velcro" que ajudaran a ajudar i fixar el quant i els sensors al lloc adient sobre el braç humà.

Els sensors estaran adherits mitjançant adhesiu a sobre les abraçaderes per tal de poder-les orientar correctament abans de ser utilitzades.

Aquests sensors, seran quatre giroscopis, els quals estaran repartits pel braç, esquena i mà, i un pulsador a la punta del polze per a accionar la pinça.

### 2.4.1. Descripció components del quant

Hem escollit com a giroscopis els giroscopis MPU-6050, els quals poden llegir velocitats i acceleracions tan lineals com angulars en tots tres eixos, i envien les dades per canal I<sup>2</sup>C al processador, en aquest cas, un Arduino Uno o similars.



*Fig. 10:  
Vista anterior, lateral i superior, d'una il·lustració d'una figura antropomòrfica, amb la localització dels giroscopis*

El protocol I<sup>2</sup>C té una velocitat per enviar i rebre dades de 100 kbits per segon mitjançant un bus sèrie el qual permet penjar-hi tants dispositius com necessitem. Tenint en compte que haurem d'adquirir dades des de 4 sensors diferents, de forma simultània i de forma continuada, haver escollit un sensor amb protocol I<sup>2</sup>C resulta una elecció excel·lent.

El primer giroscopi anirà darrere l'espatlla i controlarà l'angle de gir de l'esquena respecte el terra, és a dir, l'angle del pla on treballaria la resta del braç.

Un segon, situat a prop del colze controlarà la rotació del que pròpiament dit, seria el braç respecte l'espatlla.

Un tercer a l'avantbraç, com més allunyat del colze millor per tal de captar amb més precisió la rotació de l'avantbraç respecte el colze.

S'ha de tenir en compte, que si el situem sobre una part del braç que roti sobre un eix diferent que el del colze ens podria alterar les lectures del giroscopi, s'ha d'intentar que els giroscopis romanguin perpendiculars al eix de treball.

El quart giroscopi estarà col·locat al revers de la mà i controlarà la rotació de la pinça.

És de suma importància que els sensors tinguin l'eix de treball perpendicular o paral·lel al pla del terra, tal i com s'indica a la figura 14. Com més alineats treballin al seu eix teòric de funcionament, més fidels seran les lectures dels sensors.

Tal i com s'ha especificat anteriorment, no ens caldrà un giroscopi per a calcular la orientació del canell. El canell sempre s'encarregarà que la mà o pinça estigui paral·lela al terra, i podem calcular la inclinació d'aquest a partir de les lectures del braç i l'avantbraç.

Tal i com s'explica més endavant, en l'apartat 2.5 Hardware, aquests giroscopis duen una adreça interna que desencadena problemes d'identitat quan se'n

connecta més d'un alhora a un Arduino, per tant, ens veiem obligats a utilitzar un multiplexor I<sup>2</sup>C.

El polsador, el qual estarà col·locat sobre el polze i que podrem accionar prement-lo contra l'índex serà de característiques binaries (obrir o tancar), per tant, no permetrà un control analògic de la obertura de la mà, és a dir, que sempre farà la mateixa força contra qualsevol proveta.

## **2.5. Hardware**

Com a processador central, hem utilitzat una placa de tipus "Arduino Uno" o similars. Aquesta placa és cada cop més utilitzada per a aplicacions de caràcter professional degut a la facilitat d'ús i a l'ample ventall de sensors, actuadors i software així com altres dispositius, que permeten compatibilitat amb ell. Tanmateix, també duu incorporat sistemes de protecció contra sobretensions i curtcircuits, el qual el fan un dispositiu molt robust i segur.

Tal i com hem explicat anteriorment, com a giroscopis utilitzarem el sensors MPU6050. El rang de velocitat d'aquests giroscopis és molt superior a la velocitat de moviment d'un braç humà. Referent a la sensibilitat aquesta és molt superior a la que se'ns exigia com a condició de funcionament, per tant és adequat.

Tanmateix, el contractista vol tenir llibertat per a realitzar canvis i millores al disseny original al llarg del temps, si ho troba necessari, per tant reafirmem la tria d'aquest dispositiu com a processador central.

També cal afegir, que tal i com s'explica més endavant, en l'apartat 3.2 Llibreries, aquests giroscopis poden corregir l'error d'integració de la posició utilitzant els eixos de la gravetat per a orientar-se, per tant, podem afirmar que aquest component és ideal per al nostre projecte.

Cada un dels sensors, es comunica amb la unitat de control mitjançant protocol I<sup>2</sup>C. Cada sensor te assignada internament la adreça 0x68 i disposa d'un pin digital per poder-la canviar a 0x69. Si hi ha un conflicte d'adreces entre dos o més sensors, la unitat de control es bloqueja i obliga a fer un reinici forçat a base de treure-li l'alimentació i tornar-li a posar.

Originalment es recomanava assignar a tots els sensors l'adreça 0x69 tret del que volíem llegir, que li assignaríem la adreça 0x68 mitjançant una de les sortides digitals del controlador, val a dir, que tal i com hem dit abans, tot i que no estiguem llegint en cap moment de l'adreça 0x69, com que hi ha un conflicte d'adreces per part dels altres tres sensors de forma simultània, multiplexar els sensors utilitzant les sortides digitals del controlador, resulta ser una opció inviable.

Dit això ens veiem obligats a utilitzar un multiplexor I<sup>2</sup>C, en aquest cas, el Multiplexor TCA9548A .

El multiplexor TCA9548A te tres entrades digitals que li permeten modificar la seva adreça interna de 0x70 fins a 0x77. Això es fa servir per quan utilitzem varis multiplexors alhora, fins a un màxim de 8 multiplexors per a un total de 64 sensors.

Tanmateix aquest multiplexor te un registre intern que des de la unitat de control, utilitzant una comanda, podem controlar quin dels 8 ports interns volem escollir.

Dit això, tot i que tots els sensors tenen la adreça interna de 0x68, ja no hi ha conflicte d'adreces ja que és el multiplexor qui va alternant entre els diferents dispositius però la nostra unitat de control només veu la seva adreça, la qual és 0x70.

Finalment, tal i com podem veure als plànols hem col·locat un llit de ferrita entre les masses de la part digital i la part analògica. En ocasions podem observar com la part digital d'uns sistema causa interferències a la part digital analògica. Per

evitar-ho utilitzem un llit de ferrita, el qual actua com una inductància, és a dir, un filtre pas baix per a soroll d'alta freqüència. D'aquesta manera donem més robustesa al sistema.

## 2.6. Consum

En aquest apartat estudiarem les necessitats energètiques dels diferents components elèctrics i electrònics alhora d'escollir una font.

La intensitat consumida pels sensors i el multiplexor són de l'ordre de  $\mu\text{A}$ . Tenint en compte que són tan baixos, els menysprearem.

L'Arduino, consumeix 0,05 A per sortida. Si ha de controlar 6 motors i el polsador, podem arrodonir a 0,4 A.

D'acord amb el document plànols, podem veure que els servomotors estan agrupats en grups de dos i cada parella de servomotors estan alimentats pel mateix convertidor. Això és degut a que per suplir les demandes de parell dels diferents servomotors, tal i com veiem a les conseqüents "datasheet's", necessitem un voltatge específic, i en el cas de varis servomotors, aquest coincideix.

Les pèrdues dels convertidors es quantificaran, tenint en compte el rendiment. Cal recordar que l'Arduino, també estarà alimentat per un convertidor però a diferència dels altres dispositius, aquest no ha de compartir el convertidor.

Haurem de tenir en compte que aquests convertidors seran regulables, dit això, quan s'implementi, l'operari, amb l'ajuda d'un multímetre, s'haurà d'assegurar que regula el potenciòmetre de cada convertidor de forma correcta per a tenir el voltatge adequat a la sortida, tal i com s'indica als plànols.

Primerament, comprovarem que els convertidors, poden suplir les nostres necessitat d'intensitat i de potència.

El convertidor que haurà de suportar més potència, serà el que ha d'alimentar els servomotors del colze i de l'espatlla quan aquests han de suportar el parell màxim ja que son els motors de major potència. Tanmateix el consum dels servomotors no es pot trobar a la seva "datasheet", conseqüentment, de forma experimental,

hem mesurat el corrent de cada un dels servomotors pel cas més desfavorable, és a dir, quan el parell a resistir sigui màxim.

El "servomotor espatlla" pel qual hem escollit el RDS5160, i el "servomotor colze" pel qual hem escollit el RDS3135 els quals s'alimenten a 8,4 V, tenen un consum aproximat de 3 A i 1,2 A respectivament. Si utilitzem la formula de la equació 1, això es tradueix a 35,3 W de potència i 4,2 A, molt per sota de les limitacions de potència del convertidor.

Seguidament calcularem les necessitats energètiques dels altres convertidors, a fi de poder calcular el consum total:

El "servomotor canell" i el "servomotor pinça" pels quals hem escollit el RDS3218 i els quals s'alimenten a 6.8 V, tenen un consum aproximat de 0,8 A cadascun. Això es tradueix a 10,9 W de potència i 1,6 A.

$$P = V \cdot I \quad (Eq. 1)$$

El "servomotor esquena" i el "servomotor mà" pels quals hem escollit el HS-785HB i els quals s'alimenten a 6 V, tenen un consum aproximat de 0,4 A cadascun. Això es tradueix a 4.8 W de potència i 0,8 A.

Utilitzant la següent fórmula, podem calcular la potència que ha de subministrar la font:

$$P_u = P_{Arduino} + \sum P_{Motors} \quad (Eq. 2)$$

$$P_u = 5 \cdot 0,4 + 35,3 + 10,9 + 4,8 = 53 \text{ W}$$

Hem de tenir en compte que hem menyspreat els sensors. També hem de recordar que el rendiment dels convertidors, acordant amb la "datasheet" és de 0,94.



$$P_c = \frac{P_u}{\text{Rendiment}} = 56,4 \text{ W} \quad (\text{Eq. 3})$$

La potència consumida pel sistema seria, doncs, de 56,4 W.

Ens hem decantat per la font GHB 12V, la qual pot suplir les nostres necessitats energètiques i duu refrigeració incorporada. A més els pins van agrupat en parelles, la qual cosa implica que tenim el doble de pins de Vout i de GND, la qual cosa facilitaria la instal·lació tenint en compte que hem de connectar 4 convertidors al darrere.

Aquests 56.4 W per a una sortida de 12 V es traduirien a 4.7 A, els quals estan per sota de les limitacions de la font.

### **3. DESCRIPCIÓ DEL PROGRAMA**

#### **3.1. Funcionament del programa**

Inicialment hem parametritzat les condicions inicials com serien les longituds dels eixos del braç i les dimensions en l'eix X de la campana. Seguidament els hi assignem una variable constant.

El processador, adquireix la lectura dels angles de cada sensor de forma continuada.

Cada cop que s'executa el programa, pren aquests valors dels angles prèviament adquirits, i en calcula la posició X, Y i Z mitjançant cinemàtica directa. El programa activa un pilot per assegurar-se que aquesta operació només es realitzi un cop cada 4 segons, tot i que els sensors no deixen en cap moment d'adquirir els valors dels angles.

Seguidament, el programa comprova si aquesta posició X de la pinça és dins de la dimensió X de la campana. Si és a dins, transmet els valor dels angles al seus respectius servomotors. Si fos fora, i transmetéssim els angles als servomotors implicaria que la pinça xocaria amb les parets de la campana, per tant, si es a fora, assignem al valor X de la posició de la pinça, el mateix valor X del límit de la campana i li fem calcular un altre cop, mitjançant cinemàtica inversa, els valors del angles de cada articulació. Finalment assignem els nous angles calculats al seu respectiu servomotor.

Cal recordar que el "servomotor canell" no te assignat cap giroscopi. L'angle de gir que se li envia a aquest servomotor, es calcula en funció dels angles de l'espatlla i del colze a fi de sempre estar paral·lels al pla del terra.

La rotació de la mà, portada a terme pel "servomotor mà", no està subjectada a cap dels càlculs anteriors. Tal i com podem veure al ANNEX A.2 Modelització matemàtica del braç, per poder simplificar els càlculs matricials, no hem tingut en compte la rotació de la mà en les equacions, ja que només repercuteix en la rotació del punt final del braç, no en la posició en cap els 3 eixos d'aquesta.

Finalment, després d'haver enviat les consignes d'angle als servomotors, es comprova si el pulsador ha estat premut o no a fi de controlar la obertura i el tancament de la pinça utilitzant el "servomotor pinça".

Un cop han passat 4 segons, es fa un "reset" del pilot que s'ocupava que els càlculs només es realitzessin un cop per cicle, i es fa un "reset" al "timer" a fi de tornar a iniciar un cicle.

