



TecnoCampus
Escola Superior
Politécnica

Centre adscrit a la



Universitat
Pompeu Fabra
Barcelona

Enginyeria Tècnica Industrial: Especialitat Electrònica Industrial

DISSENY ELECTRÒNIC D'UN MANIPULADOR TELEOPERAT

Memòria

Marc Coll Coll
PONENT: Joan Ramon Gomà

TARDOR/PRIMAVERA 2022



TecnoCampus
Mataró-Maresme

Resum

En aquest informe es mostrarà com s'ha realitzat el disseny electrònic d'un manipulador teleoperat. L'objectiu final d'aquest projecte és mostrar els coneixements adquirits al llarg de tota la carrera d'electrònica (és un treball de final de grau) a la vegada que es dissenya una sèrie de circuits electrònics que eventualment es podrien utilitzar per construir el sistema de control del manipulador teleoperat descrit en aquesta memòria. Cal comentar que, encara que aquest TFG conté informació de practicament tot lo necessari, el manipulador i sobretot el controlador tècnicament segueixen en fase de desenvolupament, amb lo qual tot està 100% pulit.

Resumen

En este informe se mostrará como se ha realizado el diseño electrónico de un manipulador teleoperado. El objetivo final de este proyecto es mostrar los conocimientos adquiridos a lo largo de toda la carrera de electrónica (es un trabajo de final de grado) a la vez que se diseña una serie de circuitos electrónicos que eventualmente se podrían usar para construir el sistema de control del manipulador teleoperado descrito en esta memoria. Cabe mencionar que, aunque este TFG contiene información de prácticamente todo lo necesario, el manipulador i sobretodo el controlador técnicamente siguen en fase de desarrollo, con lo cual todo no está 100% pulido

Abstract

This report will show how the electronic design of a teleoperated manipulator has been made. The ultimate goal of this project is to show the knowledge acquired throughout the electronic career (it is a final degree project) while designing some electronic circuits that could eventually be used to build the control of the teleoperated manipulator described in this memory. . It is worth noting that, while this TFG contain information about practicly everything needed, the manipulator and specially the controler is tecnichly in a development procees, meaning that not everything has been 100% polished yet.

Índex.

Índex de figures.....	III
Índex de taules.	V
Glossari de termes.	VII
1. Objectius.	1
1.1. Propòsit.....	1
1.2. Finalitat.....	2
1.3. Objecte.....	2
1.4. Abast.....	2
1.5. Context en les línies de recerca del Tecnocampus.	3
2. Introducció teòrica.....	5
2.1. Marc conceptual.....	5
2.2. Antecedents investigats inicialment.	7
2.3. Anàlisi dels paràmetres d'un braç humà.....	9
3. Disseny de l'electrònica del manipulador.	13
3.1. Bases de l'electrònica del manipulador.	13
3.2. Tipus d'actuadors a considerar.	15
3.3. Exemple de circuit potencial.	17
3.4. Circuit electronic utilitzat.	19
3.4.1. Inversor trifasic i feedback de corrent.	22
3.4.2. Motor i sensors de posició.	23
3.4.3. Controlador digital PID.....	23
3.4.4. Descodificació dels sensors HALL.....	24
4. Disseny de l'electrònica de l'aparell de control.	29
4.1. Diferències entre el manipulador i el seu controlador.	29

4.2. Calcul de la traducció de consignes.....	29
5. Disseny del feedback de control.	35
5.1. Disseny lògic del feedback a utilitzar.	35
5.2. Programació dels aparells informatics.....	37
5.2.1. Programació del PID del Edge Computer.....	38
5.2.2. Programació d'altres funcions del Edge Computer.	40
5.2.3. Progrmació de la Fog Computer (CPU).....	43
6. Impacte mediambiental.	47
7. Propietat intel·lectual i copyright.	49
8. Perspectiva de gènere e igualtat.....	51
9. Conclusions. ***	53
10. Referències.....	55

Índex de figures.

Fig. 2.1. Xarxa LAN d'un manipulador teleoperat [1] [2].....	5
Fig. 2.2. Extremitats superiors de l'esquelet humà [3]	10
Fig. 2.3. Model simplificat d'un braç humà [1]	10
Fig. 3.1. Pins d'un microcontrolador ARDUINO DUE [4]	15
Fig. 3.2. Circuit elèctric d'un inversor per motors Brushless [1] [8].....	17
Fig. 3.3. Esquema de les 3 fases de condicionament de la senyal d'un sensor [1] [8].....	18
Fig. 3.4. Esquema del circuit analògic per descodificar sensors HALL en PSIM [1] [10].....	19
Fig. 3.5. Components d'un circuit de control d'un motor BLDC [6].....	20
Fig. 3.6. Valors que 2 grups de sensors HALL mostren al llarg d'una revolució [1] [7].....	21
Fig. 3.7. Circuit elèctric del circuit de potència (inversor trifàsic) [1] [10]	22
Fig. 3.8. Circuit utilitzat per implementar la lògica de control de la Taula 3.1 [1] [10]	25
Fig. 4.3. Comparació de l'evolució dels sistemes de coordenades [1] [11] [15]	30
Fig. 4.4. Resultats fixos calculats mitjançant Excel (posició de repos) [1] [12]	31
Fig. 4.5. Resultats fixos calculats mitjançant MatLab (Posició de repos) [1] [13]	32
Fig. 4.6. Codi de MatLab generalitzat [1] [13]	32
Fig. 4.6. Matriu de rotació “resumida” de l'espatlla del manipulador [1] [13]	33
Fig. 4.7. Matriu de rotació “resumida” de l'espatlla del manipulador [1] [13]	34
Fig. 5.1. Diagrama flowchart dels diferents components del sistema [1] [8].....	36
Fig. 5.2. Fotografia del Data Hub utilitzat per obtenir els ports USB 3.0 addicionals [8]	43

Índex de taules.

Taula 2.1. Graus de moviment (DoF) en cada direcció [4].....	11
Taula 3.1. Taula de la veritat per descodificar els 16 estats de 3 sensor HALL [1] [6].....	25
Taula 5.1. Taula de la veritat utilitzat per en el programa de l'Arduino [1] [6].....	42

Glossari de termes.

φ	Angle entre la posició d'una part del manipulador i un punt de referència
ω	Velocitat de gir d'una part del manipulador. És la derivada de l'angle φ
BJT	Transistor bastant utilitzat controlat via corrent (<i>Bipolar Junction Yransistor</i>)
BLDC	Motor de corrent continu sense escombretes (<i>Brushless Direct Current</i>)
CdG	Centre de gravetat, localitzat en el centre teòric d'un objecte. Sinònim de CdM
CdM	Centre de masses, localitzat en el centre teòric d'un objecte. Sinònim de GdG
CPU	Microcontrolador principal que controla part del sistema (<i>Central Processing Unit</i>)
DIY	Component que ha sigut dissenyat específicament pel projecte (<i>Design It Yourself</i>). Oposat d'un producte comercial comprat directament a un proveïdor o fabricant.
DoF	Graus de llibertat (Degrees of Freedoms) que indiquen que classes de moviments pot realitzar una articulació (eixos de rotació i/o translació)
Driver	Circuit electrònic encarregat de controlar algun actuator, adaptant les senyals que li arriben (consignes) en cas que sigui necessari
Encode	Dispositiu electrònic utilitzat per mesurar el moviment angular respecte un eix de rotació. És un dels sensors més importants per mesurar la φ
ESC	Circuit electronic encarregat de controlar la velocitat del motor adaptant les consignes de la CPU (<i>Electronic Speed Controler</i>). Alguns models comercials permeten decidir si es vol controlar la velocitat, la posició o el parell motor.
EUPMT	Escola Universitària Politècnica de Mataró
F	Força lineal aplicada a un punt determinat. Sol significar valor (en Newtons) de la carrega aplicada al manipulador i/o reacció causada per alguna articulació.
IGBT	Transistor bastant utilitzat controlat via tensió (<i>Insulated Gate Bipolar Transistor</i>)
m	massa (generalment en kilograms) que té un objecte o part del manipulador

VIII

PFC Projecte Final de Carrera. Sinònim de TFC i TFG

PWM Modulació de l'ample de pols (*Pulse Width Modulation*)

TFC Treball de Final de Carrera. Sinònim de PFC i TFG

TFG Treball de Final de Grau. Sinònim de PFC i TFC

1. Objectius.

1.1. Propòsit.

L'objectiu principal d'aquest TFG serà dissenyar les parts electròniques d'un sistema de control tele operat. És a dir un sistema que detecta els moviments que l'usuari fa per tal de transformar-los en unes consignes per operar un manipulador de forma bastant precisa. Cal comentar que aquest treball formarà part d'una parella de TFGs (un tractarà sobretot la part mecànica, mentre l'altre es centrarà en l'electrònica i el sistema de control), sent necessari llegir els 2 TFGs per tal d'entendre al 100% el disseny final que detallaran.

