Universitat de Girona
Escola Politècnica Superior

Treball final de grau

Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol:

Disseny d'un braç robòtic controlat per un guant electrònic

Document 1: Memòria i Annexos

Alumne: Josep Rueda Collell

Tutor: Miquel Rustullet Reñe
Departament: Enginyeria Elèctrica, Electrònica i Automàtica
Àrea: Enginyeria de Sistemes i Automàtica

Convocatòria: Juny/2019

INDEX:

1. Introducció:

1.1 Antecedents

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

Fig. 1: Logotip del Duve Institute

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.

El Sr. Marcel McCullough, científic dels laboratoris SCBD (Stem Cells Biology and Development) del Duve Institute ens explica que, 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 projecte

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 juntament amb el programa que el controli.

1.3 Requeriments i abast

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 velocitat superior a 0,4 rad/s i cap articulació tindrà una acceleració superior a 0,2 rad/s^2.

La feina del projectista acaba un cop dissenyada la solució del projecte i s'entrega el projecte al client.

2. Conjunt braç

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

Per 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, hem realitzat un estudi per veure quina és la forma més optima per fer-lo servir. Per a fer aquest estudi, hem utilitzat les dimensions per a la campana descrites als requeriments del projecte.

Fig. 3: 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.

En el cas on tenim el braç perpendicular al vidre de la campana, estaríem obligats a fer el tronc o esquena més llarg, a fi de poder aprofitar millor l'espai a dins de la campana.

La figura 4 ens presenta una vista lateral de la campana amb varies posicions dels tres

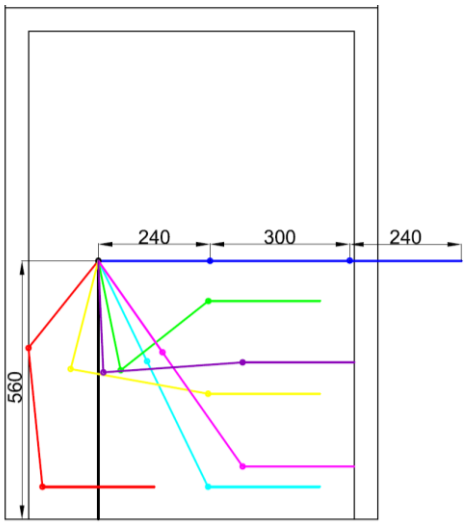


Fig. 4: Vista lateral de varies posicions del braç

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. En el cas de la distribució on tenim el braç paral·lel al vidre de la campana, el tronc o esquena podria ser més curt, el qual ens permetria més llibertat de moviment.

La figura 5 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.

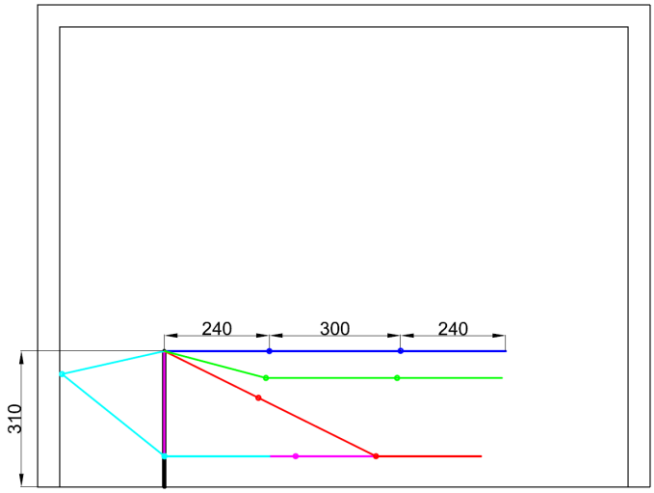


Fig. 5: Vista frontal de varies posicions del braç

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

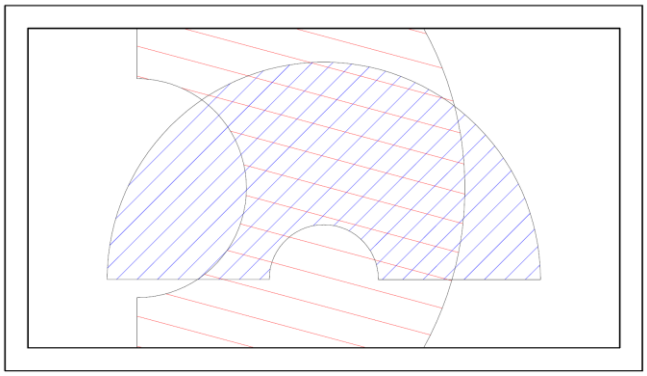


Fig. 6: Vista superior comparativa entre rangs

d'aquesta.