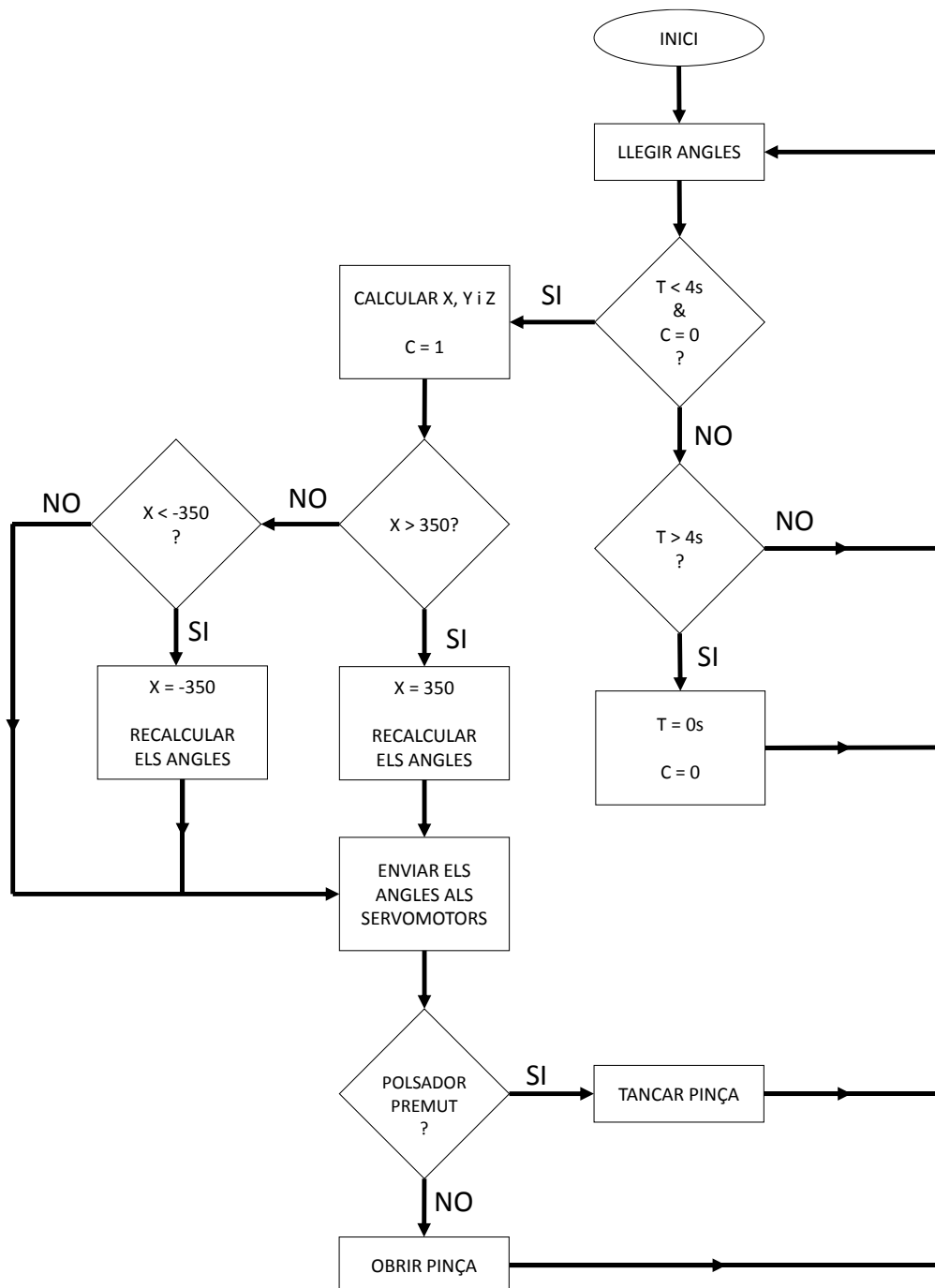


Fig. 11:  
Diagrama de flux del programa

### 3.2. Codi

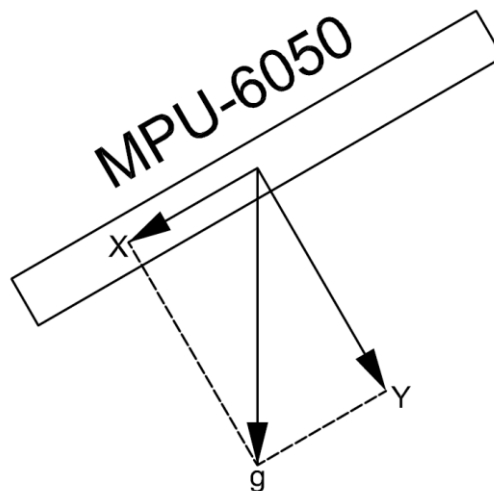
El codi, el qual és un requeriment del projecte, el podrem trobar al ANNEX C.1. Codi Comentat, tanmateix també el podrem trobar en format digital al disc annexat al projecte, juntament amb les llibreries necessàries perquè funcioni.

### 3.3. Llibreries

Hem utilitzat una llibreria específica anomenada "MPU6050\_tockn" la qual serveix per a adquirir totes les dades possibles que ens pot donar el giroscopi "MPU6050" les quals inclouen acceleracions, acceleracions angulars, velocitats i velocitats angulars en tots 3 eixos i que incorpora un integrador per a calcular el desplaçament i el gir.

Cada llibreria té assignades les adreces internes necessàries per a connectar-se amb el giroscopi així com tot un seguit de subrutines que l'usuari pot utilitzar per a adquirir els valors de les lectures desitjades.

A més la llibreria duu incorporada un tros de codi que utilitza la gravetat per a poder corregir l'error d'integració del gir.



*Fig. 12:  
Esquema de la lectura de la força de la gravetat en l'eix X i Y d'un hipotètic sensor MPU-6050*

Cada sensor, s'ha estudiat en quin eix treballarà. Partint d'aquí, s'ha retocat el codi perquè en funció de les lectures de la gravetat dels altres dos eixos, pugui calcular la seva inclinació i corregir el valor instantani del gir.

El sensor que controla el gir de l'esquena, el qual gira en un pla perpendicular al terra, i conseqüentment, paral·lel a la gravetat, no pot corregir aquest error, per

tant, duu incorporat un filtre de soroll, que menysprea les lectures de velocitat dins el rang de  $\pm 0.5$  °/s, per tal que no s'acumuli error, almenys, mentre el sensor romanguí quiet.

Tenint en compte els aspectes anteriors, cada sensor, disposa de la seva pròpia llibreria amb les particulars modificacions que repercuteixen al seu concret funcionament.

Per defecte, el compilador d'Arduino, inclou la llibreria "servo.h" la qual ofereix un sistema fàcil d'utilitzar per a controlar servomotors. Tanmateix nosaltres hem optat per a utilitzar la llibreria "VarSpeedServo.h" la qual ofereix les mateixes prestacions que l'anterior en quant a control de posició, amb la possibilitat de controlar la velocitat de gir escollint valors des de 1 a 255 on 1 és molt a poc a poc i 255 a màxima velocitat.

Seguidament prosseguirem a les funcions utilitzades en el programa principal que provenen de les llibreries:

```
mpu6050X.update();
```

Aquesta funció ordena als sensors que refresquin els seus valors actuals.

```
mpu6050X.calcGyroOffsets();
```

En iniciar el programa, aquesta comanda fa que el giroscopi X, on X representa el sensor segons el criteri anteriorment imposat, calibri els seus offsets interns per tal d'eliminar o reduir els errors de lectura. Aquesta funció s'executa un cop per a cada sensor i aquests han de romandre quiets durant el procés.

```
mpu6050X.getAngleX();
```

EL programa obté el valor del gir en l'eix X. Val a dir que hem modificat totes les llibreries perquè l'eix X s'adapti al eix de gir en el que treballarà el guant, prèviament descrit al apartat 2.4 Guant.

### 3.4. Funcions

El programa està organitzat sobre un simple bucle o "loop" el qual crida les conseqüents funcions quan les necessita. En aquest apartat les explicarem.

Hi ha funcions que van lligades a llibreries específiques per a cada sensor, tal i com havíem explicat a l'apartat anterior, dit això, quan es parli del sensor 0 estarem parlant del sensor que detecta la rotació de l'esquena, quan es parli del sensor 1 estarem parlant del sensor que llegeix la inclinació de l'espatlla, quan es parli del sensor 2 estarem parlant del sensor que llegeix la inclinació del avantbraç i, finalment, quan es parli del sensor 3 estarem parlant del sensor que identifica la rotació de la mà.

Començarem per les funcions declarades al propi codi principal:

```
void tcaselect(X);
```

Utilitzant aquesta funció podem escollir amb quin dels sensors ens comunicarem, on X representa el sensor segons el criteri anteriorment imposat.

```
void adquirir_dades();
```

Aquesta funció adquireix els valors dels angles de tots els sensors.

```
void motors();
```

Aquesta funció transmet els valors corresponents dels angles als conseqüents servomotors.

```
void imprimir_sensors();
```

Aquesta funció també s'usa exclusivament amb finalitats de "debugging". Ens imprimeix els valors "raw" llegits directament dels sensors, sense que hagin estat tractats de cap manera.

```
void calculs();
```

Aquesta funció calcula el valor de la posició de la pinça utilitzant cinemàtica directa, comprova si la X surt fora de la campana, tal i com hem explicat a l'apartat anterior, i si és necessari, recalcula els angles que ho requereixin.

```
void imprimir_valors();
```

Aquesta funció també s'usa exclusivament amb finalitats de "debbing". Ens imprimeix els valors dels angles, i de la posició de la pinça, així com els angles recalculats en el cas que ho hagin hagut de ser.

#### **4. RESUM DEL PRESSUPOST**

El pressupost d'execució material incloent les despeses generals, el benefici industrial i l'IVA, el cost total suma un total de **2.678,22 €** (dos mil sis-cents setanta-vuit euros i vint-i-dos cèntims).

Per informació més detallada sobre el pressupost, consultar el **Document Pressupost**.



## 5. CONCLUSIONS

Aquest braç està pensat per treballar dins d'una campana de químics, a fi de realitzar tasques en unes condicions on un braç humà no podria fer-ho correctament degut al risc a les lesions i perjudicis contra entitats humanes. Alhora s'ha de tenir en compte la dificultat que implica per a una persona, treballar amb la càrrega d'un equip de protecció, el qual sol minar les capacitats humanes a cost d'incrementar la seguretat per a certes tasques.

El resultat ha estat un braç de 5 graus de llibertat més el control de la obertura de la pinça. Tanmateix, el requeriment que la mà de la pinça s'havia de mantenir sempre paral·lela al pla del terra, fa que virtualment sigui un robot amb 4 graus de llibertat més el control de l'obertura de la pinça.

El disseny del braç no permet que la pinça topi contra el vidre de les parets de la campana on treballa, tal i com s'havia especificat als requeriments i ja que la normativa no permet. Tanmateix si que podria topar contra el terra, ja que no s'ha tingut en compte cap limitació respecte aquest.

Quan l'operari obliga al robot a fer un moviment, el qual faria xocar el braç contra el vidre de la campana, el programa, tal i com s'havia especificat als requeriments s'ha d'abstenir de topar-hi. Quan es dona aquesta situació, el braç passa per un seguit de càlculs amb base trigonomètrica que li permeten emular moviments semblants als de l'operari en quant al desplaçament lineal de la pinça, però divergents en quant als valors absoluts dels angles de les articulacions del quant.

El braç té limitacions respecte als angles de treball. Les equacions amb base trigonomètrica que el nostre sistema duu integrades, poden calcular varis valors diferents pels angles per a certes posicions del braç. D'aquí que puguin sorgir interpretacions errònies de la posició real del braç. Dit això, l'operari s'ha d'abstenir a que la espatlla, tingui una inclinació superior a un pla paral·lel al terra. Recordem que aquesta limitació només sorgeix quan hi ha risc que el braç topi amb les parets de la campana.

El braç té un temps de resposta de 4 segons o menys en funció del punt on es trobés del "loop" en el programa. Es podria haver implementat el programa de tal manera que el programa s'hagués anat executant de forma continua, enlloc de utilitzant cicles de 4 segons, però tenint en compte que el braç ha de transportar substàncies perilloses, ens ha semblat que fer-ho per etapes facilitaria la feina al operari ja que aniria més a poc a poc i li transmetria una sensació de seguretat.

Josep Rueda Collell  
Graduat en Enginyeria Mecànica.

Girona, 6 de juny de 2019

## **6. RELACIÓ DE DOCUMENTS**

Document 1: Memòria i Annexos

Annex A: Descripcions tècniques

Annex B: Càlculs

Annex C: Programa

Annex D: Simulació Informàtica del braç

Document 2: Plànols

Document 3: Plec de Condicions

Document 4: Estat d'amidaments

Document 5: Pressupost

## **7. GLOSSARI**

USB	Bus Sèrie Universal
PLA	Àcid Polilàctic
I <sup>2</sup> C	Inter-Integrated Circuit

# ANNEX A: DESCRIPCIONS TÈCNIQUES

## A.1. Peces de compra

En aquest apartat, hi posarem fragments rellevants de les "datasheet's" dels components més importants.

### A.1.1. Servo-motors

#### A.1.1.1. Servo-motor RDS5160

Per a l'articulació de l'espatlla, tal i com s'ha explicat anteriorment, utilitzarem el servomotor RDS5160 de la casa DSServos o similars.

Voltatge:	6 - 8,4 V
Parell:	58 kg·cm (6 V)
	65 kg·cm (7,4 V)
	70 kg·cm (8,4 V)
Pes	162g
Mida	65 x 30 x 48

*Taula 1:  
Detalls tècnics del  
Servo-motor RDS5160*



*Fig. 12:  
Servo-motor RDS5160*

#### A.1.1.2. Servo-motor RDS3135

Per a l'articulació del colze, tal i com s'ha explicat anteriorment, utilitzarem el servomotor RDS5135 de la casa DSServos o similars.

Voltatge:	6 - 8,4 V
Parell:	26 kg·cm (6 V)
	31 kg·cm (7,4 V)
	35 kg·cm (8,4 V)
Pes	60 g
Mida	64 × 55,6 × 20

*Taula 2:  
Detalls tècnics del Servo-  
motor RDS3135*



*Fig. 13:  
Servo-motor RDS3135*

#### A.1.1.3. Servo-motor RDS3218

Per a l'articulació del canell i per a l'accionament de la pinça, tal i com s'ha explicat anteriorment, utilitzarem el servomotor RDS5135 de la casa DSServos o similars.

Voltatge:	5 - 6,8 V
Parell:	20 kg·cm (5 V)
	36 kg·cm (6,8 V)
Pes	60 g
Mida	40 × 20 × 40,5

*Taula 3:  
Detalls tècnics del  
Servo-motor RDS3218*



*Fig. 14:  
Servo-motor RDS3218*

#### A.1.1.4. Servo-motor HS-785HB

Per a l'orientació de l'esquena i l'orientació de la pinça, tal i com s'ha explicat anteriorment, utilitzarem el servomotor RDS5135 de la casa DSServos o similars els quals aniran allotjats dins un servoblock.

Voltatge:	5 - 6 V
Parell:	13,5 kg·cm (6 V)
Pes	60 g
Mida	40 × 20 × 40,5

*Taula 4:  
Detalls tècnics del  
Servo-motor HS-785HB*



*Fig. 15:  
Servo-motor HS-785HB*

### A.1.2. Hardware

#### A.1.2.1. Arduino Uno

L'Arduino Uno, de la casa Arduino o similars, és el processador que hem escollit per a realitzar el control tan dels sensors del guant com els actuadors del braç.

Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

*Taula 5:  
Detalls tècnics del Arduino Uno*



*Fig. 16:  
Arduino Uno*



#### A.1.2.2. Transformador DC-DC XH-M401

Cada component necessita el seu voltatge específic. Aquest regulador de voltatge és ideal per l'ample rang de voltatges amb els quals pot treballar.

Regulating mode:	PWM
Input voltage:	DC 4 - 40V
Output range:	DC 1.25 - 36V
Maximum current:	8A
Maximum power:	200 W
Conversion efficiency:	94%
Switching frequency:	180KHz
Size:	61 × 41 × 27mm

Taula 6:  
Detalls tècnics del XH-M401



Fig. 17:  
XH-M401

#### A.1.2.3. Giroscopi MPU6050

El dispositiu MPU6050, ens és molt útil ja que és capaç tant de llegir velocitats angulars com acceleracions lineals.

Tal i com hem especificat a l'apartat 3.3 Llibreries, podem corregir l'error d'integració de la velocitat angular utilitzant els acceleròmetres, per a saber en quina direcció es troba l'eix de al gravetat.

VCC	2.38 V - 5.46V
Giroscopi	
Rang	± 2000 °/s
Sensibilitat	0,2 °/s
Acceleròmetre	
Rang	± 16 g
Sensibilitat Eix X	± 0,05 g
Sensibilitat Eix Y	± 0,05 g
Sensibilitat Eix Z	± 0,08 g

Taula 7:  
Detalls tècnics del MPU6050

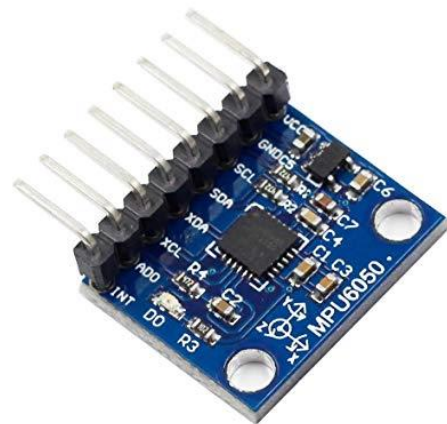


Fig. 18:  
MPU6050

#### A.1.2.4. Multiplexor TCA9548A

Aquest dispositiu l'utilitzarem per eliminar l'error que surgeix quan hi ha un conflicte d'adreces entre varis dispositius I2C.

Com que tenim quatre giroscopis i no podem assignar una adreça diferent a cada un, optarem per utilitzar aquest multiplexor.

VCC	1,65 V - 5,5 V
-----	----------------

Taula 8:  
Detalls tècnics del TCA9548A



Fig. 19:  
TCA9548A

#### A.1.3. GHB 12V

Ens hem decantat per la font GHB 12V, la qual pot suplir les nostres necessitats energètiques i duu refrigeració incorporada.

$V_{in}$	230 V ~
$V_{out}$	12 V =
$I_{max}$	15 A
$P_{max}$	180 W

Taula 9:  
Detalls tècnics de la font  
GHB 12V



Fig. 20:  
GHB 12V

# ANNEX B: CÀLCULS

### B.1. Càlcul de càrregues

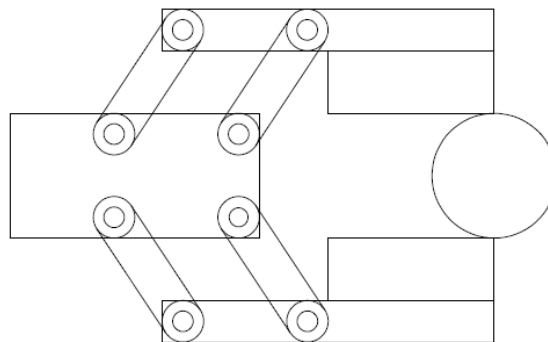
Calcularem les càrregues que han de suportar els motors de les articulacions en els casos més desfavorables pel braç.

Per al càlcul dels components, tal i com hem esmentat a la memòria, hem utilitzat perfils rectangulars d'Alumini..

També recordem que per a les condicions de disseny, vam suposar que no aixecaríem un pes superior a 0.200 g, i que no accediríem una velocitat superior a 0.4 m/s que l'acceleració màxima seria de  $0.2 \text{ m/s}^2$ . Per a aquests valors, de velocitat i acceleració, la força deguda a l'acceleració radial i tangencial que experimenta la proveta, tenint en compte la longitud del braç en un cas on estigués purament estirat, eren pràcticament nul·les.

#### B.1.1. Càlcul de la força necessària per sostenir una proveta

La pinça que utilitzarem per a sostenir provetes, tindrà un recobriment de goma per incrementar la fricció.



*Fig. 21:  
Representació de la pinça sostenint una proveta en la posició més desfavorable*

Per determinar el coeficient de fricció estàtic entre el vidre i la goma, hem anat a buscar el valor a una taula comparativa entre materials. En el nostre cas hem

utilitzat la taula de "The engineering Toolbox". El coeficient de fricció entre el vidre (una proveta convencional) i la goma, es pot trobar és de  $\mu = 0.85$ .

Tal i com hem especificat, en el plec de condicions, el diàmetre màxim d'una proveta seria 60 mm, i representaria el cas en el que el motor que acciona la mà, treballa de forma més desfavorable.

$$F_F = \mu \cdot N \quad (Eq. 4)$$
$$N = 2.36 \text{ N}$$

La força de fricció, tindrà el mateix valor que el pes, i la força Normal, és la força que li haurem d'aplicar a la proveta amb la pinça a fi que no rellisqui. Si sabem que el pes de la proveta és de com a màxim 200 g, la força normal per sostenir-la serà de 2.36 N.

### **B.1.2. Determinació del pes de la pinça**

Tal i com hem explicat a la memòria, hem utilitzat un plàstic extrudible, amb una densitat de  $1,25 \text{ g/cm}^3$  segons el fabricant.

La mà, tenint en compte que hi hem d'incloure un servomotor de poca potència per obrir i tancar la pinça, li podem estimar una massa no superior a 400 grams.

### B.1.3. Càlcul del parell del servomotor de la pinça

Suposarem que la força que haurà de fer la peça contra la proveta serà la hem calculat a l'apartat anterior, és a dir que  $F_P = 2,36$  N. Suposarem que la força deguda a l'acceleració és menyspreable.

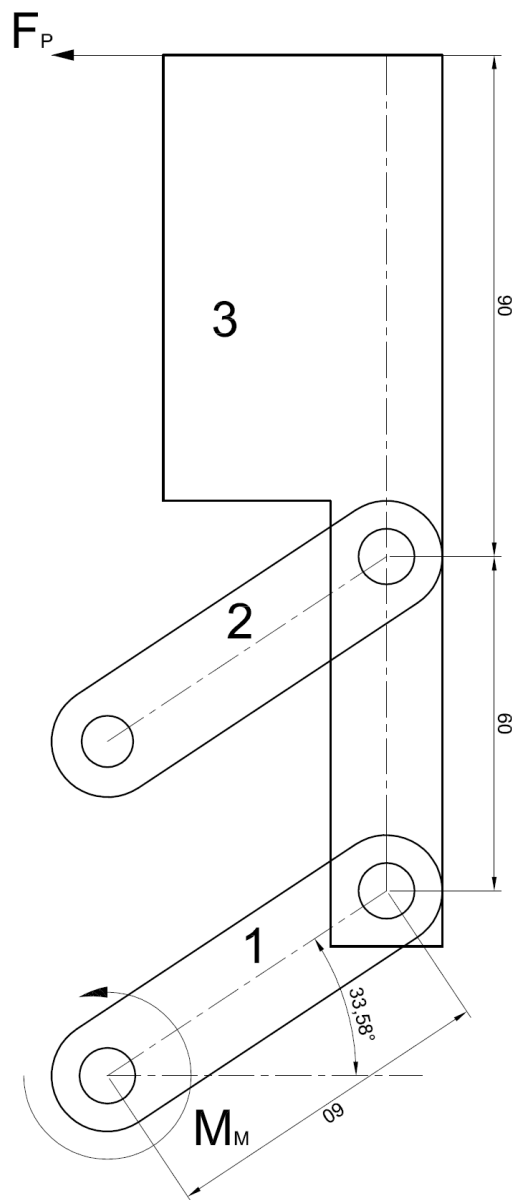


Fig. 22:  
Distribució de les forces i moments externs, aplicats a la meitat dreta de la pinça

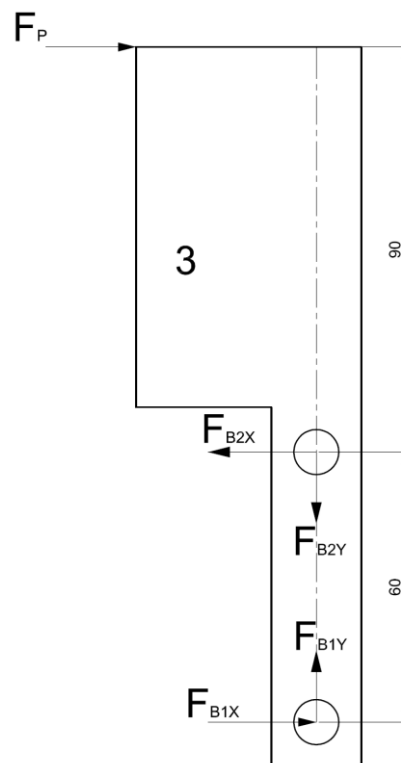


Fig. 23:  
Representació de les forces en una  
vista superior de la peça "suport  
goma"

Aplicant les fórmules dels sumatoris de forces i de moments, podem aïllar les forces en X de la biela 2 i les forces en X de la biela 1.

$$F_{B2X} = 5,9 \text{ N}$$

$$F_{B1X} = 3,54 \text{ N}$$

Com que sabem que la biela 2 no pot transmetre tallant degut a les seves condicions de contorn, i al fet que les forces estiguin aplicades a un dels seus extrems, podem assumir que només transmetrà forces axials.

Dit això, com que sabem que el valor de l'angle d'inclinació de la biela val  $\alpha = 33,58^\circ$  i sabem el valor de la força en X de la biela 2, podem aïllar utilitzant trigonometria, el valor de la força en Y de la biela 2.

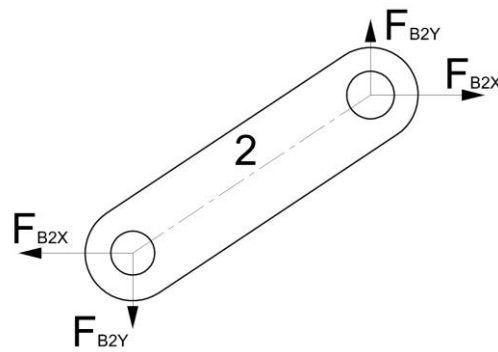


Fig. 24:  
Representació de les forces en una  
vista superior de la biela 2

$$F_{B2Y} = 3,91 \text{ N}$$

Tenint en compte que la força de la biela 1 en Y i la força de la biela 2 en Y tenen el mateix valor però sentit contrari, podem saber el valor de la força en Y de la biela 1.

$$F_{B1Y} = 3,91 \text{ N}$$

Recordem que el criteri de signes ve imposat segons la direcció de les forces en les figures.

Ara que sabem les forces en X i en Y a l'extrem de la biela 1, ens disposem a transformar-les a tangencial i axial. La component tangencial resultant serà la que s'oposarà al parell del motor.

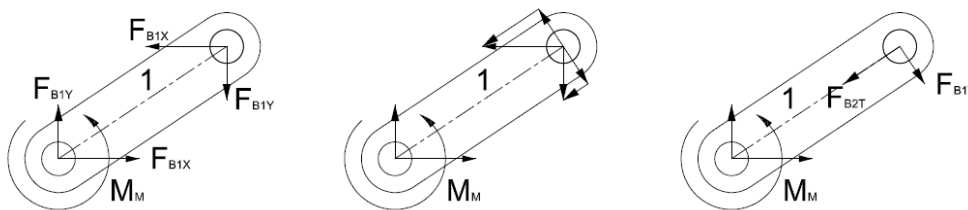


Fig. 25:  
Representació de les forces en una vista superior de la biela 1 i  
descomposició en tangencial i axial d'aquestes.



$$F_{B1T} = F_{B1Y} \cdot \cos(\alpha) - F_{B1X} \cdot \sin(\alpha) = 1,3 \text{ N} \quad (\text{Eq. 5})$$

Podríem calcular el valor de la axial però no ens seria d'utilitat.

Recordem per realitzar el càlcul del moment que la distància entre centres de la biela és de 40 mm.

$$M_M = F_{B1T} \cdot d \approx 52 \text{ N} \cdot \text{mm} = 0,53 \text{ Kg} \cdot \text{cm} \quad (\text{Eq. 6})$$

El motor haurà fer girar les dues meitats de la peça, per tant, s'ha de multiplicar el parell per 2. Sit això, hem calculat que haurà de tenir una parell mínim aproximat de 1.06 kg·cm. Tanmateix, per seguretat, ens interessaria que aquest valor fos més alt.

Utilitzarem el mateix servomotor que per la mà, ja que és poc pesat, i al ser iguals els podem alimentar al mateix voltatge.

### B.1.4. Càlcul de càrregues al canell

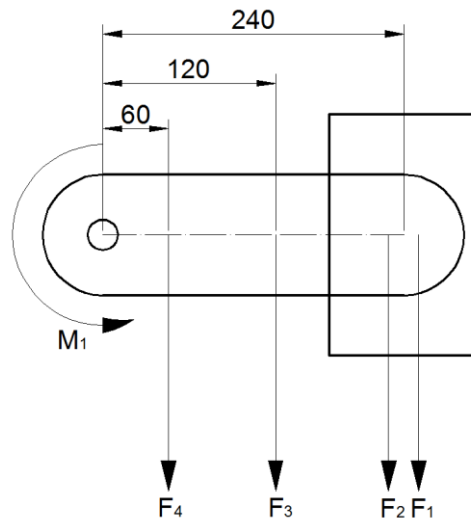


Fig. 26:  
Representació de les forces que ha de suportar  
el moment al canell

Al treballar amb acceleracions petites, suposarem per a tots als càlculs que les inèrcies són menyspreables, tan sols tindrem en compte el pes de l'objecte i l'acceleració d'aquest.