En quant a aquest TFG, els seus objectius principals seran:

- Disseny dels sensors de moviment principals (aparell que genera les consignes del robot) que cal col·locar en el braç de l'usuari. Aquests sensors probablement acabaran sent integrat en una peça de roba (enganxat a l'exterior de la mateixa) o part d'un exoesquelet.
- Disseny de l'electrònica de l'exoesquelet que permetrà un feedback de força cap al braç de l'usuari (replicarà la força que esta suportant el manipulador en el braç de l'usuari). Aquesta electrònica contarà principalment dels "divers" dels actuadors a utilitzar per generar el feedback de força. Aquest feedback replicarà la força que esta suportant el manipulador en el braç de l'usuari a més de potencialment aprofitar els sensors de posició del manipulador. És interessant recordar que és possible que aquest exoesquelet es fusioni amb el controlador (utilitzar únicament 1 aparell per realitzar les 2 funcions), lo qual obligaria a dissenyar els 2 circuits tenint en compte que formaran part d'un mateix aparell.
- Disseny de de la part electrònica del manipulador. Un cop més el circuit de "divers" dependrà en gran part dels actuadors (motors) utilitzats. Aquest manipulador serà controlat utilitzant consignes de posició (on ha d'anar) i potencialment també consignes de força (màxima) a aplicar, per replicar amb major fidelitat lo que succeirà.
- Disseny del loop de control (PID) que conte els sensors del robot. Aquests sensors són els que s'utilitzaran per generar un llaç de control que garanteixi una precisió optima (sensors de posició que assegurin que posició manipulador \approx posició controlador), juntament amb crear el feedback de força cap a l'exoesquelet del controlador (la força que rep el manipulador serà aplicada en l'exoesquelet del feedback de control).

- Disseny dels microcontroladors que serviran com CPUs del sistema, juntament amb creació d'un programa informàtic que serveixi per dur a terme lo que necessitem.

1.2. Finalitat.

Des de l'inici de la robòtica durant la 3^a revolució industrial, l'ús de per substituir a operaris a l'hora de realitzar tasques manuals repetitives s'ha desenvolupat moltíssim. Desgraciadament aquest camp de l'automatització sol topa amb dificultats a l'hora d'aplicar-se en certs treballs. En general els robots són perfectes per manipular objectes sòlids, rígids i amb dimensions molt precises, però té problemes bastant importants al manipular objectes flexibles, elàstics i/o amb formes molt variables. Mentre que en el primer cas és factible programar el manipulador de forma precisa (els moviments rarament s'han d'alterar), per manipular objectes del 2n grup farà falta una tecnologia més complexa, que aprofiti les virtuts de la indústria 4.0 actual. Tecnologia que dissenyarem en aquest TFG:

1.3. Objecte.

Desgraciadament, degut a la complexitat del disseny exposat en aquest TFG, no s'espera deixar objectes físics (prototips o manipuladors ja funcionals) més enllà informe (informació, plànols i programes informàtics). Cal comentar que degut a no poder-se fer proves amb un prototip físic, és possible que fos necessari iterar el disseny un cop es construís (modificar algun actuator per posar-ne un de més potent) a l'observar fenòmens que es creien menyspreables.

1.4. Abast.

Tal com s'ha comentat anteriorment, l'abast d'aquest TFG únicament incorporarà parts relacionades amb el disseny electrònic (incloent la programació de la CPU) o que son necessàries per comprendre'l (es mencionaran dimensions i/o necessitats dels actuadors / sensors). Dit d'una altre forma, detalls relacionats purament amb la part mecànica i/o telecomunicació dels diversos aparells no s'inclouran en aquest TFG.

Cal tenir en compte que degut a la complexitat d'una ma humana, s'ha decidit que el manipulador únicament representaria la zona entre l'espatlla i el canell (canell no inclòs). Dit

d'una altre manera, el manipulador representa l'espatlla, el braç "superior" (la mitat superior, en ocasions anomenat simplement "braç"), el colze i l'avantbraç (mitat inferior). Deixant tant el canell com la mà fora de l'abast, ja que es considerarà que aquest tipus d'eina (part que s'acobla al final del braç robòtic) té una complexitat que podria requerir un TFG addicional.

1.5. Context en les línies de recerca del Tecnocampus.

Aquest TFG (i la seva parella) va ser elegit per el professor ponent (membre del Tecnocampus) degut a que tenia us bastant clar per les línies de recerca del centre:

Aquest projecte "doble" (el qual ha sigut proposada per el tutor dels TFGs) ha sigut escollit ja que, un cop acabats, aquets TFGs podran ser utilitzat com a base d'altres treballs d'investigació (ja sigui en forma de TFGs o no) on es podria aplicar aquest disseny (construir-lo) i/o seguir-lo ampliant per tal d'aplicar-lo a un us pràctic.

Concretament, aquest projecte és el primer pas (disseny de les parts essencials) per arribar a obtenir un equip de laboratori que permeti investigar l'automatització de tasques molt més complexes que impliquen objectes amb variabilitat de forma i/o materials elàstics i altament flexibles. Sent l'idea principal que un cop dissenyat i construït tant el manipulador (braç robòtic que manipularà els objectes) com l'aparell de control (peça de roba i/o exoesquelet amb els sensors i feedbacks de força necessaris), un usuari es col·locaria l'aparell de control i faria els gestos necessaris perquè el manipulador realitzi la tasca que es vol automatitzar. Tot això amb la intenció de enregistrar tant els moviments de l'usuari (consignes electròniques que arriben a les CPUs) com el vídeo que 2 (o més) càmeres han gravat on es veu perfectament els moviments que el manipulador ha realitzat. Finalment amb aquestes consignes i vídeos es realitzaria una base de dades amb molts experiments diferents, que involucrin el manipulador agafant diferents objectes conflictius, per tal d'obtenir informació útil amb la que alimentar alguna IA de Machine Learning (i/o Deep Learning) per tal de que la pròpia IA pogués començar a solucionar de forma "automàtica" problemes similars als que l'operari li hagi ensenyat.

2. Introducció teòrica.

2.1. Marc conceptual.

Aquest TFG tracta sobre dissenyar un manipulador teleoperat. Com a norma general, anomenem “manipulador teleoperat” a un conjunt de components electrònics i/o mecànics que permeten moure un manipulador de forma que imiti els moviments d’un usuari. En aquest cas tindrem un robot articulad (tots els moviments seran girs) que replicarà els seus moviments.

Aquest tipo de sistemes mecànics pot arribar a ser molt complex, degut a la gran quantitat de parts a considerar. A continuació és pot observar una xarxa LAN que incorpora la majoria d’aparells que es poden arribar a necessitar:



Fig. 2.1. Xarxa LAN d'un manipulador teleoperat [1] [2]

Es pot observar que aquest sistema conte 4 subsistemes principals:

1) Camara de videovigilància i ordinador (o pantalla) on veure que esta fent el manipulador “in real time”. Útil en cas que es necessiti fer zoom i/o no hi hagi una línia de visió directa.

2) Les telecomunicacions entre diferents aparells (cables, router amb accés a internet i switch per connectar-ho tot), juntament amb un servidor on guardar informació útil (en cas que es vulgui tenir registre de l'ús del manipulador i/o del vídeo de la càmera de vigilància).

3) Connexió entre el hardware del manipulador (motors i sensors de realimentació) i la seva CPU (placa base que serveixi de I/O per rebre o enviar informació a més de controlador PID per assegurar uns moviments precisos i relativament òptims).

4) Connexió entre el hardware de l'aparell de control (sensors de posició i actuadors del feedback de força) i la seva CPU (placa base que serveixi de I/O).