En definitiva, tal i com podem veure a la figura 6 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 del braç paral·lel al vidre és que tan si treballa 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 en 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 per aquest 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 Braç

Tenint en compte l'estudi de forma de l'apartat 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

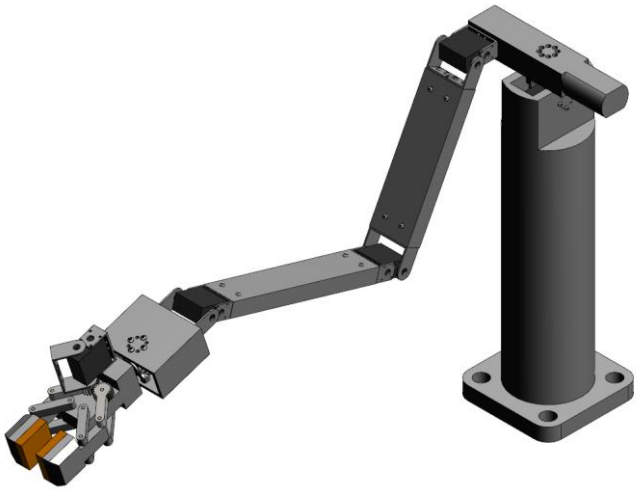


Fig. 7: Visualització de la solució proposada

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

Fig. 8: Servo-block

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 degut a la seva complexa geometria l'haurem d'imprimir mitjançant una impressora 3D i conseqüentment estarà composta de PLA per a impressora 3D

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" si fos necessari.

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 penjaria 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 A.2 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 i conseqüentment hem escollit els servomotors d'acord a 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.

Voltatge:	5 - 6 V
Parell:	13,5 kg·cm (6 V)
	31 kg·cm (6,8 V)
Pes	60 g
Mida	40 × 20 × 40,5

Fig. 9: Servo-motor HS-785HB

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* el qual està preparat per al nostre "servo-block". Aquest servomotor pot arribar a subministrar un parell de 13.2 kg·cm.

Voltatje:	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

Fig. 11: Servo-motor RDS3135

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*, el qual pot subministrar un parell de 60 kg·cm.

El "servomotor colze" estaria situat al colze. Hem escollit el RDS3135 de la casa *Robot Digital Servos*, 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 l'RDS3128 de la casa *Robot Digital Servos*, 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.

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

Fig. 12: Servo-motor RDS3218

La peça base braç tindrà allotjat dins seu 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.

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 necessitarem allocar-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.

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".

La peça espatlla i la peça base braç són peces de fabricació les quals 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. 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 les 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 "base mà" sobre la qual hi construiríem els mecanismes de la pròpia pinça.

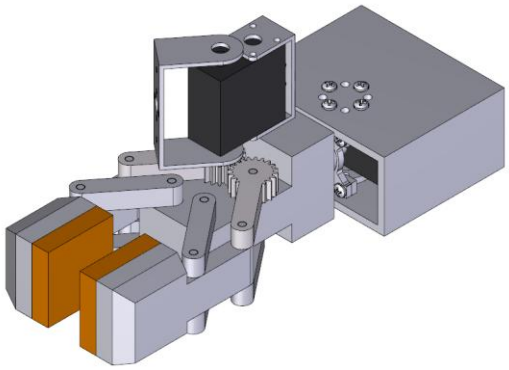


Fig. 13: Visualització 3D de la mà o pinça

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.

Una biela de cada quadrat articulat estarà dentada, a mode d'engranatge, 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 "base mà" i s'utilitzarà per accionar el mecanisme de la pinça.

2.3.1 Descripció dels 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", l'HS-785HB de la casa *Hitec*, 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 l'RDS3135 de la casa *Robot Digital Servos*, el qual pot subministrar un parell de fins a 20 kg·cm.

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 "base 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 complert.

Hem escollit un nombre de dents imparell a fi que els 2 engranatges puguin encaixar entre ells sense quedin desfasats mig pas. El centre instantani de rotació entre les dos bieles dentades no és estàtic com seria el cas d'una xarnera, sinó que es desplaça segons la posició de les bieles. Tenint en compte això, hem col·locat l'eix de rotació del servomotor sobre l'eix de rotació d'un dels engranatges.

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 Guant

El guant 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 guant i els sensors al lloc adient sobre el braç humà.

Els sensors estaran adherits mitjançant adhesiu a sobre les abraçadores 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 polsador a la punta del polze per a accionar la pinça.

2.4.1 Descripció dels components del guant

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^2C al processador, en aquest cas, un Arduino Uno.

El protocol I^2C te una velocitat per enviar i rebre dades de100 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^2C 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 del 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^2C.