Tal i com hem indicat a les especificacions no aixecarem pesos superiors a 0.2 kg.

Pes a sostenir:

$$F_1 = m \cdot g = 1.96 \text{ N} \quad (\text{Eq. 7})$$

Força resultant d'accelerar el pes:

$$F_2 = m \cdot a = 0.04 \text{ N} \quad (\text{Eq. 8})$$

Pes propi de la pinça amb servomotor inclòs:

$$F_3 = m \cdot g = 2.94 \text{ N} \quad (\text{Eq. 9})$$

Pes del servomotor + "servo-block":

$$F_4 = m \cdot g = 0.98 \text{ N} \quad (\text{Eq. 10})$$

Parell servomotor:

$$M_1 = (F_1 + F_2) \cdot 240 + F_3 \cdot 120 + F_4 \cdot 60 = 891.6 \text{ N} \cdot \text{mm} \approx 9.1 \text{ kg} \cdot \text{cm} \quad (\text{Eq. 11})$$

### B.1.5. Càlcul de càrregues al colze

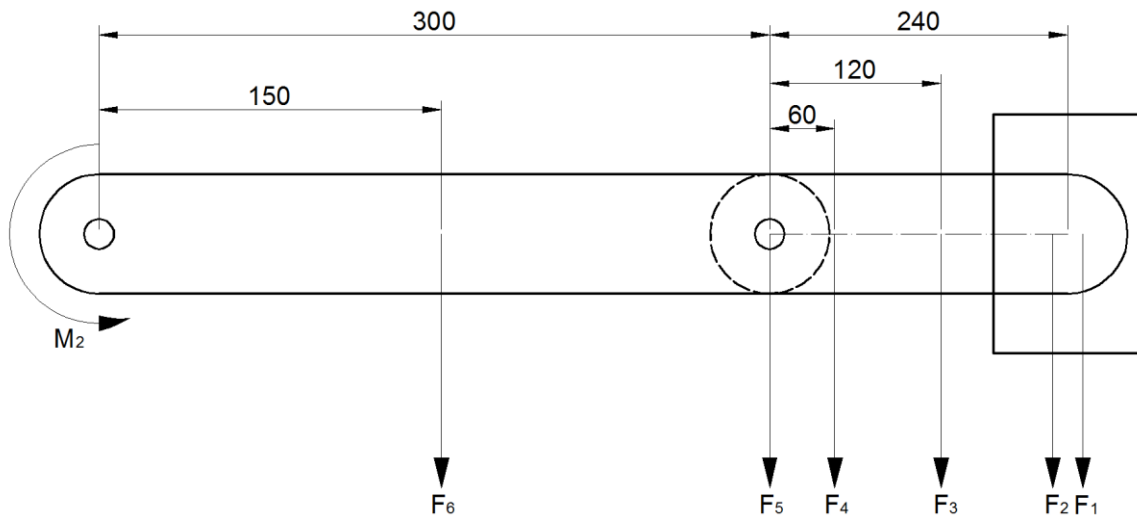


Fig. 26:

Representació de les forces que ha de suportar el moment al colze

Pes servomotor:

$$F_5 = m \cdot g = 0.589 \text{ N} \quad (\text{Eq. 12})$$

Pes propi del braç:

$$F_6 = m \cdot g = Est \cdot l \cdot g = 1.332 \text{ N} \quad (\text{Eq. 13})$$

Parell servomotor:

$$M_2 = (F_1 + F_2) \cdot (240 + 300) + F_3 \cdot (120 + 300) + F_4 \cdot (60 + 300) + F_5 \cdot 300 + F_6 \cdot 150 \quad (\text{Eq. 14})$$

$$M_2 = 3027.3 \text{ N} \cdot \text{mm} \approx 31 \text{ kg} \cdot \text{cm}$$

### B.1.6. Càlcul de càrregues a l'espatlla

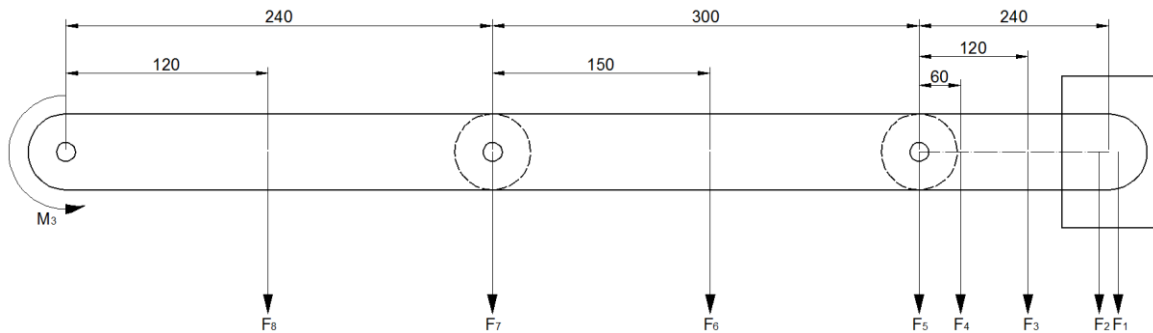


Fig. 27:

Representació de les forces que ha de suportar el moment a l'espatlla

Pes servomotor:

$$F_7 = m \cdot g = 0.589 \text{ N} \quad (\text{Eq. 15})$$

Pes propi del braç:

$$F_8 = m \cdot g = Est \cdot l \cdot g = 1.074 \text{ N} \quad (\text{Eq. 16})$$

Parell servomotor:

$$\begin{aligned} M_3 = & (F_1 + F_2) \cdot (240 + 300 + 240) + F_3 \cdot (120 + 300 + 240) \\ & + F_4 \cdot (60 + 300 + 240) + F_5 \cdot (300 + 240) + F_6 \cdot (150 + 240) \\ & + F_7 \cdot 240 + F_8 \cdot 120 \\ M_3 = & 5198.82 \text{ N} \cdot \text{mm} \approx 53 \text{ kg} \cdot \text{cm} \end{aligned} \quad (\text{Eq. 17})$$

## B.2. Càlcul de càrregues dels elements

Tenint en compte que el pes de la càrrega externa és tan petita, només farem el càlcul del factor de seguretat per a una càrrega estàtica considerant el cas més desfavorable.

Calcularem els punts més desfavorables, just al final del braç, i al centre de l'espalla, on les càrregues són més desfavorables.

### B.2.1. Càlcul a l'extrem del braç

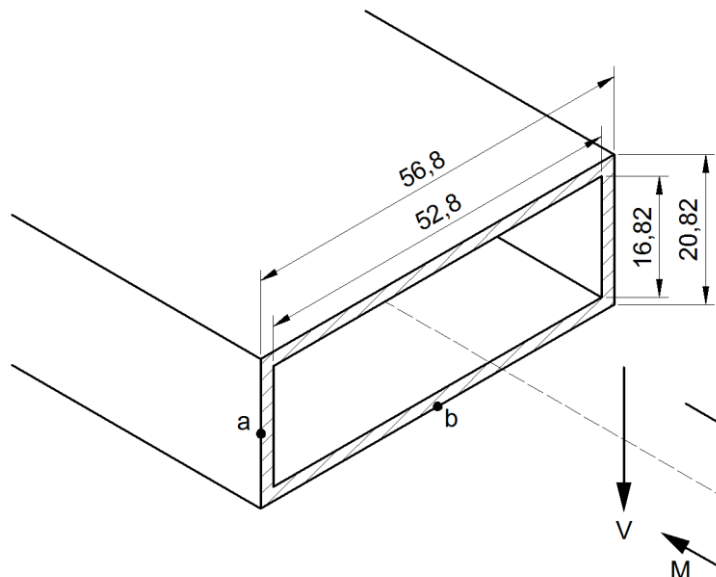


Fig. 28:  
Representació de les forces i moments que carreguen contra la secció del braç, a prop de l'espalla

Tallant:

$$V = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7 + F_8 \quad (\text{Eq. 18})$$

$$V = 9.52 \text{ N}$$

Moment:

$$\begin{aligned}
 M &= (F_1 + F_2) \cdot (240 + 300 + 240) + F_3 \cdot (120 + 300 + 240) \\
 &\quad + F_4 \cdot (60 + 300 + 240) + F_5 \cdot (300 + 240) + F_6 \cdot (150 + 240) \\
 &\quad + F_7 \cdot 240 + F_8 \cdot 120 \\
 M &= 5172.42 \text{ N} \cdot \text{mm} = 5.2 \text{ N} \cdot \text{m}
 \end{aligned}
 \tag{Eq. 19}$$

#### B.2.1.1. Càlcul del punt a

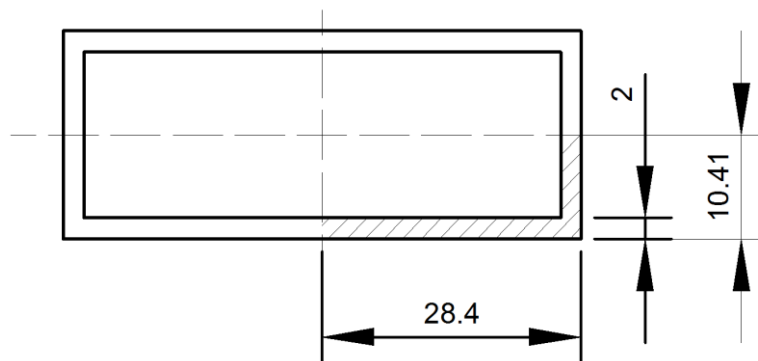


Fig. 29:  
Representació de l'Àrea utilitzada en el càlcul del moment estàtic

Moment estàtic:

$$M_e = A \cdot d_{cdgx} = 605.21 \text{ mm}^3 \tag{Eq. 20}$$

Inèrcia:

$$I_x = \Sigma \frac{b \cdot h^3}{12} = 21780.06 \text{ mm}^4 \tag{Eq. 21}$$

Esforç tallant:

On  $b = 2$

$$\tau(V) = \frac{V_y \cdot M_e}{b \cdot I_x} = 0.13 \text{ MPa} \tag{Eq. 22}$$

## B.2.1.2. Càlcul al punt b

Esforç a normal:

On  $y = 5$

$$\sigma(M_x) = \frac{M_x \cdot y}{I_x} = 2.49 \text{ MPa} \quad (\text{Eq. 23})$$

Càlcul del factor de seguretat:

Només calcularem el del punt b, ja que els esforços són molt més grans que al punt a.

La  $S_y$  de l'alumini 6061 és de 276 MPa.

$$n = \frac{S_y}{\sigma_A} = \frac{276}{2.49} = 111.3 \quad (\text{Eq. 24})$$

### B.2.2. Càlcul a l'espalla

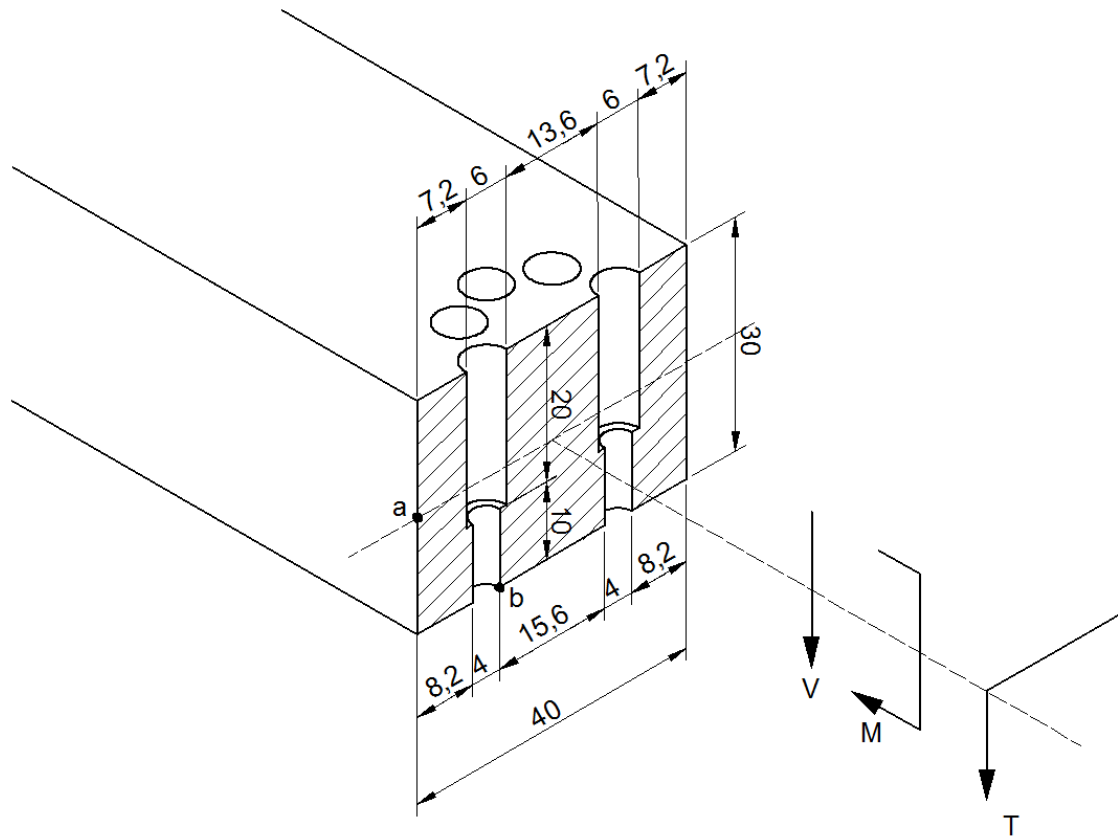


Fig. 30:  
Representació de les forces i moments que carreguen contra la secció de l'espalla,  
a la zona d'unió amb l'esquena