Afortunadament casi tots els components electrònics es poden trobar al mercat i no requereixen gaires coneixements per ser instal·lats. Degut a això, l'única part important a tenir en compte és el hardware dels subsistemes 3) i 4), juntament amb la programació de les seves CPUs.

Des d'un punt purament electrònic, únicament s'ha de dissenyar 5 components:

1) El circuit electrònic del robot: Un conjunt de diferents motors (articulacions), sensors (ja siguin del llaç de control de posició o del feedback de força) i subcircuitos de potencia.

2) CPU del robot: S'ha d'incorporar un microcontrolador (CPU) per tal de gestionar els dades dels sensors, processar-les (control PID digital) i enviar-les utilitzant el canal indicat (ja sigui a l'altre CPU o a un actuator en particular del robot).

3) El circuit electrònic del controlador: Circuit que connecta els diferents sensors de les diferents parts del cos i adapta la seva senyal per tal de ser apte per ser processada amb el seu microcontrolador.

4) Actuadors del feedback de força: Una estructura en forma d'exoesquelet que conte diferents tipus d'actuadors (pistons, motors, fils, etc.) que intenten replicar els esforços de compressió i/o tracció que afecten al robot. Encara que aquest component no és essencial, ja que normalment el simple fet de veure la posició actual del manipulador (amb o sense ajuda d'una càmera de videovigilància) és suficient, aquesta complexitat addicional sol permetre que l'usuari realitzi un control més intuïtiu (detecta que parts fa contacte).

5) CPU del controlador / feedback de força: S'ha d'incorporar un microcontrolador (CPU) que gestiona tan els sensors de l'aparell de control com els actuadors del feedback de força. Ha de codificar les senyals del sensor per enviar-les a la CPU del robot, a la vegada que rep instruccions d'aquella CPU que ha d'enviar als actuadors del feedback de força.

Finalment, cal comentar un parell de definicions importants que caldrà tenir en compte:

- **Usuari:** Persona utilitza l'aparell de control en un dels seus braços per tal de controlar el manipulador teleoperat. Únicament una persona es pot col·locar l'aparell de control en un dels seus braços, per tal que enregistri els moviments del mateix, de manera que únicament pot haver-hi un usuari controlant el manipulador teleoperat.
- **Robot / Manipulador:** És la màquina que es mou (gracies a diversos motors), seguint les consignes que li indiquen la posició on s'ha de moure. En alguns casos “manipulador teleoperat” pot referenciar a tot el sistema (la xarxa LAN) de forma general.
- **Controlador / Aparell de control:** És l'aparell gracies al qual es pot enregistrar els moviments de l'usuari (mitjançant diversos sensors de moviment i/o posició), per tal de crear les consignes que s'envien al robot. És important no confondre-ho amb un “controlador PID / controlador digital”, lo qual és un aparell electrònic i/o programa informàtic part de la CPU del manipulador que no forma part de l'abast d'aquest TFG.
- **Feedback de força:** És el conjunt d'actuadors que s'uniran amb l'aparell de control per tal de generar forces en el braç de l'usuari equivalents a les que afecten les diferents parts del robot. El seu us principal és que l'aparell faci pressió en punts del braç de l'usuari per tal de que tingui una idea aproximada de que parts del manipulador esta en contacte amb algun objecte (per exemple per saber si ha glopejat una paret o ja ha fet contacte amb l'objecte a manipular). En ocasions es pot mencionar els “sensors del feedback de força” fent referencia a sensors de pressió i/o força instal·lats en el manipulador.

2.2. Antecedents investigats inicialment.

El concepte d'un manipulador que es mogui seguint les instruccions que un usuari li indica no és res gaire revolucionari. En major o menor mesura la majoria de manipuladors no-autònoms (o amb un mode “manual”) es podrien considerar manipuladors teleoperats. Degut a això és important delimitar el rang de manipuladors que considerarem “similars” al que estem intentant dissenyar. De forma molt resumida podem trobar 2 grups especialment importants:

- **Els braços prostètics realistes:** Especialment des de l'inici de la impressió 3D, han hagut molts projectes diferents enfocats a intentar escanejar i/o imprimir braços de persones que s'han tingut que amputar un braç (ja sigui degut a un accident o malaltia) per tal de que pugin recuperar un

braç relativament funcional. En els millors casos s'utilitzen manipuladors teleoperats que repliquen la ma i la part del braç amputat, sent que aquests manipuladors són altament complexos (degut a la dificultat de replicar una ma) i solen accionant-se amb consignes obtingudes del propi sistema nerviós de l'usuari. És interessant com les consignes es solen extreure amb un casc que llegeix ones cerebrals, un sensor que detecta l'activitat dels nervis en la zona amputada i/o amb alguna classe de cirurgia permanent.

-Manipuladors teleoperats industrials: Manipuladors utilitzats en diferents indústries per manipular components perillosos o que estan dins d'una zona de difícil accés. A diferència dels braços prostètics aquests manipuladors generalment tindrà una forma menys humana (poden ser més grans, tenir pinces no gaire complexes o simplement tenir un nombre inferior d'articulacions).

Encara que el aquest manipulador segurament tindria un ús més proper al món industrial (no es tindran en compte certes necessitats d'una pròtesi), el disseny del mateix molt probablement tindrà més en comú amb les pròtesis en si (moviments 1:1, amb feedback de força cap a l'usuari).

En quant al sistema de control (sensors que capturaran els moviments de l'usuari) tenim bastantes tecnologies diferents d'on obtenir inspiració. Algunes bastant interessants són:

- Sensors connectats al circuit nerviós: Tal com s'ha comentat anteriorment, aquesta és una forma molt típica d'obtenir aquest tipus de consignes en el món dels braços prostètics d'alta gamma.

- Controladors de videojocs: Durant les últimes generacions de consoles ha sigut bastant típic tenir algun tipus de control de posició (generalment algun acceleròmetre i/o giroscopi) integrat en els controls de certes consoles. Un dels millors exemples ha sigut els Joy-Con de la Nintendo Switch que en les condicions adequades (per exemple les maquetes de cartró de "Nintendo Labo") permeten crear controls de "VR i/o AR" que simulen certs objectes. Degut a això el concepte de simplement replicar les funcions relatives als seus giroscopis i col·locar varies còpies en zones importants del braç pot arribar a ser una estratègia potencialment viable per generar l'aparell de control.

- Controls de Realitat Virtual: Seguint en l'esfera dels videojocs, els aparells més complexos de realitat virtual (especialment els que tenen algun tipus de vestit hàptic) podrien ser una de les solucions més completes del sistema de control. La tecnologia hàptica en VR sol ser aquella on s'utilitza alguna peça de roba crear alguna classe de feedback de força (vibracions, ultrasons

o petits motors) depenent de lo que passa en el mon virtual. Cal comentar que els vestits hàptics generalment tindran integrat un vestits de captura de moviment (senyors de posició).

- Vestits de captura de moviment: El Mo-Cap (Motion Capture) són, tal com s'ha mencionat en el paràgraf anterior, un tipo de vestits amb senyors de posició i/o LEDs que determina els moviments de l'usuari. Encara que es pot trobar molts tipus de vestits Mo-Cap, lo més important és saber si tenen tecnologia hàptica integrada i quin tipo de senyors utilitza (cada sensor sol tenir requeriments mecànics i/o informàtics addicionals).

- Data Gloves: Una subcategoria de vestit Mo-Cap especialitzada en tenir una precisió bastant elevada en una part concreta del cos (generalment una única mà). Degut a la seva mida relativament petita, els millors data gloves es solen poder combinar amb alguna classe d'exoesquelet per tal d'obtenir un feedback de força bastant complex. Encara que el aquet manipulador no inclogui una ma com a tal, és possible que es pugi aplicar parts d'aquesta tecnologia per dissenyar el sistema de control i/o el feedback de força.

Degut a la gran quantitat de tecnologies relatives als sistemes de control (i feedbacks de forces), juntament amb les diferències en el funcionament de la seva electrònica, abans de decidir quin tipo d'estructura s'utilitzarà farà falta realitzar bastanta investigació (en aquest moment no hi ha un candidat potencial únic).