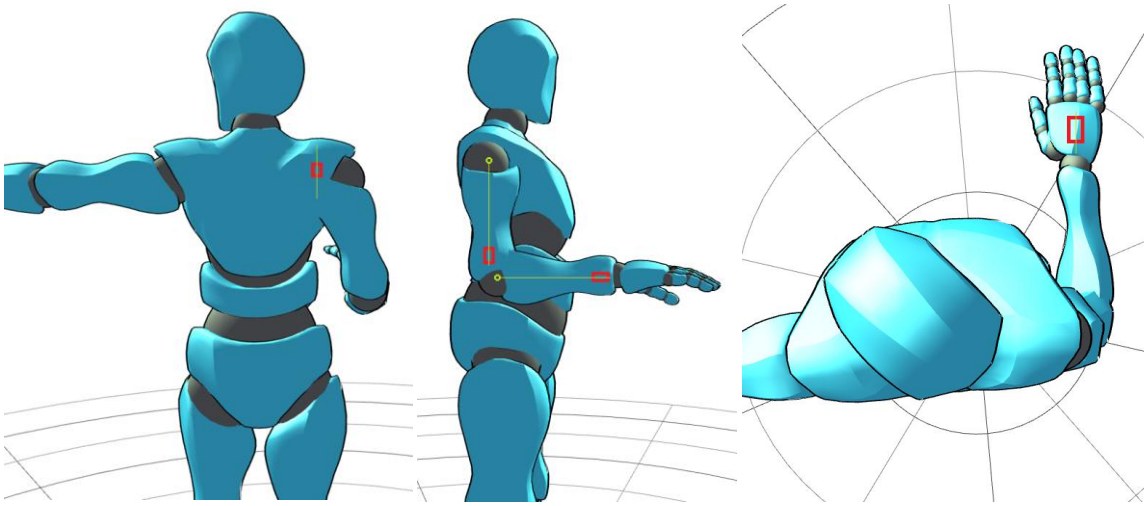


Fig. 14: Vista anterior, lateral i superior, amb la localització dels giroscopis

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". Aquesta placa és cada cop més utilitzada per a aplicacions de caràcter professional degut a la facilitat d'us 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.

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

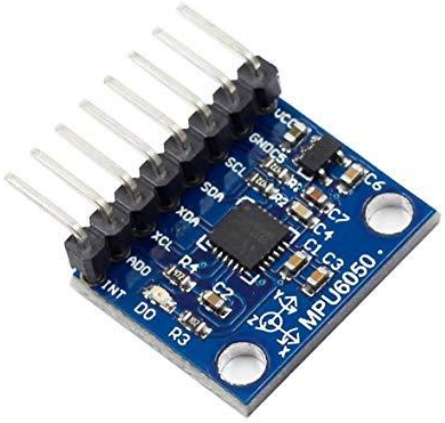
Fig. 15: Arduino Uno

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^2C. 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.



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

Fig. 16: MPU6050

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.

VCC	1,65 V - 5,5 V

Fig. 17: TCA9548A

Dit això ens veiem obligats a utilitzar un multiplexor I^2C, 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.

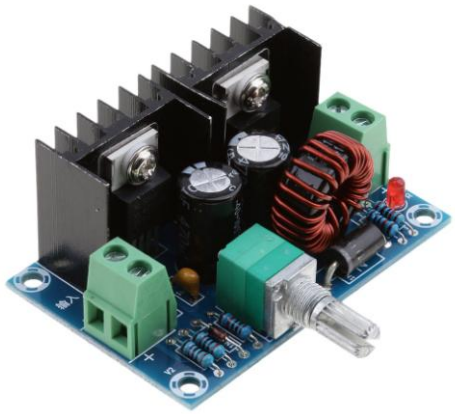
La intensitat consumida pels sensors i el multiplexor són de l'ordre de µA. Tenint en compte que són tan baixos, els menysprearem.

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.



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

Fig. 18: XH-M401

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 correcte 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.

$$P = V \cdot I \qquad \text{(Eq. 1)}$$

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.

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.

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

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

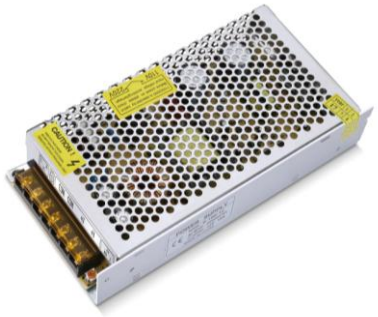
Utilitzant la següent fórmula, podem calcular la potència que ha de subministrar la font:

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.

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

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 instalació tenint en compte que hem de conectar 4 convertidors al darrere.



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

Fig. 19: GHB 12V

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 directe. 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 ANEX 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 polsador 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.