Forces Tallants:

$$V = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7 + F_8 = 0.4 \cdot 9.81 = 9.52 \text{ N} \quad (\text{Eq. 26})$$

Moment flector:

$$M = (F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7 + F_8) \cdot 80 \quad (\text{Eq. 27})$$

$$M = 761.6 \text{ N} \cdot \text{mm} = 0.76 \text{ N} \cdot \text{m}$$

Moment Torsor:

$$T = (F_1 + F_2) \cdot (240 + 300 + 240) + F_3 \cdot (120 + 300 + 240) \quad (\text{Eq. 28})$$

$$+ F_4 \cdot (60 + 300 + 240) + F_5 \cdot (300 + 240) + F_6 \cdot (150 + 240)$$

$$+ F_7 \cdot 240 + F_8 \cdot 120$$

$$T = 5172.42 \text{ N} \cdot \text{mm} = 5.2 \text{ N} \cdot \text{m}$$



## B.2.2.1. Càlcul del punt a:

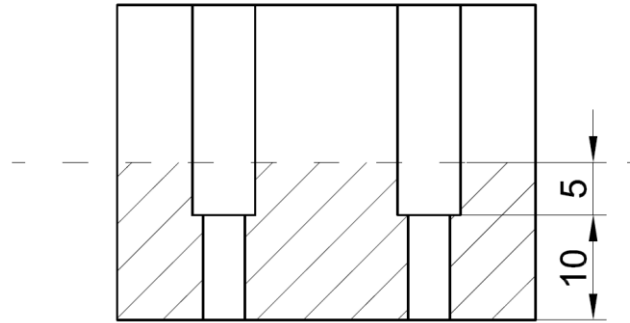


Fig. 31:  
Representació de l'Àrea utilitzada en el càlcul del moment estàtic

Moment estàtic:

$$M_e = A \cdot d_{cdgx} \approx 3150 \text{ mm}^3 \quad (\text{Eq. 29})$$

Inèrcia:

$$I_x = \Sigma \frac{b \cdot h^3}{12} = 13\,820.09 \text{ mm}^4 \quad (\text{Eq. 30})$$

Esforç tallant:

$$\tau(V) = \frac{V_y \cdot M_e}{b \cdot I_x} = 0.07 \text{ MPa} \quad (\text{Eq. 31})$$

Càlcul de l'esforç tallant:

On  $\mu$  es troba a les taules i val  $\mu = 0.22$ ,  $b$  és el costat llarg i val  $b = 40$  i  $e$  és el costat curt i val  $e = 30$

$$\tau(T) = \frac{T}{\mu \cdot b \cdot e^2} = 0.65 \text{ MPa} \quad (\text{Eq. 32})$$

$$\sigma_A = \tau(V) = 0.59 + 0.07 = 0.63 \text{ MPa} \quad (\text{Eq. 33})$$

Càlcul del factor de seguretat:

La  $S_y$  del filament PLA per a impressores 3D és de 35.9 MPa.

$$n = \frac{S_y}{\sigma_A} = \frac{35.9}{0.63} = 56.98 \quad (\text{Eq. 34})$$

## B.2.2.2. Càlcul del punt b:

Càlcul de l'esforç tallant:

$$\tau(T) = \frac{T}{\mu \cdot b \cdot e^2} = 0.65 \text{ MPa} \quad (\text{Eq. 35})$$

Càlcul de l'esforç normal:

$$\sigma(M_x) = \frac{M_x \cdot y}{I_x} = \frac{761.6 \cdot 15}{13\,820.09} = 0.83 \text{ MPa} \quad (\text{Eq. 36})$$

Hauríem d'aplicar concentració de tensions, però al assumir que el PLA és un material dúctil,  $K_f=1$ .

Càlcul del factor de seguretat:

$$\sigma_A = \sqrt{\sigma_x^2 + 3 \cdot \tau_{xy}^2} = 1.4 \text{ MPa} \quad (\text{Eq. 37})$$

La  $S_y$  del filament PLA per a impressores 3D és de 35.9 MPa.

$$n = \frac{S_y}{\sigma_A} = \frac{35.9}{1.4} = 30.19 \quad (\text{Eq. 38})$$

### B.3. Comprovació de la flexió

Quan tenim el braç estirat i perpendicular al terra, i amb la pinça carregada, és quan més probabilitats hi ha que els seus components flectin. Sempre suposant que els motors treballen de forma òptima i que, per tant, el braç no flectirà per les articulacions degut als motors. Tanmateix, degut a la poca llargada que té l'espatlla i tenint en compte que és una peça massissa, també suposarem que els únics components que flectiran, seran els perfils d'alumini.

Podríem suposar 2 casos.

On el braç el tractem com una única barra d'alumini enquestada des de l'espatlla o un cas on suposem 2 barres enquestades on la flexió de la primera s'acumula a la segona.

Tot i que el primer cas és més desfavorable, el segon és més acord a la realitat, per tant calcularem aquest.

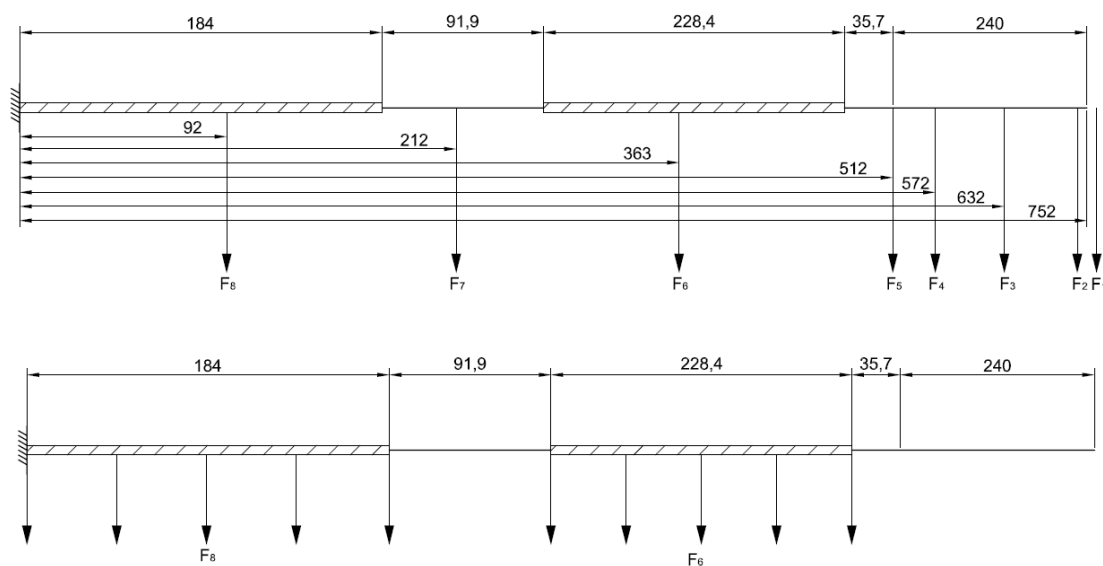


Fig. 32:  
Representació de les forces que s'apliquen a les diferents parts del braç.

A fi que els càlculs siguin més veraçs, aplicarem les càrregues 6 i 8, les quals equivalen als pesos dels perfils d'Alumini, com a càrregues distribuïdes.

Primer calcularem la Inèrcia de cada secció:



Fig. 33:  
Secció dels 2 perfils d'alumini pels qual estudiarem la flexió,

Ens referirem a la inèrcia B per referir-nos a la inèrcia del braç i la inèrcia A per refer-nos al avantbraç.

$$I_x = \Sigma \frac{b \cdot h^3}{12} \quad (\text{Eq. 39})$$

$$I_B = 21780,06 \text{ mm}^4$$

$$I_A = 28441.63 \text{ mm}^4$$

El mòdul estàtic de l'alumini 6061 és de 69 GPa.

Seguidament dibuixarem el diagrama de moments de l'estructura anterior.

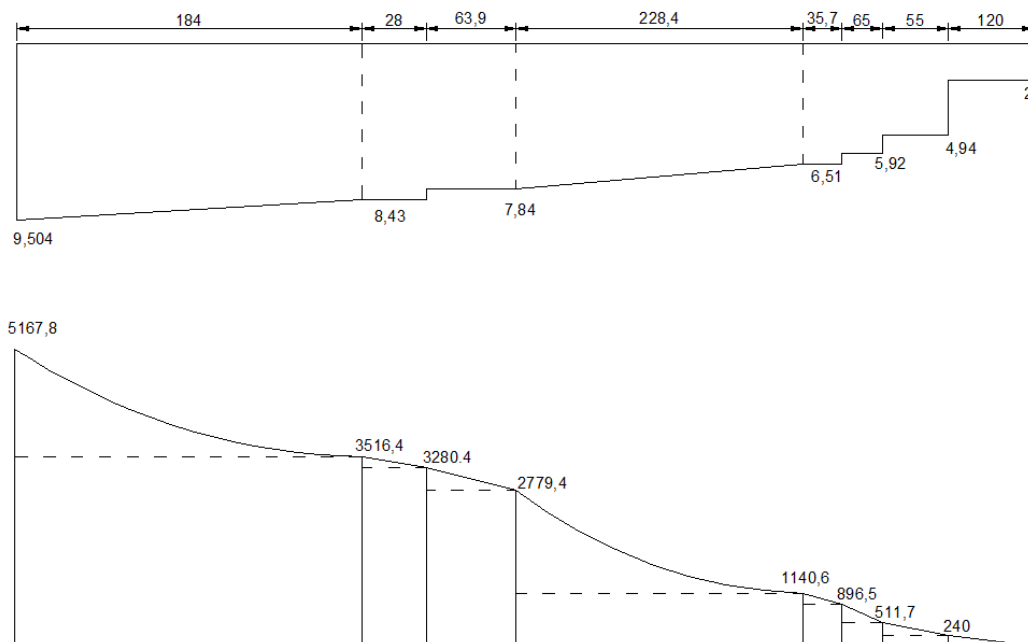


Fig. 34:  
Diagrama de tallants i de moments del braç anterior

Aplicarem les formules de Navier Bresse, analitzant cada tram per separat degut a que la longitud, la inèrcia i el mòdul de Young són diferents per a cada tram.

$$\theta_c = \theta_A - \frac{1}{E \cdot I} \cdot M(CA/A) \quad (Eq. 40)$$

$$\Delta_c = \Delta_A + \theta \cdot L_{AC} - \frac{1}{E \cdot I} \cdot M_{est}(CA/A) \quad (Eq. 41)$$

Tenint en compte la gran quantitat d'operacions que s'han de realitzar, utilitzarem un full de càlcul.

	Tram 1	Tram2	Tram3	Tram4	Tram5	Tram6	Tram7	Tram8
M	5167,811	3516,411	3280,371	2779,395	1140,625	896,5	511,7	240
E (N/mm2)	69000	1E+100	1E+100	69000	1E+100	1E+100	1E+100	1E+100
I (mm4)	21780	1E+100	1E+100	28441,6	1E+100	1E+100	1E+100	1E+100
L (mm)	184	28	63,9	228,4	35,7	65	55	120
$\theta$ (rad)	0,00050	0,00050	0,00050	0,00069	0,00069	0,00069	0,00069	0,00069
$\Delta$ (mm)	0,04891	0,06285	0,09467	0,23445	0,25923	0,30436	0,34255	0,42586

Taula 10:

Angle girat i fletxa al final de cada tram.

Tal i com podem veure, hem suposat unes inèrcies i uns mòduls elàstics, molt grans a fi d'acomplir la suposició que a les articulacions no flectarien.

Les dos últimes files ens mostren la fletxa i l'angle girat al final de cada tram, és a dir que cada columna, va sumant respecte el valor de l'anterior.

Finalment, la fletxa al final de la pinça serà de menys de mig mil·límetre. Podem donar el valor com a acceptable.

#### B.4. Modelització matemàtica del braç:

Tal i com s'ha explicat en l'apartat 3.1. *Funcionament del programa*, de forma continuada haurem d'aplicar cinemàtica directa a fi de trobar la posició X, Y i Z de la punta del final del braç en funció de  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ,  $\theta_4$  i  $\theta_5$ .

$\theta_1$  equival a l'angle de rotació de l'esquena,  $\theta_2$  equival a l'angle de rotació de l'espatlla,  $\theta_3$  equival a l'angle de rotació del colze,  $\theta_4$  equival a l'angle de rotació del canell i  $\theta_5$  equival a l'angle de rotació de la pinça.

Tanmateix, també haurem de ser capaços de trobar els valors de  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ,  $\theta_4$  aplicant cinemàtica inversa quant li proporcionem X, Y i Z.