2.3. Anàlisi dels paràmetres d'un braç humà.

Una de les parts més importants de qualsevol classe de robot és quin tipus d'articulacions tindrà. En el cas del manipulador teleoperat dissenyat en aquest TFG una de les prioritats més importants és imitar al màxim les capacitats d'un braç humà. Degut a això és important investigar els valors més importants que un braç humà sol tenir.

Primer de tot, cal saber l'estructura d'un braç humà (ossos i articulacions). A continuació es pot observar un esquelet humà (fig. 2.2) i un model simplificat de les parts del mateix (fig. 2.3):

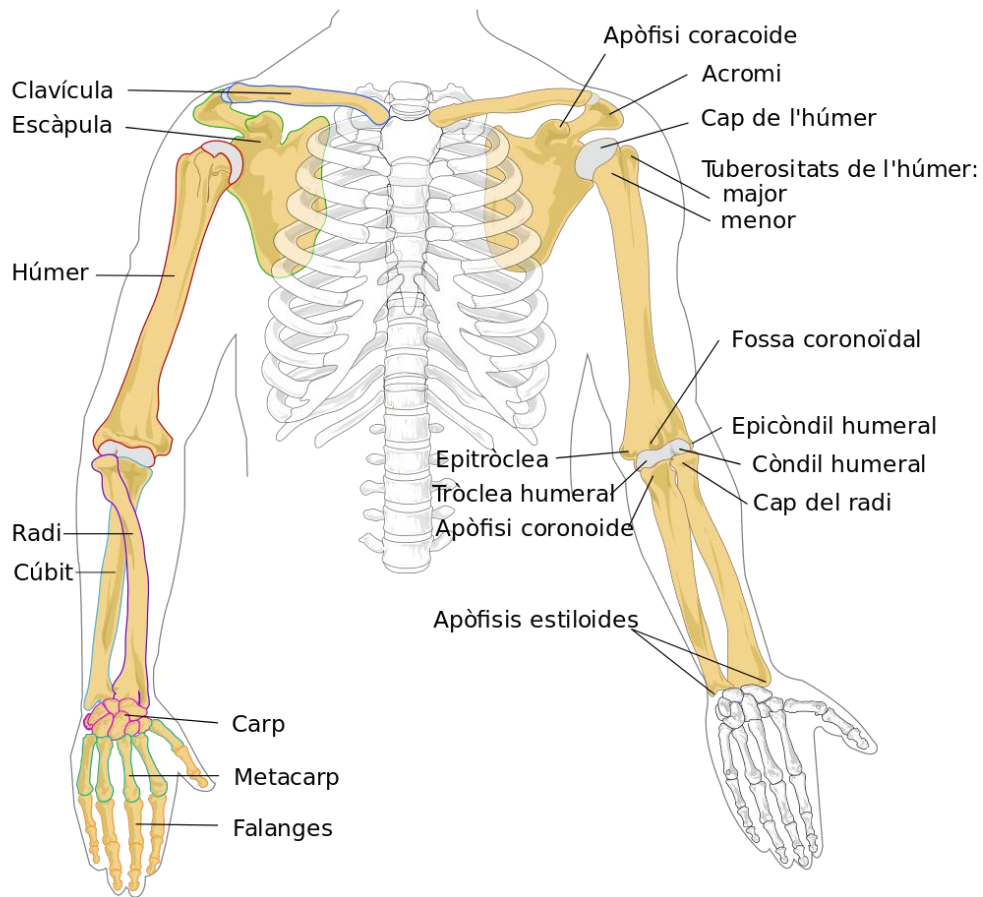


Fig. 2.2. Extremitats superiors de l'esquelet humà [3]

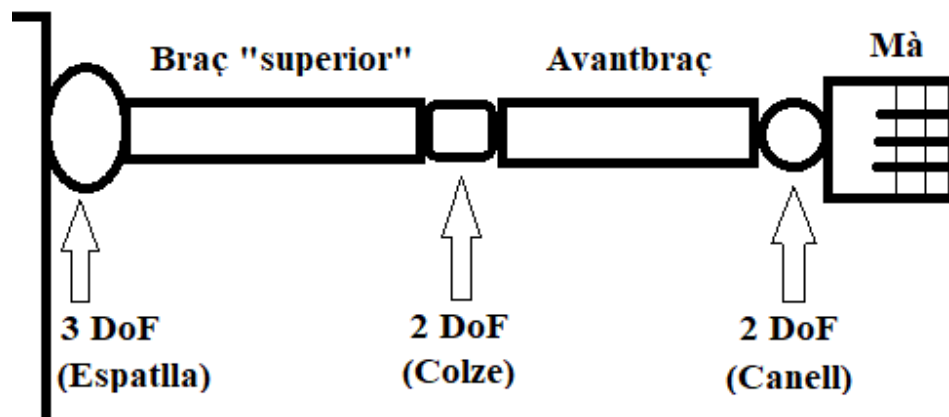


Fig. 2.3. Model simplificat d'un braç humà [1]

Tal com es pot observar en la Fig. 2.3, un braç humà bàsicament està compost per un suport vertical (el tors de la persona), la mà en si (que queda fora de l'abast d'aquest TFG), 2 barres rectes (braç superior i avantbraç) i 3 articulacions, de 2 o 3 graus de llibertat, que uneixen tots els elements anteriorment mencionats. Un cop conegut l'estructura en si, fa falta investigar els següents 4 paràmetres addicionals:

1) Tipus de DoF que té cada articulació i rangs màxims als quals pot arribar. Aquest és un dels paràmetres més fàcils de trobar. Un resum de tots els valors es pot trobar en la taula 2.1.

Articulació	Tipus de moviment	Direcció	Rangs màxims [°]
Estatlla (3 DoF)	Extensió – Flexió	Vertical	50 – 180
	Abducció – Adducció	Horitzontal	180 – 30
	Rotació Interna – Externa	Rotació sobre si mateix	100 – 80
Cotze (2 DoF)	Extensió – Flexió	Vertical	0 – 145
	Pronació - Supinació	Rotació sobre si mateix	85 – 90
Canell (2 DoF)	Extensió – Flexió	Vertical	85 – 85
	Abducció – Adducció	Horitzontal	15 – 45

Taula 2.1. Graus de moviment (DoF) en cada direcció [4]

Cal recordar que en aquest TFG únicament es dissenyarà les 5 primeres articulacions equivalents al cotze i a l'estatlla. Degut a això farà falta dissenyar únicament 2 articulacions de rotació sobre si mateixa (l'eix és paral·lel a la barra que estan movent), una articulació 2D d'aproximadament 150° que simuli el moviment principal del cotze i una articulació (o parella d'articulacions) que simulin el moviment 3D de l'estatlla.

2) Carrega màxima que un braç humà pot manipular sense problemes. La carrega màxima depèn en gran mesura de la posició del braç ja que si el braç està estès, per exemple per posar la carrega en algun lloc lluny, un mateix pes causarà un moment bastant més superior al que s'obté si es col·loca la carrega tan a prop del cos com sigui possible.

Com a norma general, diversos informes d'ergonomia i seguretat laboral han estimat que per treballs continus (moure constantment objectes) la carrega no hauria de superar els 15 kg.

3) Longitud de les 2 “barres” (braç superior i avantbraç). Les dimensions d'un braç humà poden variar bastant, sent que una longitud d'aproximadament 30 cm entre el centre de les articulacions hauria de ser suficientment exacte per qualsevol de les 2 barres si es vol que el manipulador tingui una escala 1:1 amb un braç promig.

Cal recordar que en robòtica augmentar les longitud millora el rang del robot (pot interactuar amb zones més llunyanes) però causa una disminució de la carrega màxima que pot arribar a manipular, ja que causarà moments més elevats l'estar més allunyada.

4) Perfil i diàmetre de les 2 “barres” (mida del seu perfil). Degut a que l'àrea del perfil depèn de molts factors (edat, sexe, genètica, musculatura, etc.) és impossible tenir un valor “definitiu”.

Tenint en compte això i que el braç robòtic només hauria d'interactuar amb l'exterior mitjançant l'eina en si (la mà que es podria dissenyar en un TFG posterior), s'ha preferit no crear cap restricció relacionada amb aquest paràmetre a l'hora de dissenyar el manipulador.