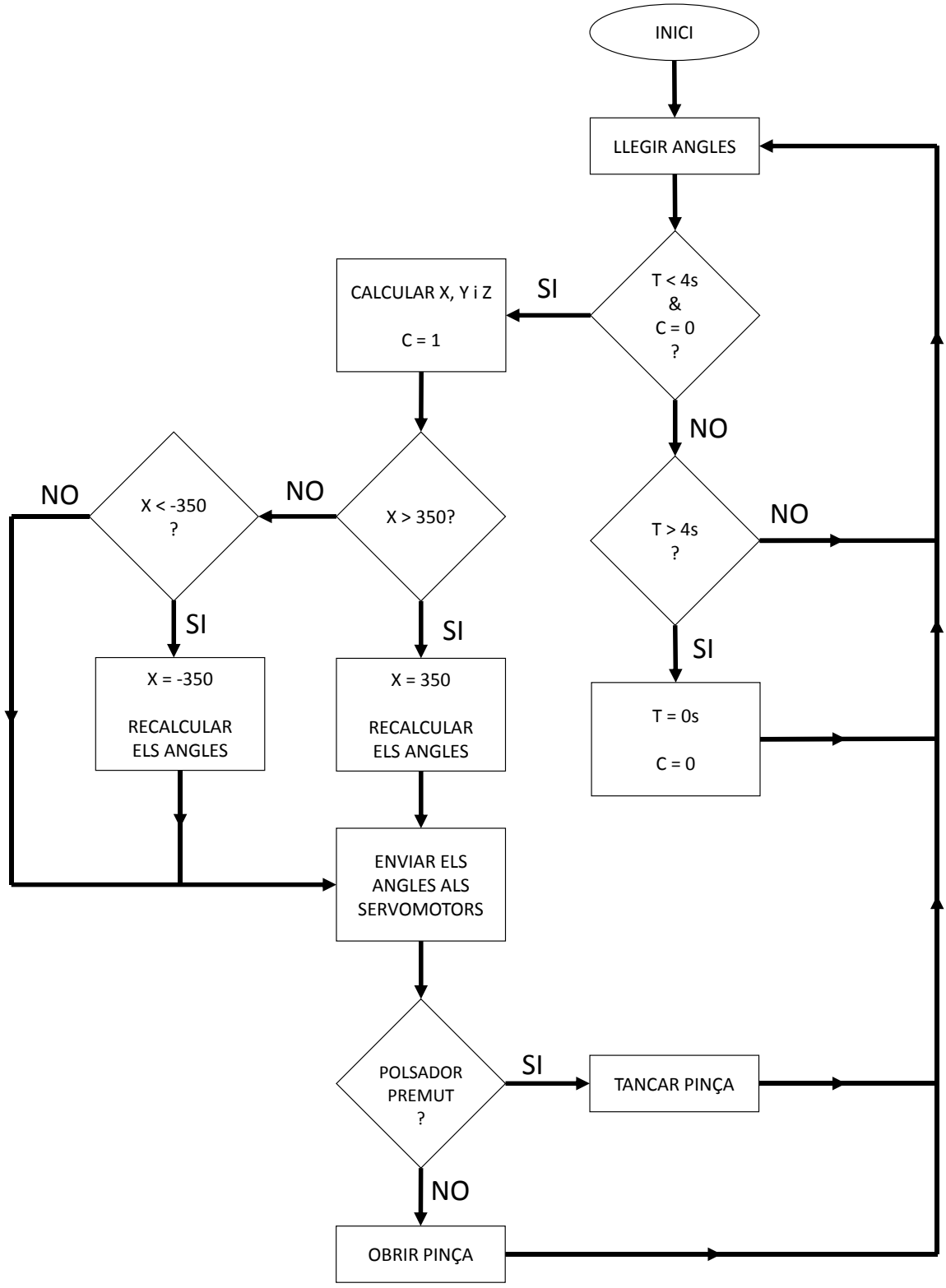


Fig. 20: Diagrama de flux del programa

3.2 Llibreries

Hem utilitzat una llibreria especifica 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 te 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.

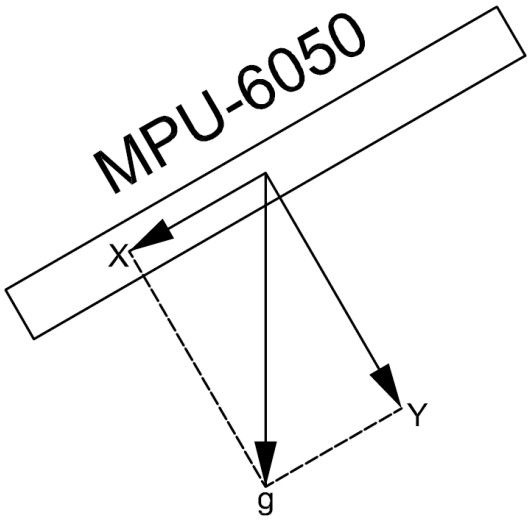


Fig. 21:Lectura de la força de la gravetat en dos eixos en un giroscopi

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 ± 0.5 º/s, per tal que no s'acumuli error, almenys, mentre el sensor romangui 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.