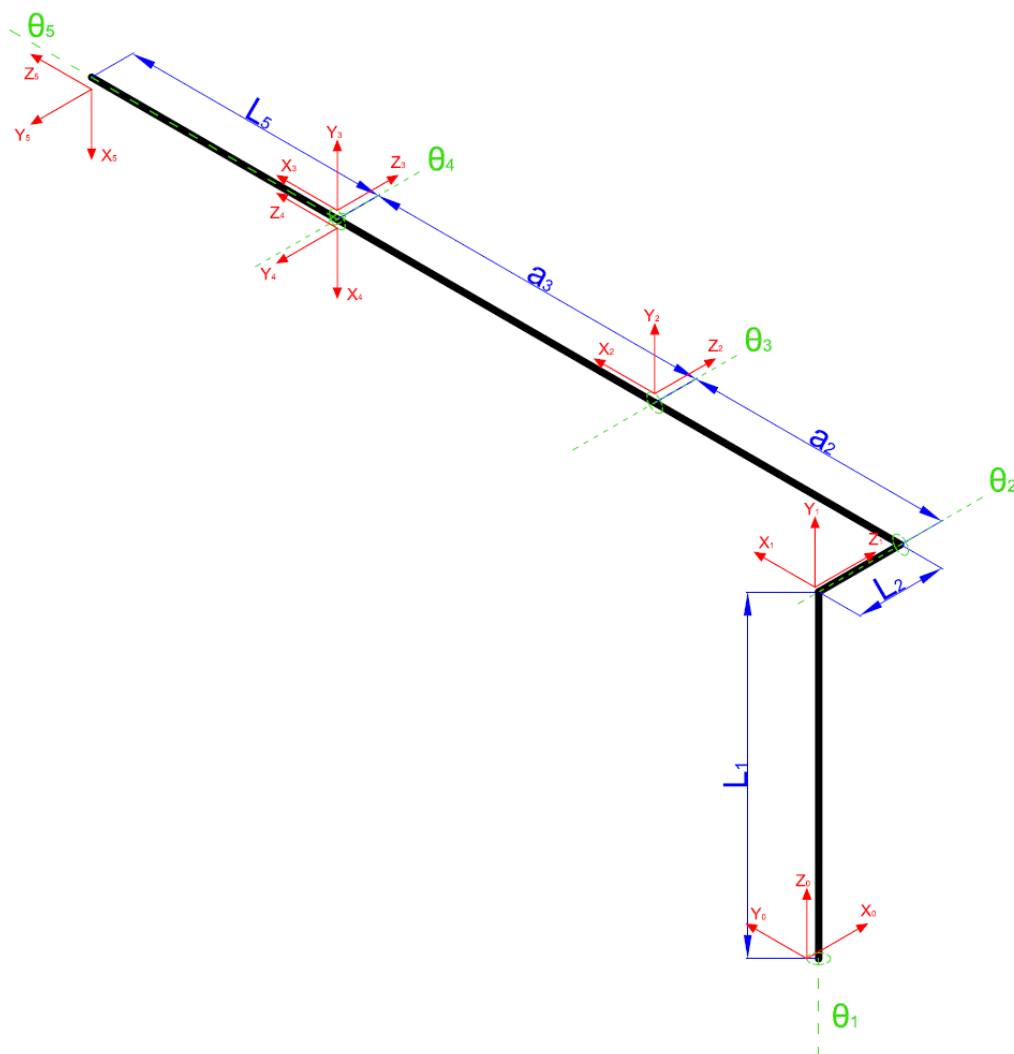


Fig. 35:  
Esquema del braç robòtic amb vectors

Per tal de poder aplicar la cinemàtica directa i la cinemàtica inversa, anteriorment esmentades, necessitem modelitzar tot el sistema a un model matemàtic.

Utilitzant matrius de rotació i matrius de translació podem transformar el sistema de la figura a una matriu en funció de totes les variables.

A la figura anterior podem veure un braç modelitzat amb simples vectors en una perspectiva isomètrica. Els eixos negres representarien cada eix físic del robot. De color blau hi ha acotades les mides amb les quals reconeixem cada dimensió del braç.

Amb color verd, els eixos de rotació de les articulacions.

Finalment, de color vermell hi ha representats els eixos de coordenades amb els quals podem parametritzar la orientació de cada eix.

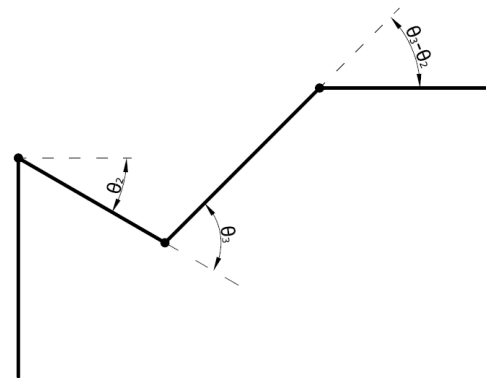


Fig. 36:  
Representació gràfica de la relació entre  $\theta_4$  i  $\theta_2$  i  $\theta_3$ .

Tal i com hem explicat anteriorment, només volem buscar les coordenades X, Y i Z de la punta del braç. L'orientació no ens és d'utilitat. Dit això, a l'hora de realitzar els càlculs no tindrem en compte  $\theta_5$  ja que només repercuteix en l'angle de rotació de la pinça.

L'angle de rotació del canell, el qual està vinculat a  $\theta_4$ , no està enllaçat a cap sensor, tal i com hem explicat anteriorment.

L'eix de treball de la pinça sempre ha de romandre paral·lel al pla del terra, per tant, tal i com podem veure a la figura 36, podem eliminar una variable, si suposem la següent igualtat:

$$\theta_4 = \theta_3 - \theta_2 \quad (Eq. 42)$$

## B.5. Càlcul matricial

L'última columna de la matriu és l'única que utilitzarem perquè és l'única que fa referència a la posició.

Per obtenir la matriu del sistema, hem d'escriure en sèrie totes les matrius de rotació i translació en ordre.

En acord amb la figura 29 anirem resseguint el moviment de cada origen de coordenades (els eixos de color vermell) seguint les regles de Denavit-Hartenberg.

Equacions del nus 0 al nus 1:

$$Rot(Z, \theta_1) = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 43)$$

$$Rot(X, 90^\circ) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(90^\circ) & -\sin(90^\circ) & 0 \\ 0 & \sin(90^\circ) & \cos(90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 44)$$

$$Trans(Y, L_1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 45)$$

I en acord a les equacions anteriors obtenim la següent:

$${}^0A_1 = Rot(Z, \theta_1) \cdot Rot(X, 90^\circ) \cdot Trans(Y, L_1) \quad (Eq. 46)$$

$${}^0A_1 = \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Equacions del nus 1 al nus 2:

$$Rot(Z, \theta_2) = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 47)$$

$$Trans(Z, L_2) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 48)$$

$$Trans(X, a_2) = \begin{pmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 49)$$

I en acord a les equacions anteriors obtenim la següent:

$${}^1A_2 = Rot(Z, \theta_2) \cdot Trans(Z, L_2) \cdot Trans(X, a_2) \quad (Eq. 50)$$

$${}^1A_2 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 \cdot \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \cdot \sin(\theta_2) \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Equacions del nus 2 al nus 3:

$$Rot(Z, \theta_3) = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 51)$$

$$Trans(X, a_3) = \begin{pmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 52)$$

I en acord a les equacions anteriors obtenim la següent:

$${}^2A_3 = Rot(Z, \theta_3) \cdot Trans(X, a_3) \quad (Eq. 53)$$

$${}^2A_3 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \cdot \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3 \cdot \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Equacions del nus 3 al nus 4:

$$Rot(Z, \theta_4) = \begin{pmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 54)$$

$$Rot(X, -90^\circ) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-90^\circ) & -\sin(-90^\circ) & 0 \\ 0 & \sin(-90^\circ) & \cos(-90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 55)$$

I en acord a les equacions anteriors obtenim la següent:

$${}^3A_4 = Rot(Z, \theta_4) \cdot Rot(X, -90^\circ) \quad (Eq. 56)$$

$${}^3A_4 = \begin{pmatrix} \cos(\theta_4) & 0 & -\sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & \cos(\theta_4) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Que segons l'equació 42 ens queda:

$${}^3A_4 = \begin{pmatrix} \cos(\theta_3 - \theta_2) & 0 & -\sin(\theta_3 - \theta_2) & 0 \\ \sin(\theta_3 - \theta_2) & 0 & \cos(\theta_3 - \theta_2) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 57)$$

Les equacions del nus 4 al nus 5, tal i com hem explicat anteriorment, les podem menysprear.

I en acord a les equacions anteriors obtenim la següent:

$${}^0A_4(\theta_1, \theta_2, \theta_3) = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \quad (Eq. 58)$$

$${}^0A_4(\theta_1, \theta_2, \theta_3) = \begin{pmatrix} & & X \\ R & & Y \\ & & Z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (Eq. 59)$$

De la matriu  ${}^0A_4$  només ens n'interessen les caselles X, Y i Z les quals resulten massa extenses per a ser plasmades al projecte, però tal i com hem explicat a l'apartat 3.1 *Funcionament del programa* el processador ha de realitzar aquest càlcul periòdicament i de forma continuada, per tant, la trobarem escrita dins els codi d'Arduino.

Seguidament, tal i com hem explicat a l'apartat 3 Descripció del programa, el programa, mitjançant cinemàtica inversa ha de ser capaç de trobar els valors de  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  quan li proporcionem els valors de X, Y i Z.

Per a realitzar aquesta tasca, hem utilitzat un "solver" com seria el de Maple, li hem introduït, les equacions de X, Y i Z obtingudes a l'apartat anterior i hem aïllat les equacions de cinemàtica inversa de  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  en funció de X, Y i Z.

Moltes d'aquestes equacions, tenen una base trigonomètrica, on ens podíem trobar que  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , tinguessin més d'un valor real.

Per evitar varis resultats per a les mateixes equacions, hem utilitzat el "solver" anteriorment esmentat, per a dos casos diferents, quan el braç és a la dreta i quan el braç és a l'esquerra.

Dit això, quan observem el codi, podrem trobar sis equacions. Tres d'aquestes per una direcció i tres més per a l'altre.

Tot i això, hem hagut de retallar els angles fins a certs valors:

$\theta_1$  pot treballar de  $0^\circ$  a  $180^\circ$ , on  $0^\circ$  seria mirant a mà esquerra,  $90^\circ$  estaria paral·lel al vidre frontal de la campana i  $180$  graus estaria mirant a mà dreta.

$\theta_2$  pot treballar de  $-90^\circ$  a  $90^\circ$ , on  $-90$  graus significaria posar el braç paral·lel al terra cap enrere (la qual cosa representa un moviment impossible per a la fisiologia humana) i  $90^\circ$  significaria posar el braç paral·lel al terra endavant (el qual seria un moviment quotidià) la qual cosa implica que no podríem inclinar el braç per sobre de tenir-lo paral·lel al terra, però tenint en compte l'entorn en el que treballem, tampoc és necessari.

$\theta_3$  treballarà de  $0^\circ$  a  $180^\circ$ , on  $0^\circ$  voldria dir tenir l'avantbraç paral·lel al braç i  $180^\circ$  seria tenir-lo igualment paral·lel però en sentit contrari (el qual també representa un moviment impossible per a la fisiologia humana). Això implicaria que no podríem doblegar el colze enrere, la qual cosa no ens suposa cap impediment real.

De la mateixa manera que amb l'apartat anterior, les equacions resultants, son massa extenses per a ser plasmades a paper, però les podrem trobar dins el codi de l'Arduino.