Per altre banda aquesta variabilitat s'haurà de tractar a l'hora de dissenyar el controlador, per tal de permetre que qualsevol persona el pugui utilitzar independentment de la seva forma física, sexe o altre factor que influeixi en la forma i/o longitud del seu braç.

3. Disseny de l'electrònica del manipulador.

3.1. Bases de l'electrònica del manipulador.

Un dels objectius principals d'aquest TFG és dissenyar l'electrònica que permetrà controlar els actuadors que formaran el manipulador teleoperat. Com a norma general es pot observar 4 circuits electrònics que cal considerar a l'hora de dissenyar el manipulador:

1) Drivers dels actuadors: Aquest sistema s'encarrega de transformar la consigna (sortida) de la CPU en una ona elèctrica que tingui les propietats necessàries per generar la resposta desitjada en l'actuador. En el cas del manipulador la consigna ha de garantir un moment (força angular) concret en cada articulació. Aquest és un dels circuits més variables ja que cada tipus d'actuador tindrà unes normes diferents en tant a la forma (pot requerir polsos, una ona sinusoidal, corrent constant...) com a la relació entre paràmetres elèctrics i paràmetres físics.

2) Circuit del sensor de posició: Encara que no s'utilitzi directament en el manipulador, és important mesurar en tot moment la posició de les seves articulacions per tal de que el controlador intenti imitar al manipulador (generant el feedback de força). Aquest circuit s'encarrega de transformar la sortida del sensor de posició (encoder i/o sensor d'efecte HALL) en un voltatge que el ADC del microcontrolador pugui acceptar (per exemple entre 1 ~ 3 V).

3) Circuit del sensor de moment: Part indispensable pel llaç de control del manipulador. La diferència principal és que en la majoria d'actuadors el parell que està generant un moment determinat es pot mesurar utilitzant els paràmetres elèctrics de l'entrada de l'actuador. En molts models d'actuador el moment és directament proporcional a la intensitat (o voltatge) de les entrades, de manera que el "sensor" utilitzat pot ser un simple multímetre que mesuri intensitat i/o voltatge. Cal comentar que en motors hidràulics el moment dependrà únicament de la pressió en la sortida del motor, de manera que únicament en aquest cas el sensor serà un manòmetre. Independentment del sensor utilitzat, el circuit simplement serà incorporar el sensor en qüestió (ha de tenir un pin que generi una sortida elèctrica) al sistema elèctric o hidràulic del motor, connectar la sortida del sensor a un adaptador corrent – tensió en cas que la sortida del sensor utilitzi intensitat en lloc de tensió, aplicar-li un guany (amplificador operacional) en cas que s'hagi de modificar el rang de la sortida i enviar-lo cap al ADC del microcontrolador.

4) Entrades de la CPU: Aquest circuit simplement serveix perquè les entrades (ADC) del microcontrolador es puguin connectar de forma adequada a tots els sistemes de cada sensor.

Degut a tenir 5 articulacions (10 sensors) és possible que aquest circuit consti d'un parell de multiplexor analògic (només una entrada es connecta amb la sortida) que deixin passar únicament 2 entrades (una posició i un parell) cap al ADC de la placa en cada moment. L'alternativa és trobar un microcontrolador amb 10 entrades, tal com un aparell de control prefabricat especialitzat en aquesta classe de situacions.

5) Sortides de la CPU: A diferència de l'entrada, la sortida del microcontrolador ha d'assegurar una consigna constant si es vol moure diverses articulacions al mateix moment. En cas que les consignes siguin valors analògics constants (s'actualitzen 1 vegada per període) es podria aconseguir estabilitzar 5 consignes utilitzant un desmultiplexors (indica cap on ha d'anar la sortida de la CPU), un circuit (o xip) que mantingui el voltatge fins que no s'indiqui que el valor cal ser actualitzat, per exemple connectant els triggers/ables de les 5 memòries analògiques a un descodificador binari (indica quina sortida s'ha d'activar depenent de l'estat de la CPU). Degut a això i a que 5 sortides no són tantes lo ideal es trobar un microcontrolador amb 5 sortides (i per exemple 2 entrades) o intentar que el driver de l'actuador pugui funcionar amb sortides binaries PWM, on la informació es transmet mitjançant el duty ratio. En cas de poder utilitzar sortides PWM lo més probable és que fessin falta únicament 5 sortides (motor pas a pas) o 15 sortides (motor brushless). Cal comentar que encara que 15 sortides PWM és un nombre elevat, és possible trobar microcontroladors amb 16 sortides PWM (2 x 2 x 4).

Cal comentar que per exemple l'ARDUINO DUE conté 12 ADC (2 més dels necessaris), 12 PWM de 8 bits (faltarien 3 sortides PWM addicionals) i 2 DAC, de manera que amb un sol microcontrolador es podria controlar 4 dels 5 actuadors mitjançant tríos de PWM. Problema que es pot solucionar si 1 actuador utilitza una de les sortides DAC (o algunes de les binaries no PWM) o es connectés a un 2n microcontrolador, el qual segurament s'hauria d'instal·lar eventualment per controlar el canell i la mà (es podria fer que també controlés el gir del colze).

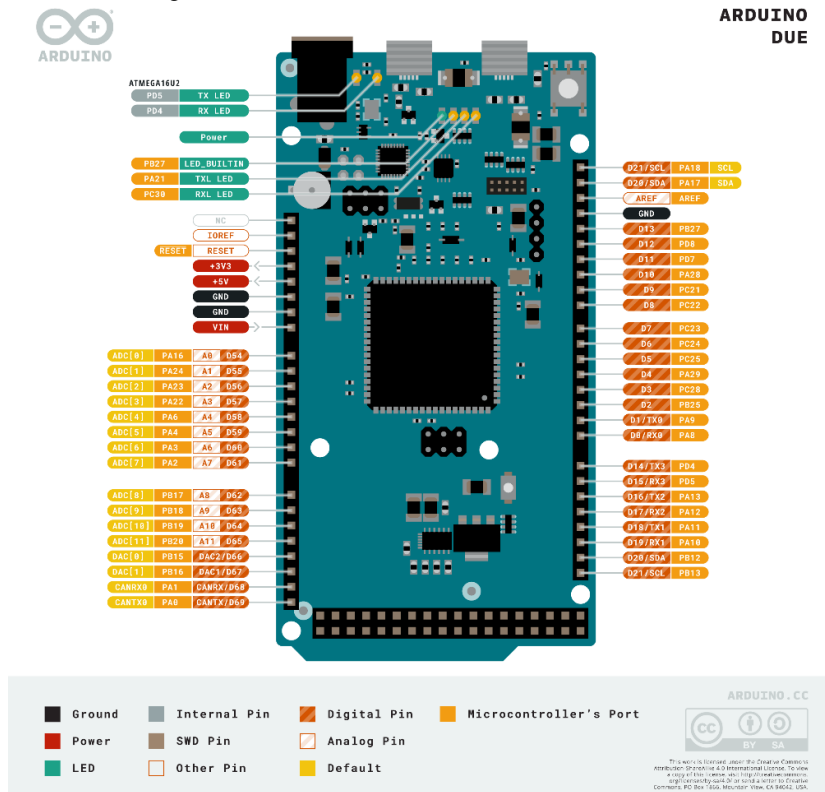


Fig. 3.1. Pins d'un microcontrolador ARDUINO DUE [4]

3.2. Tipus d'actuadors a considerar.

L'electrònica de potencia conté un gran nombre de tipus d'actuadors, els més significatius són:

- 1) Motor asíncron de corrent altern: El motor asíncron és un dels més utilitzats en aplicacions industrials degut a poder-se utilitzar en ambients perillosos i que a dia d'avui no son tan difícils de controlar, gracies xips comercials que modifiquen el voltatge i la freqüència de forma fàcil.
- 2) Motor pas a pas: Motors basats en fer petits salts per cada puls que li arriba. El seu control de posició és relativament intuïtiu ja que es pot fer amb un llaç obert (únicament cal contar nombre polsos / salts) i tenen un parell motor bastant elevat a baixes velocitats. El seu problema principal és que quan s'aplica massa parell pot "saltar-se passos", desincronitzant la posició real amb la que hauria de tenir (no es mou quan li arriba un pols).
- 3) Motor amb escobretes de corrent continu: Un dels motors més simples de controlar sense cap classe de xip comercial degut a requerir una corrent continua on simplement cal modificar el voltatge i/o intensitat. Encara que aquest motor és molt econòmic, té problemes de manteniment (les escobretes es desgasten a l'utilitzar-les), parell bastant petit (es solen especialitzar en velocitats elevades) i el fet de que estan pensats per tenir velocitats constants.