3.3 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 especifiques 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 "debbuging". 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 "debbuging". 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 té un cost de dos mil dos-cents tretze euros i quaranta cèntims, sense IVA.

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 dur 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 guant.

El braç te 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 te 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 Electrònica Industrial i Automàtica.

Girona, 6 de juny de 2019

6. Relació de documents:

Aquest projecte conté els següents documents: Memòria, Plànols, Plec de Condicions, Estat d'Amidaments i Pressupost.

7. Glossari

USB	Bus Sèrie Universal
PLA	Àcid Polilàctic
I^2C	Inter-Integrated Circuit

A. CÀLCULS

A.1. 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 directe a fi de trobar la posició X, Y i Z de la punta del final del braç en funció de θ_1, θ_2, θ_3, θ_4 i θ_5, essent θ_1 l'angle de rotació de l'esquena, θ_2 l'angle de rotació de l'espatlla, θ_3 l'angle de rotació del colze, θ_4 l'angle de rotació del canell i θ_5 l'angle de rotació de la pinça.

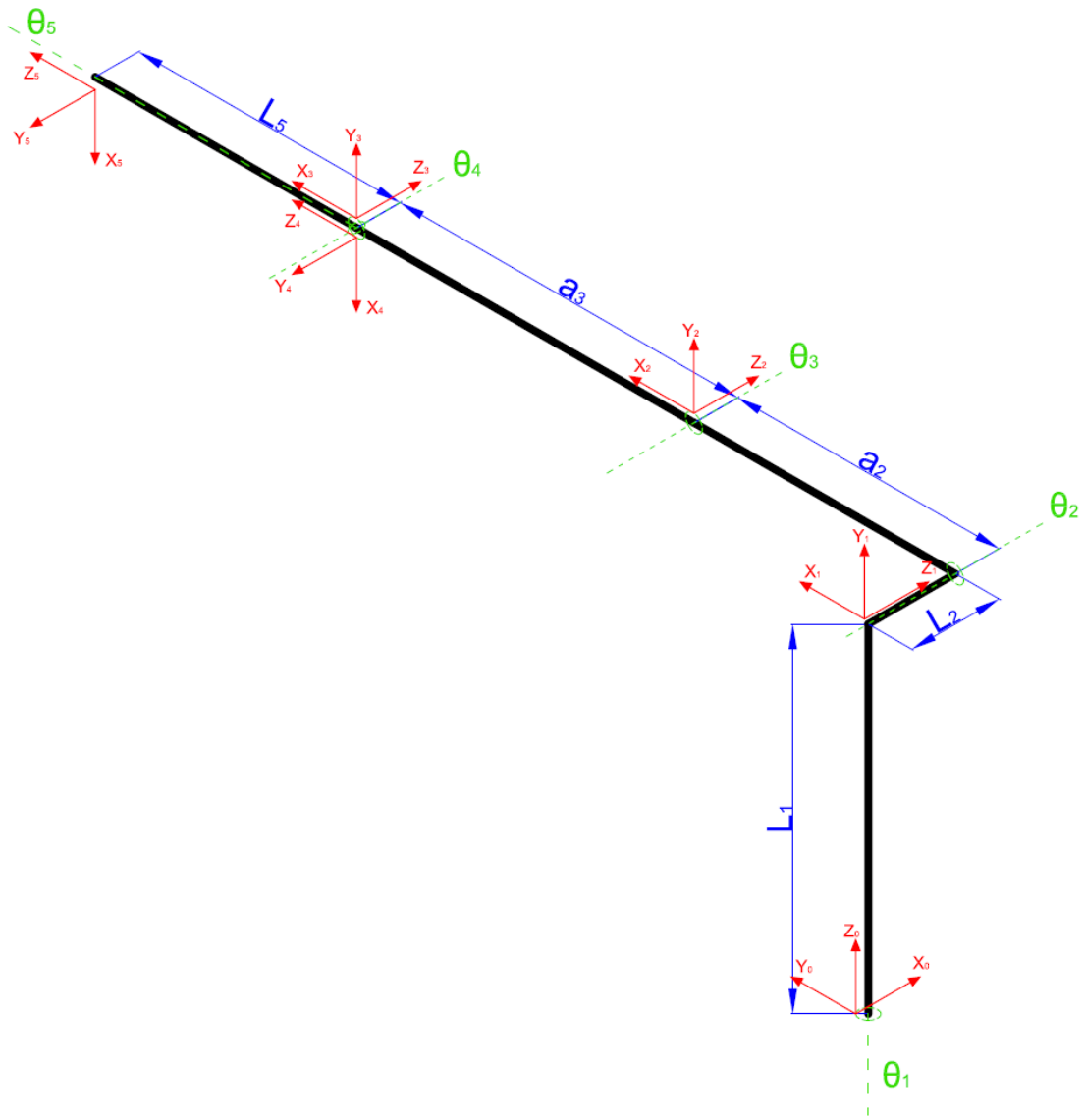


Fig. 22: Representació vectorial del braç robòtic des d'un punt de vista isomètric

Tanmateix, també haurem de ser capaços de trobar els valors de θ_1, θ_2, θ_3, θ_4 aplicant cinemàtica inversa quant li proporcionem X, Y i Z.