# ANNEX C: PROGRAMA

### C.1. Codi Comentat

```
//incloem una llibreria per a cada sensor
#include <MPU6050_tockn0.h>
#include <MPU6050_tockn1.h>
#include <MPU6050_tockn2.h>
#include <MPU6050_tockn3.h>
#include <Wire.h>
#include <VarSpeedServo.h>

//enllacem cada dispositiu amb les comandes que utilitza la seva
llibreria
MPU60500 mpu60500(Wire);
MPU60501 mpu60501(Wire);
MPU60502 mpu60502(Wire);
MPU60502 mpu60503(Wire);

//variable que utilitzarem per controlar cada quan executem el
programa
long timer = 0;
//variable que utilitzarem perquè els sensors només mostrin els seus
valors un cop per cicle
bool sensor = false;
//variable que utilitzarem perquè els càlculs només s'executin un cop
per cicle
bool calcul = false;

//adreça del multiplexor TCA9548A
#define TCAADDR 0x70

//subrutina per escollir de quin canal llegeix el multiplexor
void tcaselect(uint8_t i)
{
    if (i > 7) return;

    Wire.beginTransmission(TCAADDR);
    Wire.write(1 << i);
```

```
Wire.endTransmission();
}

//definim els servomotors
VarSpeedServo espatlla;
VarSpeedServo colze;
VarSpeedServo canell;
VarSpeedServo pinca;
VarSpeedServo esquena;
VarSpeedServo ma;

//definim les variables que utilitzarà la subrutina de càlcul
const float pi = 3.14159265359;

float theta1;
float theta2;
float theta3;
float theta5;

float X;
float Y;
float Z;

float R;
float R12;
float R1;
float R2;
float R3;
float R4;
long R5;

float S;
float T;

int L1 = 310;
int L2 = 80;
```

```
int a2 = 240;
int a3 = 300;
int L5 = 240;

float teta1;
float teta2;
float teta3;
float teta4;
float teta5;

float Axy;
float Az;

void setup()
{
    //enllacem cada servomotor a un PIN PWM de l'Arduino
    pinca.attach(3);
    canell .attach(5);
    colze.attach(6);
    espatlla.attach(9);
    esquena.attach(10);
    ma.attach(11);

    //declarem el pin 4 com a entrada
    pinMode(4, INPUT);

    //iniciem el port sèrie i l'I2C
    Serial.begin(9600);
    Wire.begin();

    //calibrem cada giroscopi cada cop que iniciem el programa
    tcselect(0);
    mpu60500.begin();
    mpu60500.calcGyroOffsets(true);
```



```
tcselect(1);
mpu60501.begin();
mpu60501.calcGyroOffsets(true);

tcselect(2);
mpu60502.begin();
mpu60502.calcGyroOffsets(true);

tcselect(3);
mpu60503.begin();
mpu60503.calcGyroOffsets(true);

//definim una posició inicial pel braç
esquena.write(0, 30, false);
espatlla.write(90, 30, false);
colze.write(90, 30, false);
canell.write(90, 30, false);
ma.write(0, 30, false);
pinca.write(90, 30, false);
}

void loop()
{
    //el programa executa contínuament la subrutina que llegeix els
    valors que ens arriben dels sensors
    adquirir_dades();

    //executem aqueta part del codi al inici de cada cicle
    if ((millis() - timer > 0) && (millis() - timer < 1000) && sensor
    == false)
    {
        //executa la subrutina que imprimeix els valors llegits
        anteriorment pel canal sèrie per ajudar a l'usuari a poder
        "debugar" amb més facilitat
        imprimir_sensors();
        //fem que la comanda anterior s'executi només 1 cop per cicle
    }
}
```

```
    sensor = true;
}
//executem aqueta part del codi quan hagi passat 1 segon
else if ((millis() - timer > 1000) && (millis() - timer < 4000) &&
calcul == false)
{
    //executa la subrutina que calcula els valors dels angles que
    s'han de transmetre a cada articulació
    calculs();
    //executa la subrutina que imprimeix els valors calculats a la
    subrutina void calculs() pel canal sèrie per ajudar a l'usuari a
    poder "debugar" amb més facilitat
    imprimir_valors();
    //fem que les comandes anteriors s'executin només 1 cop per cicle
    calcul = true;
    //executa la subrutina que transmet els angles adients als
    servomotors
    motors();
}
//executem aqueta part del codi quan hagin passat 4 segons
else if ((millis() - timer > 4000))
{
    //fem reset al comptador i a les variables booleanes a fi de poder
    tornar a executar el programa.
    timer = millis();
    sensor = false;
    calcul = false;
}
}

void adquirir_dades()
{
    tcselect(0);
    mpu60500.update();
    tcselect(1);
    mpu60501.update();
}
```

```
    tcaselect(2);
    mpu60502.update();
    tcaselect(3);
    mpu60503.update();
}

void motors()
{
    //cada servo executa l'angle que li pertoca. Li sumem un offset que
    depèn de la posició inicial del servo.
    int angle1 = teta1 + 0;
    int angle2 = teta2 + 90;
    int angle3 = teta3 + 90;
    int angle4 = teta4 + 90;
    int angle5 = teta5 + 0;

    esquena.write(angle1, 30, false);
    espatlla.write(angle2, 30, false);
    colze.write(angle3, 30, false);
    canell.write(angle4, 30, false);
    ma.write(angle5, 30, false);

    //obrim i tanquem la pinça en funció del polsador
    if (digitalRead(4) == HIGH)
    {
        pinca.write(90, 30, false);
    }
    else
    {
        pinca.write(135, 30, false);
    }
}

void imprimir_sensors()
{
    Serial.println(' ');
}
```

```
Serial.print("Sensor 0: "); Serial.println(mpu60500.getAngleX());
Serial.println(' ');
Serial.print("Sensor 1: "); Serial.println(mpu60501.getAngleX());
Serial.println(' ');
Serial.print("Sensor 2: "); Serial.println(mpu60502.getAngleX());
Serial.println(' ');
Serial.print("Sensor 3: "); Serial.println(mpu60503.getAngleX());
Serial.println(' ');
}

void calculs()
{
    //llegim els valors dels sensors i hi afegim un offset als angles
    perquè s'adeqüin a la direcció original del model matricial
    //theta1 és la lectura del sensor de l'esquena
    int theta1_offset = 90;
    theta1 = (mpu60500.getAngleX() + theta1_offset) * pi / 180;
    //theta2 és la lectura del sensor de l'espatlla
    int theta2_offset = -0;
    theta2 = (-mpu60501.getAngleX() + theta2_offset) * pi / 180;
    //theta3 és la lectura del sensor del colze
    int theta3_offset = -0;
    theta3 = (mpu60501.getAngleX() - mpu60502.getAngleX() +
    theta3_offset) * pi / 180;
    //theta5 és la lectura del sensor de la mà
    int theta5_sensivility = 90 / 80;
    theta5 = (mpu60503.getAngleX() * theta5_sensivility) * pi / 180;

    //calcules la posició X, Y i Z de la punta de la pinça
    X = +L2 * sin(theta1) - L5 * cos(theta3) * cos(-theta2 - theta3 -
    pi / 2) * cos(theta1) * sin(theta2) - L5 * cos(theta3) * sin(-theta2
    - theta3 - pi / 2) * cos(theta1) * cos(theta2) + a3 * cos(theta3)
    * cos(theta1) * cos(theta2) - L5 * sin(theta3) * cos(-theta2 -
    theta3 - pi / 2) * cos(theta1) * cos(theta2) + L5 * sin(theta3) *
    sin(-theta2 - theta3 - pi / 2) * cos(theta1) * sin(theta2) - a3 *
```

```

sin(theta3) * cos(theta1) * sin(theta2) + a2 * cos(theta1) *
cos(theta2);
Y = -L2 * cos(theta1) - L5 * cos(theta3) * cos(-theta2 - theta3 -
pi / 2) * sin(theta1) * sin(theta2) - L5 * cos(theta3) * sin(-theta2
- theta3 - pi / 2) * sin(theta1) * cos(theta2) + a3 * cos(theta3)
* sin(theta1) * cos(theta2) - L5 * sin(theta3) * cos(-theta2 -
theta3 - pi / 2) * sin(theta1) * cos(theta2) + L5 * sin(theta3) *
sin(-theta2 - theta3 - pi / 2) * sin(theta1) * sin(theta2) - a3 *
sin(theta3) * sin(theta1) * sin(theta2) + a2 * sin(theta1) *
cos(theta2);
Z = L5 * cos(theta2) * cos(theta3) * cos(-theta2 - theta3 - pi / 2)
- L5 * cos(theta2) * sin(-theta2 - theta3 - pi / 2) * sin(theta3)
+ a3 * cos(theta2) * sin(theta3) - L5 * cos(theta3) * sin(-theta2
- theta3 - pi / 2) * sin(theta2) + a3 * cos(theta3) * sin(theta2)
- L5 * cos(-theta2 - theta3 - pi / 2) * sin(theta2) * sin(theta3)
+ a2 * sin(theta2) + L1;

teta1 = theta1 * 180 / pi;
teta2 = theta2 * 180 / pi;
teta3 = theta3 * 180 / pi;

//si X és més gran que 350 establim que X = 350 i recalculem el
valor dels angles teta1, teta2 i teta 3
if (X > 350)
{
    X = 350;

    teta1 = atan(Y / 350) + asin(80 / sqrt( pow(350, 2) + pow(Y, 2)));
    teta1 = teta1 * 180 / pi;

    Axy = sqrt(pow(350, 2) + pow(Y, 2) - pow(L2, 2));
    Az = Z - 310;

    teta2 = atan((( -pow(Axy, 2) * Az - pow(Az, 3) + 480 * Axy * Az +
sqrt(-pow(Axy , 6) - 2 * pow(Axy , 4) * pow(Az , 2) - pow(Axy ,

```

```

2) * pow(Az , 4) + 1440 * pow(Axy , 5) + 1920 * pow(Axy , 3) *
pow(Az , 2) + 480 * Axy * pow(Az , 4) - 568800 * pow(Axy , 4) -
396000 * pow(Axy , 2) * pow(Az , 2) - 57600 * pow(Az , 4) -
6912000 * pow(Axy , 3) - 31104000 * Axy * pow(Az , 2) + 51204960000
* pow(Axy , 2) + 10368000000 * pow(Az , 2) - 11041920000000 * Axy
+ 727833600000000) - 25200 * Az) / (pow(Axy , 2) + pow(Az , 2) -
480 * Axy + 57600)) / ((pow(Axy , 2) + pow(Az , 2) + (-pow(Axy ,
2) * Az - pow(Az , 3) + 480 * Axy * Az + sqrt(-pow(Axy , 6) - 2
* pow(Axy , 4) * pow(Az , 2) - pow(Axy , 2) * pow(Az , 4) + 1440
* pow(Axy , 5) + 1920 * pow(Axy , 3) * pow(Az , 2) + 480 * Axy *
pow(Az , 4) - 568800 * pow(Axy , 4) - 396000 * pow(Axy , 2) *
pow(Az , 2) - 57600 * pow(Az , 4) - 6912000 * pow(Axy , 3) -
31104000 * Axy * pow(Az , 2) + 51204960000 * pow(Axy , 2) +
10368000000 * pow(Az , 2) - 11041920000000 * Axy +
727833600000000) - 25200 * Az) * Az / (pow(Axy , 2) + pow(Az , 2)
- 480 * Axy + 57600) - 480 * Axy + 25200) / (Axy - 240)));
//      teta2 = teta2 * 180 / pi;

if (teta2 < 0)
{
    teta2 = -teta2 * 180 / pi - 180;
}
else
{
    teta2 = -teta2 * 180 / pi;
}

R12 = (pow(Axy, 2) + pow(Az, 2) - 480 * Axy - 720000 / 8);
R1 = R12 / 144000;
R2 = (pow(Axy, 2) * Az + pow(Az, 3) - 480 * Axy * Az + 25200 *
Az);
R3 = ((144000 * Axy - 34560000) * (pow(Axy, 2) + pow(Az, 2) - 480
* Axy + 57600));
R4 = -pow(Axy, 6) - 2 * pow(Axy, 4) * pow(Az, 2) - pow(Axy, 2) *
pow(Az, 4) + 1440 * pow(Axy, 5) + 1920 * pow(Axy, 3) * pow(Az, 2)
+ 480 * Axy * pow(Az, 4) - 568800 * pow(Axy, 4) - 396000 * pow(Axy,

```

```

2) * pow(Az, 2) - 57600 * pow(Az, 4) - 6912000 * pow(Axy, 3) -
31104000 * Axy * pow(Az, 2) + 51204960000 * pow(Axy, 2) +
10368000000 * pow(Az, 2) - 11041920000000 * Axy + 7278336000000000;
R5 = 480 * pow(Axy, 2) + 480 * pow(Az, 2) - 230400 * Axy +
27648000;

teta3 = atan(((1 / 480.0) * (R5) * (-pow(Axy, 2) * Az - pow(Az,
3) + 480 * Axy * Az + sqrt(R4) - 25200.0 * Az) / R3 + R2 /
(144000.0 * Axy - 34560000.0)) / R1);

if (teta3 < 0)
{
    teta3 = teta3 * 180 / pi + 180;
}
else
{
    teta3 = teta3 * 180 / pi;
}
}

//si X és més petita que -350 establim que X = -350 i re-calcularem
el valor dels angles teta1, teta2 i teta 3
else if (X < -350)
{
    X = -350;
    teta1 = atan(350 / Y) + asin(80 / (sqrt(pow(Y , 2) + pow(350 ,
2))));
    teta1 = teta1 * 180 / pi + 90;

    Axy = sqrt(pow(350 , 2) + pow(Y , 2) - pow(L2 , 2));
    Az = Z - 310;

    teta2 = atan((-pow(Axy, 2) * Az - pow(Az, 3) + 480 * Axy * Az +
sqrt(-pow(Axy , 6) - 2 * pow(Axy , 4) * pow(Az , 2) - pow(Axy ,
2) * pow(Az , 4) + 1440 * pow(Axy , 5) + 1920 * pow(Axy , 3) *
pow(Az , 2) + 480 * Axy * pow(Az , 4) - 568800 * pow(Axy , 4) -

```

```

396000 * pow(Axy , 2) * pow(Az , 2) - 57600 * pow(Az , 4) -
6912000 * pow(Axy , 3) - 31104000 * Axy * pow(Az , 2) + 51204960000
* pow(Axy , 2) + 10368000000 * pow(Az , 2) - 11041920000000 * Axy
+ 727833600000000) - 25200 * Az) / (pow(Axy , 2) + pow(Az , 2) -
480 * Axy + 57600)) / ((pow(Axy , 2) + pow(Az , 2) + (-pow(Axy ,
2) * Az - pow(Az , 3) + 480 * Axy * Az + sqrt(-pow(Axy , 6) - 2
* pow(Axy , 4) * pow(Az , 2) - pow(Axy , 2) * pow(Az , 4) + 1440
* pow(Axy , 5) + 1920 * pow(Axy , 3) * pow(Az , 2) + 480 * Axy *
pow(Az , 4) - 568800 * pow(Axy , 4) - 396000 * pow(Axy , 2) *
pow(Az , 2) - 57600 * pow(Az , 4) - 6912000 * pow(Axy , 3) -
31104000 * Axy * pow(Az , 2) + 51204960000 * pow(Axy , 2) +
10368000000 * pow(Az , 2) - 11041920000000 * Axy +
727833600000000) - 25200 * Az) * Az / (pow(Axy , 2) + pow(Az , 2)
- 480 * Axy + 57600) - 480 * Axy + 25200) / (Axy - 240)));

```

```

if (teta2 < 0)

```

```

{
    teta2 = -teta2 * 180 / pi - 180;
}

```

```

else

```

```

{
    teta2 = -teta2 * 180 / pi;
}

```

```

R12 = (pow(Axy, 2) + pow(Az, 2) - 480 * Axy - 720000 / 8);

```

```

R1 = R12 / 144000;

```

```

R2 = (pow(Axy, 2) * Az + pow(Az, 3) - 480 * Axy * Az + 25200 *
Az);

```

```

R3 = ((144000 * Axy - 34560000) * (pow(Axy, 2) + pow(Az, 2) - 480
* Axy + 57600));

```

```

R4 = -pow(Axy, 6) - 2 * pow(Axy, 4) * pow(Az, 2) - pow(Axy, 2) *
pow(Az, 4) + 1440 * pow(Axy, 5) + 1920 * pow(Axy, 3) * pow(Az, 2)
+ 480 * Axy * pow(Az, 4) - 568800 * pow(Axy, 4) - 396000 * pow(Axy,
2) * pow(Az, 2) - 57600 * pow(Az, 4) - 6912000 * pow(Axy, 3) -
31104000 * Axy * pow(Az, 2) + 51204960000 * pow(Axy, 2) +
10368000000 * pow(Az, 2) - 11041920000000 * Axy + 727833600000000;

```



```

R5 = 480 * pow(Axy, 2) + 480 * pow(Az, 2) - 230400 * Axy +
27648000;
teta3 = atan(((1 / 480.0) * (R5) * (-pow(Axy, 2) * Az - pow(Az,
3) + 480 * Axy * Az + sqrt(R4) - 25200.0 * Az) / R3 + R2 /
(144000.0 * Axy - 34560000.0)) / R1);

if (teta3 < 0)
{
    teta3 = teta3 * 180 / pi + 180;
}
else
{
    teta3 = teta3 * 180 / pi;
}

}
else
{
    teta1 = theta1;
    teta2 = theta2;
    teta3 = theta3;
    teta1 = teta1 / pi * 180;
    teta2 = teta2 / pi * 180;
    teta3 = teta3 / pi * 180;
}

teta1 = teta1 / 180 * pi;
teta2 = teta2 / 180 * pi;
teta3 = teta3 / 180 * pi;

R = +L2 * sin(teta1) - L5 * cos(teta3) * cos(-teta2 - teta3 - pi /
2) * cos(teta1) * sin(teta2) - L5 * cos(teta3) * sin(-teta2 - teta3
- pi / 2) * cos(teta1) * cos(teta2) + a3 * cos(teta3) * cos(teta1)
* cos(teta2) - L5 * sin(teta3) * cos(-teta2 - teta3 - pi / 2) *
cos(teta1) * cos(teta2) + L5 * sin(teta3) * sin(-teta2 - teta3 - pi