4) Motor sense escombretes de corrent continu: El BLDC (Brushless Direct Current), també anomenat motor sincrònic d'imants permanents, és l'evolució del motor amb escombretes. El BLDC sacrifica la simplicitat de canviar automàticament de polaritat amb les escombretes, el qual s'haurà d'incorporar en el circuit electrònic de control, per tal de millorar bastantes prestacions. Algunes de les millores és una vida útil molt superior (al no haver de canviar escombretes), ser més silenciosos i no tenir una velocitat màxima limitada per el nombre de pols. Apart del control relativament complex, el problema principal d'aquesta classe de motors és que requereixen caixes de canvis per tenir parells motors relativament elevats.

5) Servomotors: Tecnicament no és una classe “normal” de motor sinó un adjectiu que motors de les categories anteriors (principalment AC, DC o BLDC) poden rebre si compleixen certes condicions. Com a norma general s'anomena servomotor a qualsevol motor elèctric que té un circuit de control relativament complex en llaç tancat de forma integrada o que com a mínim es podria controlar de forma externa si es desitges (té el hardware i sensors necessaris).

6) Motor hidràulic: L'última alternativa és substituir l'actuador elèctric per un de hidràulic. Alternativa relativament òptima per aquelles ocasions on el parell màxim que un sistema hidràulic pot subministrar és suficient raó per justificar el cost i la complexitat addicional de disseny d'aquesta classe de sistemes. Els sistemes hidràulics es basen en obrir, tancar i/o invertir vàlvules hidràuliques per deixar passar el fluid des de una bomba (inici) fins un tanc de recuperació (a diferència de la pneumàtica no es pot deixar anar en qualsevol punt). Degut a aquest comportament “binari” (generalment tindrem 1 o 2 entrades binàries en cada vàlvula) i a que el parell motor es pot calcular fàcilment utilitzant la pressió que surt del sistema, el sistema de control d'aquest tipus de motor pot ser relativament simple. El problema més gran és el cost del hardware addicional (vàlvules, bomba...) i deixar espai per tal d'introduir els tubs hidràulics.

Cal comentar que independentment de l'actuador escollit, sempre farà falta un sensor de posició (encoder i/o sensors d'efecte Hall) per conèixer l'angle de cada motor i un multímetre (en el cas del motor hidràulic s'ha de substituir per un manòmetre) per mesurar els paràmetres que indiquen el valor actual del par motor. Sent que la diferència més important serà en el circuit de control que s'exposarà en el TFG sobre el disseny electrònic.

En el moment de l'entrega d'aquest informe parcial encara no s'ha decidit quin model de motor és el més apte (encara s'està investigant diverses possibilitats). Sent les opcions aparentment més òptimes el motor hidràulic si es vol maximitzar el parell, el pas a pas si es vol simplificar el circuit de drivers o el BLDC si es vol un punt intermedi relativament estudiat (hi ha bastants circuits de control integrats en un producte comercial).

3.3. Exemple de circuit potencial.

Encara que no s'ha decidit la classe de motor que es vol utilitzar, és possible idear un circuit potencial i posteriorment modificar les parts que depenen de la classe d'actuador. En aquest cas s'ha suposat que els actuadors són BLDC activats mitjançant senyals PWM.

Els motors BLDC es solen controlar amb senyals PWM on el duty cycle esta canviant constantment per tal de que la tensió mitja segueixi una forma aproximadament sinusoidal amb un voltatge promig concret. Per aconseguir això lo normal és generar les consignes PWM de baixa tensió en una CPU per acte seguit utilitzar un inversor trifàsic d'alta tensió. Un inversor trifàsic és simplement 3 parelles de transistors (generalment IGBT o BJT) on cada transistor funciona com un switch (en tot moment únicament un dels 2 estarà actiu). D'aquesta manera es pot obtenir 3 sortides independents (generalment són la mateixa amb un desfasament de $\pm 120^\circ$) que oscil·len entre 0V (GND) i el voltatge d'alimentació del sistema. Un diagrama d'aquesta classe de circuit trifàsic es pot observar en la fig. 3.2:

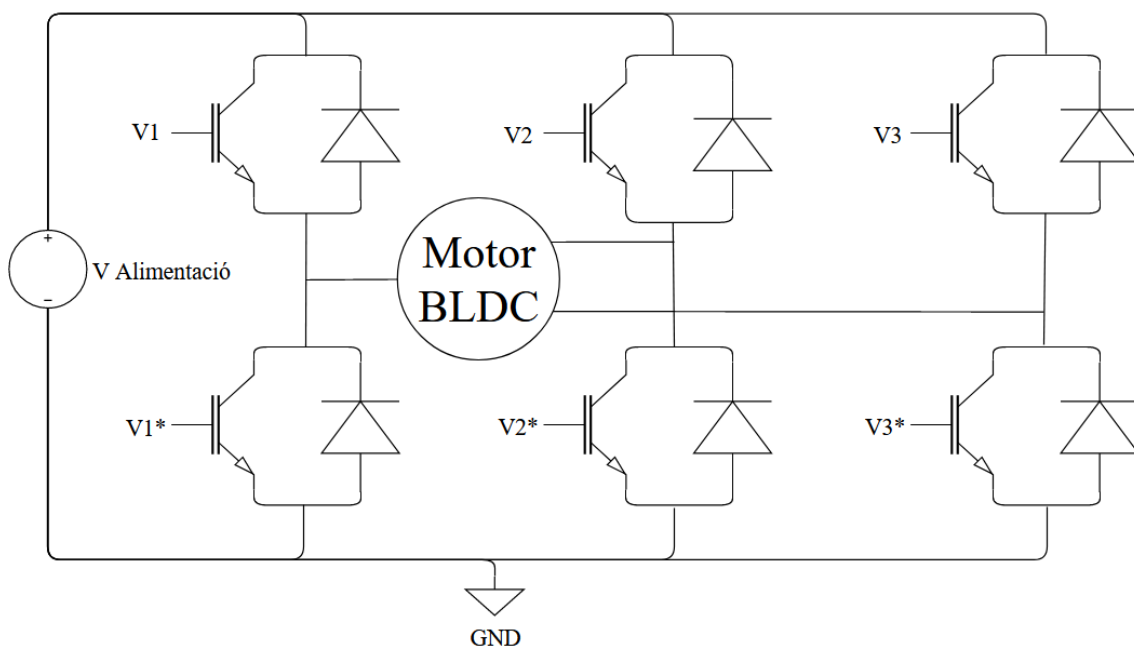


Fig. 3.2. Circuit elèctric d'un inversor per motors Brushless [1] [8]

Cal comentar que aquesta classe d'inversors requereixen únicament 3 senyals principals provinents de la CPU (V1, V2 i V3) ja que durant el 99% del temps la seva parella (V1*, V2* i V3*) serà simplement la senyal negada (es pot integrar amb una porta NOT). Desgraciadament aquest disseny, que redueix en un 50% les sortides de la CPU necessàries, pot arribar a causar problemes, tals com pics d'intensitat, durant cada canvi d'estat degut a que durant un instant les 2 portes estan obertes a la vegada lo qual crea un curtcircuit. La millor solució a aquest problema és, un cop generada la 2a senyal amb una porta NOT, utilitzar algun xip que generi

un petit retard en els flancs de pujada de totes 2 sortides (donar temps a que l'altre es tanqui). Cal comentar que aquest circuit s'haurà d'integrar en les 5 articulacions de forma independent.