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

Utilitzant matrius de rotació i matrius de translació podem transforma 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 variables amb les quals reconeixem cada dimensió del braç.

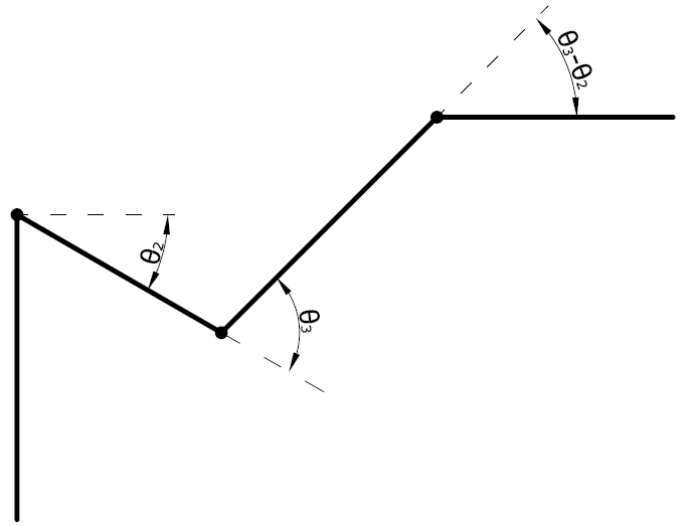


Fig. 23: Representació gràfica de la relació entre θ_4 i θ_2 i θ_3.

De 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.

Tal i com hem explicat anteriorment, només volem buscar les coordenades X, Y i Z de la punta del braç. La orientació no ens és d'utilitat. Dit això, a l'hora de realitzar els càlculs no tindrem en compte θ_5 ja que només repercuteix l'angle de rotació de la pinça.

L'angle de rotació del canell, el qual està vinculat a θ_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, podem eliminar una variable, si suposem la següent igualtat:

$$\theta_4 = \theta_3 - \theta_2 \quad \text{(Eq. 4)}$$

A.2. Càlcul matricial

La ultima columna de la matriu és la única que utilitzarem perquè és la única que fa referencia 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 lleis de Denavit–Hartenberg.

Equacions del nus 0 al nus 1:

$$Rot(Z,\theta_1) = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 5)}$$

$$Rot(X,90°) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(90°) & -\sin(90°) & 0 \\ 0 & \sin(90°) & \cos(90°) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 6)}$$

$$Trans(Y,L_1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 7)}$$

I en acord a les equacions anteriors obtenim la següent:

$$^{0}A_1 = Rot(Z,\theta_1)\cdot Rot(X,90°)\cdot Trans(Y,L_1) \quad \text{(Eq. 8)}$$

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

Equacions del nus 1 al nus 2:

$$Rot(Z,\theta_2) = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 9)}$$

$$Trans(Z,L_2) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 10)}$$

$$Trans(X,a_2) = \begin{pmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 11)}$$

I en acord a les equacions anteriors obtenim la següent:

$$^{1}A_2 = Rot(Z,\theta_2)\cdot Trans(Z,L_2)\cdot Trans(X,a_2) \quad \text{(Eq. 12)}$$

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

Equacions del nus 2 al nus 3:

$$Rot(Z,\theta_3) = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 13)}$$

$$Trans(X, a_3) = \begin{pmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 14)}$$

I en acord a les equacions anteriors obtenim la següent:

$$^2A_3 = Rot(Z, \theta_3) \cdot Trans(X, a_3) \quad \text{(Eq. 15)}$$

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

Equacions del nus 3 al nus 4:

$$Rot(Z, \theta_4) = \begin{pmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 16)}$$

$$Rot(X, -90°) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-90°) & -\sin(-90°) & 0 \\ 0 & \sin(-90°) & \cos(-90°) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 17)}$$

I en acord a les equacions anteriors obtenim la següent:

$$^3A_4 = Rot(Z, \theta_4) \cdot Rot(X, -90°) \quad \text{(Eq. 18)}$$

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

Que segons la equació 4 ens queda:

$$ {}^{3}A_4 = \begin{pmatrix} cos(\theta_3 - \theta_2) & 0 & -sin(\theta_3 - \theta_2) & 0 \\ sin(\theta_3 - \theta_2) & 0 & cos(\theta_3 - \theta_2) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 19)} $$

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

$$ {}^{0}A_4(\theta_1, \theta_2, \theta_3) = {}^{0}A_1 \cdot {}^{1}A_2 \cdot {}^{2}A_3 \cdot {}^{3}A_4 \quad \text{(Eq. 20)} $$

I en acord a les equacions anteriors obtenim la següent:

$$ {}^{0}A_4(\theta_1, \theta_2, \theta_3) = \begin{pmatrix} & R & & X \\ & & & Y \\ & & & Z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Eq. 21)} $$

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 θ_1, θ_2, θ_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 θ_1, θ_2, θ_3 en funció de X, Y i Z.