```

```

/ 2) * cos(teta1) * sin(teta2) - a3 * sin(teta3) * cos(teta1) *
sin(teta2) + a2 * cos(teta1) * cos(teta2);

S = -L2 * cos(teta1) - L5 * cos(teta3) * cos(-teta2 - teta3 - pi
/ 2) * sin(teta1) * sin(teta2) - L5 * cos(teta3) * sin(-teta2 -
teta3 - pi / 2) * sin(teta1) * cos(teta2) + a3 * cos(teta3) *
sin(teta1) * cos(teta2) - L5 * sin(teta3) * cos(-teta2 - teta3 - pi
/ 2) * sin(teta1) * cos(teta2) + L5 * sin(teta3) * sin(-teta2 -
teta3 - pi / 2) * sin(teta1) * sin(teta2) - a3 * sin(teta3) *
sin(teta1) * sin(teta2) + a2 * sin(teta1) * cos(teta2);

T = L5 * cos(teta2) * cos(teta3) * cos(-teta2 - teta3 - pi / 2)
- L5 * cos(teta2) * sin(-teta2 - teta3 - pi / 2) * sin(teta3) + a3
* cos(teta2) * sin(teta3) - L5 * cos(teta3) * sin(-teta2 - teta3 -
pi / 2) * sin(teta2) + a3 * cos(teta3) * sin(teta2) - L5 * cos(-
teta2 - teta3 - pi / 2) * sin(teta2) * sin(teta3) + a2 * sin(teta2)
+ L1;

//finalment obtenim els valors de tots els angles on teta1 equival
al servo de l'esquena, teta2 equival al servo de l'espatlla, teta3
equival al servo del colze, teta4 equival al servo del canell i
teta5 equival al servo de la rotació de la mà

teta1 = teta1 / pi * 180;
teta2 = teta2 / pi * 180;
teta3 = teta3 / pi * 180;
teta4 = (teta2 + teta3);
teta5 = theta5;

theta1 = teta1 * 180 / pi;
theta2 = teta2 * 180 / pi;
theta3 = teta3 * 180 / pi;
}

void imprimir_valors()
{

```

```
//X, Y i Z són els valors de la posició de la punta de la pinça  
calculats a partir dels valors dels angles directament llegits dels  
sensors  
Serial.print("X: ");  
Serial.println(X);  
Serial.print("Y: ");  
Serial.println(Y);  
Serial.print("Z: ");  
Serial.println(Z);  
//R, S i T són els valors de la posició de la punta de la pinça  
calculats a partir dels valors dels angles si aquests han hagut de  
ser re-calculats  
Serial.print("R: ");  
Serial.println(R);  
Serial.print("S: ");  
Serial.println(S);  
Serial.print("T: ");  
Serial.println(T);  
  
Serial.println(" ");  
  
Serial.print("Axy: ");  
Serial.println(Axy);  
Serial.print("Az: ");  
Serial.println(Az);  
  
Serial.println(" ");  
Serial.print("R1: ");  
Serial.println(R1);  
Serial.print("R2: ");  
Serial.println(R2);  
Serial.print("R3: ");  
Serial.println(R3);  
Serial.print("R4: ");  
Serial.println(R4);  
Serial.print("R5: ");
```

```
Serial.println(R5);
Serial.println(" ");
//theta són els valors dels angles de cada una de les articulacions
llegida directament dels sensors
Serial.print("theta1: ");
Serial.println(theta1);
Serial.print("theta2: ");
Serial.println(theta2);
Serial.print("theta3: ");
Serial.println(theta3);
//teta són els valors dels angles de cada una de les articulacions
després de ser re-calculades
Serial.print("teta1: ");
Serial.println(teta1);
Serial.print("teta2: ");
Serial.println(teta2);
Serial.print("teta3: ");
Serial.println(teta3);
Serial.print("teta4: ");
Serial.println(teta4);
Serial.print("teta5: ");
Serial.println(teta5);
Serial.println(' ');
}
```

# ANNEX D: SIMULACIÓ INFORMÀTICA DEL BRAÇ

### D.1. Simulació del braç amb Matlab i Robotics Toolbox

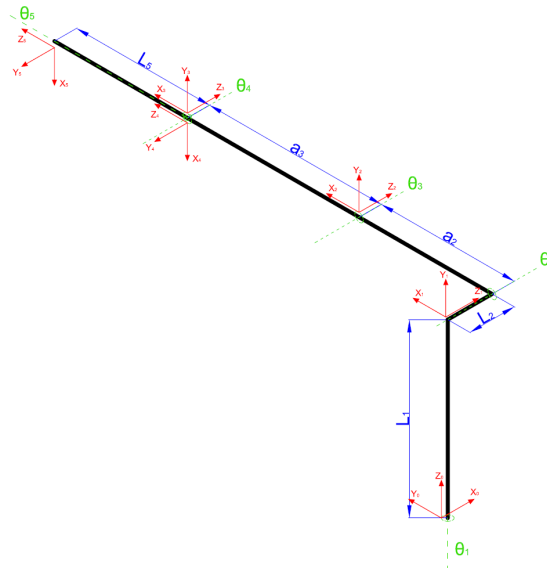


Fig. 37: Esquema del braç robòtic amb vectors

Tal i com hem vist a l'apartat 3 Descripció del programa, l'Arduino haurà de calcular, mitjançant cinemàtica directa, la posició de la punta del braç, i si s'escau, mitjançant cinemàtica inversa, tornar a calcular els valors dels angles resultants.

Per a realitzar aquest segon càlcul, habitualment, utilitzaríem un "solver", malauradament, no existeixen llibreries per a Arduino capaces de realitzar operacions amb aquest nivell de complexitat.

	$\theta$	$d$	$a$	$\alpha$	$H$
1	$q_1$	$L_1$	0	$90^\circ$	$90^\circ$
2	$q_2$	$L_2$	$a_2$	0	0
3	$q_3$	0	$a_3$	0	0
4	$q_4$	0	0	$-90^\circ$	$-90^\circ$
5	$q_5$	$L_5$	0	0	0

Taula 11:

Taula de Denavit Hartenberg del braç robòtic

Tal i com podem veure a l'annex A.2. Modelització matemàtica del braç, hem hagut de buscar les equacions de cinemàtica inversa, ja aïllades, a fi que l'Arduino les pugui processar correctament.

La simulació realitzada amb Matlab, està realitzada amb l'ajuda de la "Robotics Toolbox" la qual és fàcil d'utilitzar però alhora et permet assolir alts nivells de complexitat si és requerit. Per a poder utilitzar aquesta eina es requereix proporcionar-li al Matlab la taula de Denavit Hartenberg corresponent al braç robòtic.

Nosaltres a la simulació li hem subministrat uns valors a l'atzar els quals representarien els valors llegits pels hipotètics giroscopis. El Matlab ha calculat la posició de la punta del braç i en el cas que sortís fora de la suposada campana, torna a calcular els valors dels angles, imposant un nou valor per a la variable X.

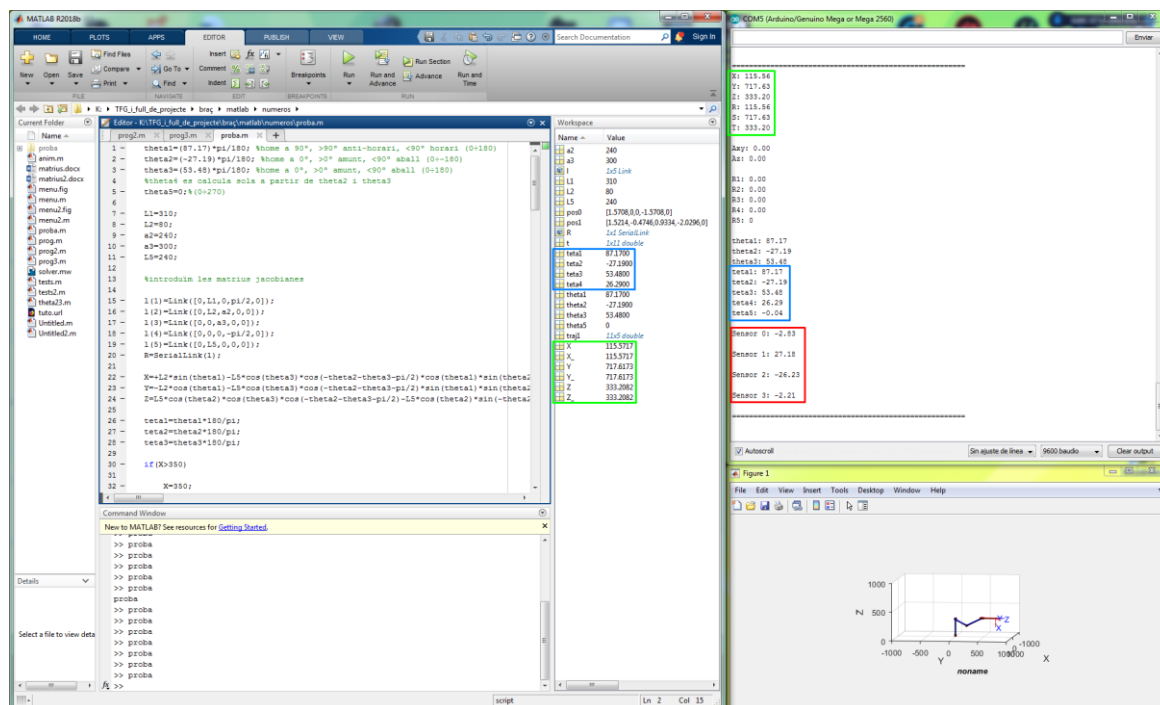


Fig. 38:  
Simulació en paral·lel entre Matlab i Arduino sense tornar a calcular els angles

El Matlab, per a fer aquest segon càlcul dels nous valors dels angles, ho ha fet de dos maneres, utilitzant el "solver" intern de Matlab i utilitzant les equacions de la cinemàtica inversa a fi de poder comparar si els resultats obtingut per ambdós mètodes és el mateix. Si així fos, voldria dir que l'Arduino és capaç de realitzar els càlculs sense equivocar-se, ja que podria obtenir el mateix resultat que amb un "solver" sense disposar d'aquest.

A la figura 32 podem veure una simulació en paral·lel entre el Matlab i l'Arduino on els valors "theta" són els que han llegit els sensors i "teta" són els valors després de tornar a calcular (el mateix valor en cas que no necessitem segon càlcul) i podem veure que com que el valor de X és dins del rang de  $\pm 350$  no hem necessitat segon càlcul per tant, els valors de les "thetes" i les "tetes" coincideixen.

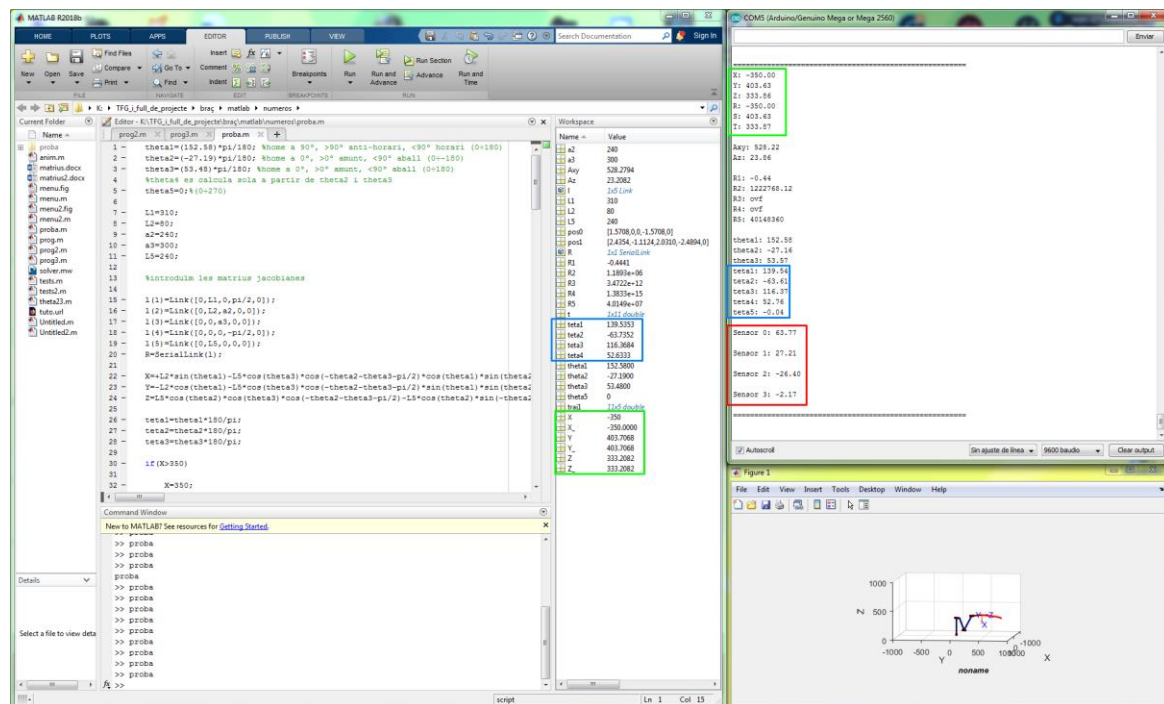


Fig. 39: Simulació en paral·lel entre Matlab i Arduino tornant a calcular els angles

En canvi la figura 33 podem veure que com que X en una primera instància estaria fora del rang de  $\pm 350$ , per tant, se li adjudica aquest valor i en fa el 2n càlcul dels angles, per això podem veure que els valors de "theta" i "teta" no coincideixen.

Observant aquests dos casos podem coincidir en què el programa funciona correctament.