Moltes d'aquestes equacions, tenen una base trigonomètrica, on ens podíem trobar que θ_1, θ_2, θ_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:

θ_1 pot treballar de 0º a 180º, on 0º seria mirant a mà esquerra, 90º estaria paral·lel al vidre frontal de la campana i 180 graus estaria mirant a mà dreta.

θ_2 pot treballar de -90º a 90º, 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 90º 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 de tenir-lo paral·lel al terra, però tenint en compte l'entorn en el que treballem, tampoc és necessari.

θ_3 treballarà de 0º a 180º, on 0º voldria dir tenir l'avantbraç paral·lel al braç i 180º 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.

B. PROGRAMA

B.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
  tcaselect(0);
  mpu60500.begin();
  mpu60500.calcGyroOffsets(true);

  tcaselect(1);
  mpu60501.begin();
  mpu60501.calcGyroOffsets(true);

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

  tcaselect(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()
{
  tcaselect(0);
  mpu60500.update();
  tcaselect(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 establirem que X = 350 i recalcularem 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 +
 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;
}
```

```
}

//si X és més petita que -350 establirem que X = -350 i recalcularem 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 = theta1 * 180 / pi;
  theta2 = theta2 * 180 / pi;
  theta3 = theta3 * 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(' ');
}
```

C. SIMULACIÓ INFORMÀTICA DEL BRAÇ

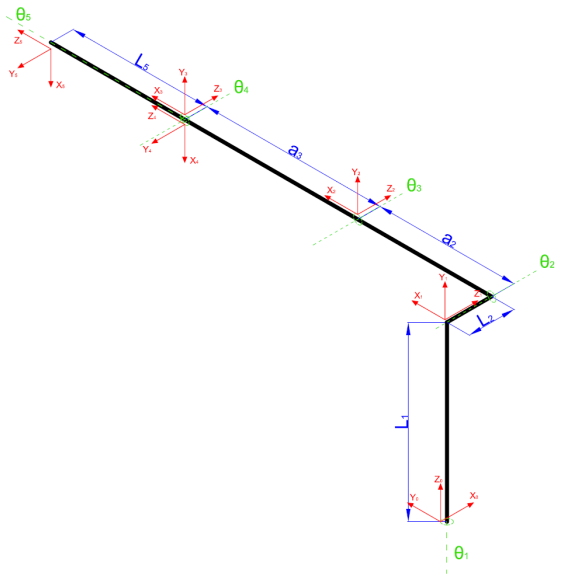


Fig. 24: Esquema del braç robòtic amb vectors

C.1. Simulació del braç amb Matlab i Robotics Toolbox

Tal i com hem vist a l'apartat 3 Descripció del programa, l'Arduino haurà de calcular, mitjançant cinemàtica directe, 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.

	θ	d	a	α	H
1	q_1	L_1	0	90º	90º
2	q_2	L_2	a_2	0	0
3	q_3	0	a_3	0	0
4	q_4	0	0	-90º	-90º
5	q_5	L_5	0	0	0

Taula 1: 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.

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.

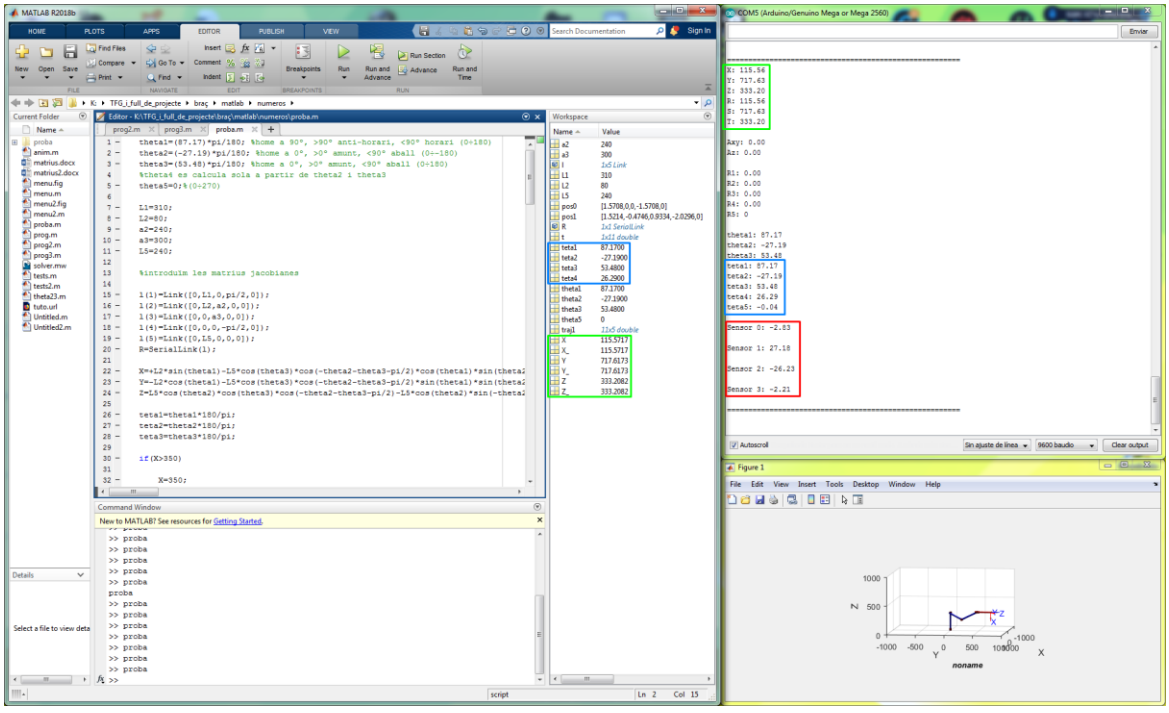


Fig. 25: Simulació en paral·lel entre Matlab i Arduino sense tornar a calcular els angles

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 ± 350 no hem necessitat segon càlcul per tant, els valors de les "thetes" i les "tetes" coincideixen.

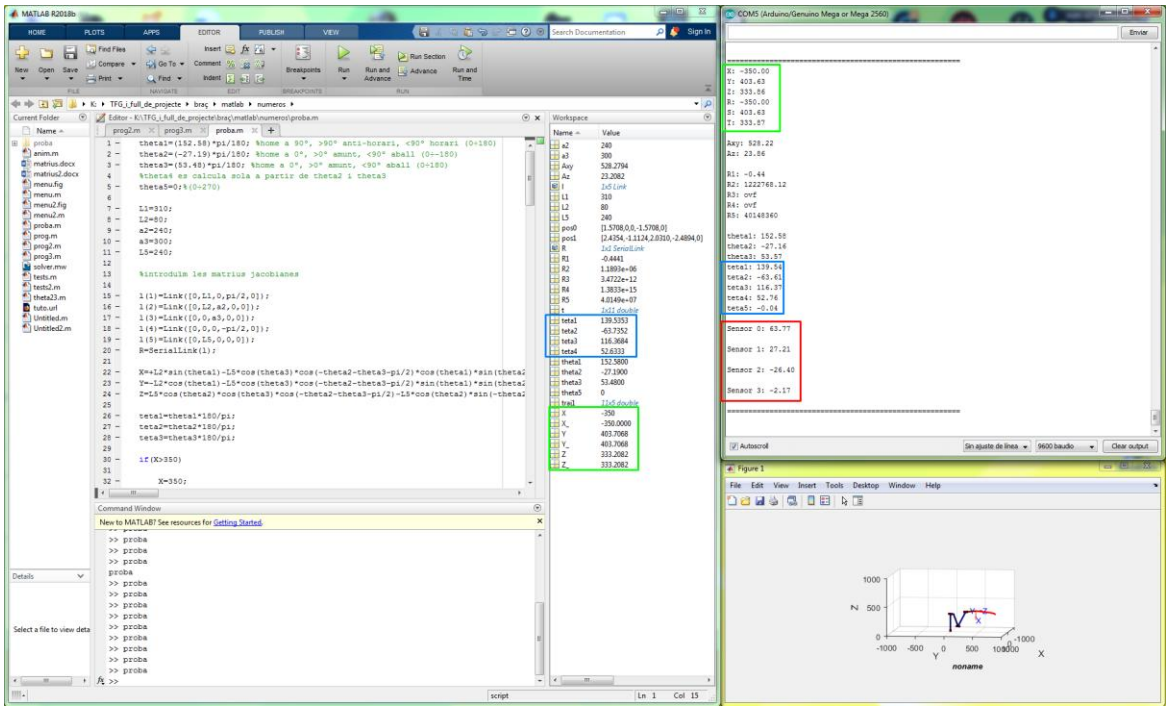


Fig. 26: Simulació en paral·lel entre Matlab i Arduino tornant a calcular els angles

En canvi la figura 33 podem veure que com que X en una primera instància estaria fora del rang de ± 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.