En quant als circuits que adaptant les sortides dels sensors que components poden fer falta depenen enormement del hardware seleccionat. Hi ha 3 etapes que poden o no ser necessàries:

- 1) Un convertidor corrent – tensió. Només es necessita si la sortida del sensor és una intensitat.
- 2) Un amplificador operacional que generi el guany necessari perquè el voltatge estigui en el rang de voltatge òptim d'aproximadament 0 a 3 volts. Aquesta amplificació (o reducció) addicionalment pot contenir una inversió (passar de negatiu a positiu) en cas que l'entrada lo requereixi, per exemple perquè ha passat per un convertidor corrent – tensió inversor.
- 3) Parella de multiplexors que connecti un nombre concret de valors (sortides de les etapes anteriors) a les entrades ADC del microcontrolador. Això únicament és necessari si el nombre d'entrades ADC és inferior a al nombre de sensors que s'estan monitoritzant. Suposant que cada sensor té una sola sortida únicament cal 10 entrades ADC (5 de posició i 5 de parell motor).

Un esquema elèctric d'aquestes 3 fases, una darrera de l'altre, es pot observar en la fig 3.3:

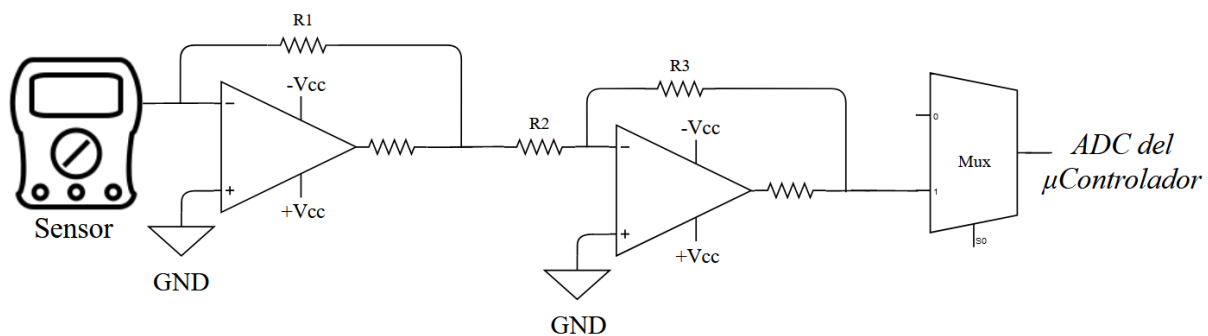


Fig. 3.3. Esquema de les 3 fases de condicionament de la senyal d'un sensor [1] [8]

Cal mencionar que en el cas d'utilitzar l'ARDUINO DUE no faria falta el multiplexor (té suficients entrades ADC), a la vegada que és molt probable que es pugi trobar sensors electrònics que ja portin incorporats algunes funcions de les primeres 2 etapes. En definitiva, aquest circuit teòric molt probablement s'haurà de revisar un cop s'hagi escollit el hardware que s'utilitzarà, sent especialment important saber els paràmetres de les mesures dels sensors.

L'últim circuit que cal la pena comentar és com es pot realitzar un descodificador analògic de les senyals que els sensors de posició del rotor del motor per tal d'enviar les senyals PWM únicament a un parell de transistors (generar les consignes V1, V1*, V2, V2*, V3 i V3* de la figura 3.2). Un circuit de com generar la logica per simular un motor BLDC en PSIM és:

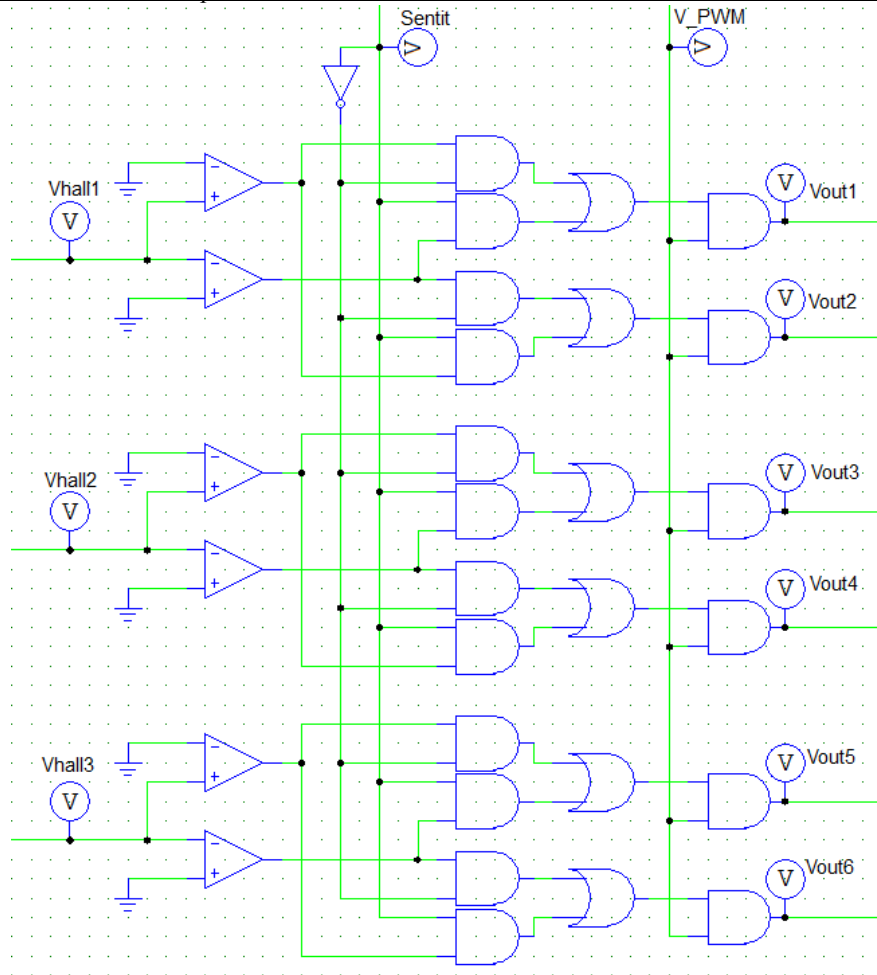


Fig. 3.4. Esquema del circuit analògic per decodificar sensors HALL en PSIM [1] [10]

Cal comentar que aquest circuit suposa que el voltatge V_{hall} pot tenir 3 valors (-1, 0 o 1), sent que la sèrie de valors que tindrà cada línia al llarg dels 6 estats seran $Hall = \{1, 1, 0, -1, -1, 0\}$. Aquesta classe de sortida no és gaire habitual (simplement és la que utilitza el model del PSIM utilitzat per unes proves inicials). En l'apartat 3.4 es podran observar circuits "equivalent" quan la lògica dels sensors de HALL és $Hall[3] = \{100, 110, 010, 011, 001, 101\}$ (valors bastant més típics en sensors comercials).

Finalment cal comentar que, tal com es sol fer sempre, tots els motors tindran una parada d'emergència, que desconnectarà de forma temporal la font d'alimentació. El circuit de la parada d'emergència serà un simple polsador reglamentari que desconnectarà el cablejat que surt directament del generador de corrent continu que alimenta tot el circuit de potència.

3.4. Circuit electrònic utilitzat.

Un cop es decideix utilitzar un motor BLDC, el diagrama general de quins components electrònics cal utilitzar generalment ha de tenir una forma similar al següent:

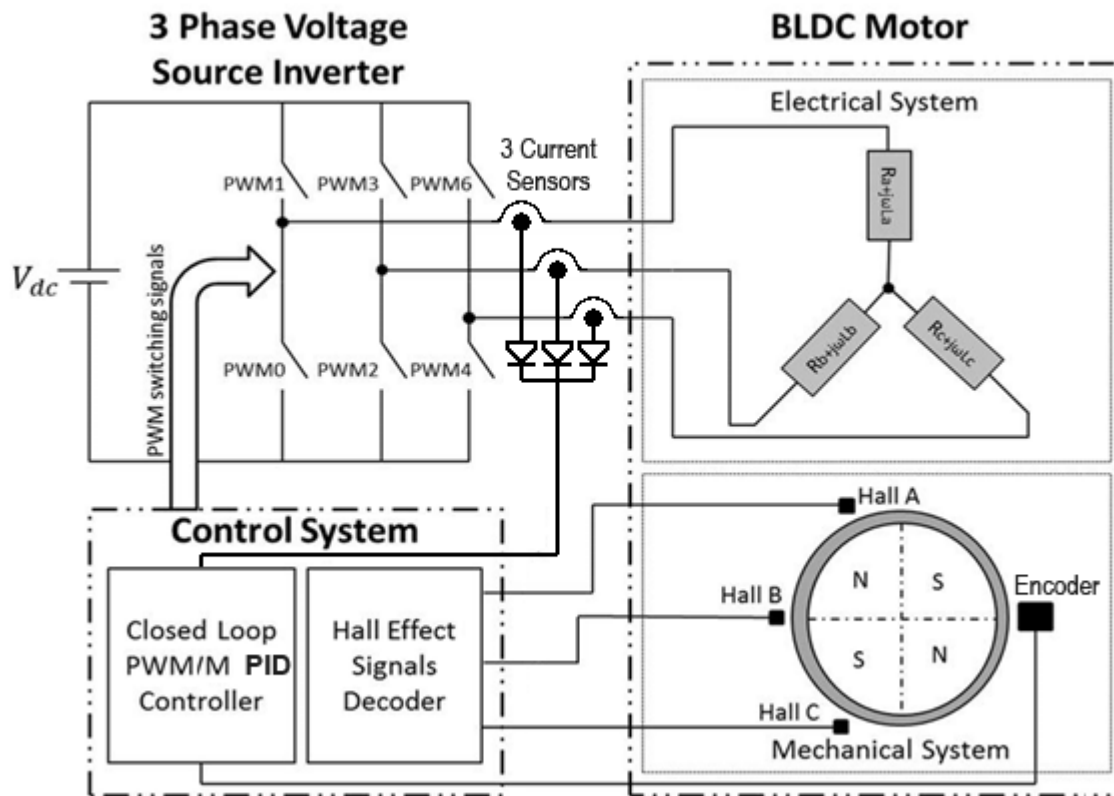


Fig. 3.5. Components d'un circuit de control d'un motor BLDC [6]

En l'imatge anterior es poden observar 6 parts principals:

1) Inversor “full-bridge” trifasic: Qualsevol motor BLDC sempre necessitarà el mateix tipus de pont trifasic, on les úniques diferències potencials són si s'utilitza transistors MOSFET o IGBT, juntament amb elements de seguretat (resistències que protegeixen el circuit de baixa potència en cas de fuges de corrent).

2) Sensors de corrent: S'ha connectat 3 sensors d'intensitat en la sortida del inversor, la sortida d'aquests sensors està connectada mitjançant 3 díodes. Aquests díodes simplement s'encarreguen de que únicament el valor positiu màxim s'envii cap el circuit de control. Cal comentar que al aplicar un control trapezoidal en tot moment hi haurà 1 intensitat positiva, 1 intensitat negativa idèntica ($I_{entrada} = -I_{sortida}$) i una 3a intensitat igual a 0 (circuit obert), de manera que els díodes únicament estan enviant el “valor absolut” de les 2 intensitats a mesurar.

3) Motor BLDC: Independentment de si es un motor estrella o triangle, el control a utilitzar ha de ser el mateix. Únicament canvia una mica els paràmetres elèctrics dins del motor.

4) Sensors del motor BLDC: La majoria de motors BLDC ja porten incorporats sensors HALL que permeten identificar en quin sector del “hexagon” (6 regions de 60° que divideixen la

circunferència de 360°) està el rotor. A aquests sensors se li ha d'agregar un encoder incorporat-lo al final de cada eix. Cal recordar que la mesura dels HALL és la posició del rotor abans de qualsevol classe de caixa de canvis, mentre que l'encoder és una mesura (amb una precisió bastant superior) de la posició de l'eix després de passar per la caixa de canvis.

El sensor utilitzats (que solen venir incorporats en el motor) per mesurar en quin dels 6 sectors del "hexagon" està el rotor pot modificar lleugerament com cal descodificar els diferents bits d'informació, però lo més comú és que hi hagin 3 sensors HALL desplaçats $\pm 120^\circ$ (dispersos al llarg de la circumferència) o, tal com es mostra en la figura anterior, desplaçats $\pm 60^\circ$ (agrupats en un terç de la circumferència). Independentment de la localització, el sensor estarà ON en un rang de $\pm 90^\circ$ (mitja circumferència centrada en el sensor). Una circumferència amb els valors que els sensors HALL assignen a cada sector, depenent de quina de les 2 configuracions anteriorment mencionades està utilitzant, es pot observar a continuació:

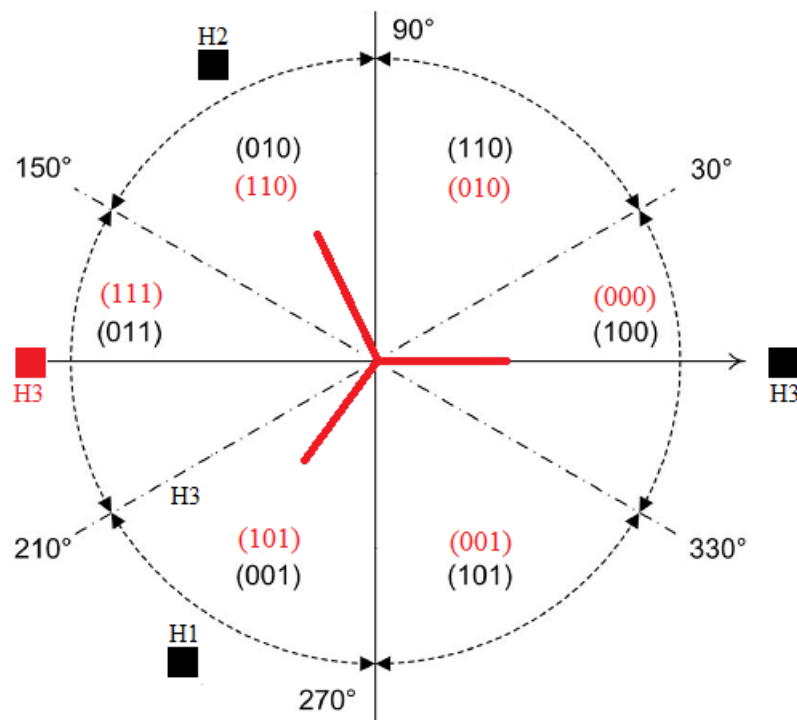


Fig. 3.6. Valors que 2 grups de sensors HALL mostren al llarg d'una revolució [1] [7]

5) Controlador digital PID: Part principal del programa informàtic que el microcontrolador ha de realitzar. De forma molt resumida, aquest programa informàtic fa un mostreig periòdic on s'obté el valor actual del sensor de corrent i de l'encoder, per tal d'utilitzar-los depenent de en quin aparell està el circuit. El manipulador envia consignes de posició i rep consignes d'intensitat, les quals es resten al valor del sensor d'intensitat per tal d'obtenir un "error" que s'utilitza per calcular la sortida del controlador PID, mentre que en el controlador és l'oposat.

El resultat d'aquest PID és un “percentatge” que s'utilitza com a duty cycle de la senyal PWM (indica el % de potencia que cal aplicar al motor) i un bit “dir” que indica el sentit de gir que cal aplicar (la majoria d'articulacions tenen un rang maxím de 180°).

6) Descodificador dels sensors d'efecte HALL: Un cop és té la senyal PWM (% de potencia) cal saber quins transistors cal accionar. Els 3 bits d'informació generada per els sensors HALL i el bit “dir” (sentit de gir) són suficients per generar 12 estats potencials, on sempre s'intentarà introduir una corrent electrica (camp magnetic) a $\pm 90^\circ$ del sector mesurat pels sensors HALL.

En els següents subcapítols es podrà observar detalls importants de cada part del circuit. Cal mencionar que el circuit que s'exposarà haurà de ser replicat 10 vegades (totes les articulacions del manipulador i del controlador tindran un circuit idèntic exceptuant els parametres del PID):

3.4.1. Inversor trifàsic i feedback de corrent.

El disseny del inversor trifàsic (el full-bridge de 3 branques) és pràcticament idèntic al mostrat en la fig. 3.2, l'únic canvi significatiu ha sigut introduir els 3 sensors d'intensitat a la sortida del mateix i el pulsador (normalment tancat) de parada d'emergència. Cal comentar que, si el generador d'alta tensió utilitzat per subministrar energia al circuit de potencia té una potencia maxima suficientment elevada, els 5 inversors es connectaran entre si tal com s'ha marcat a l'esquerra de la següent figura (el polsador de parada d'emergencia afectaria a tots ells):

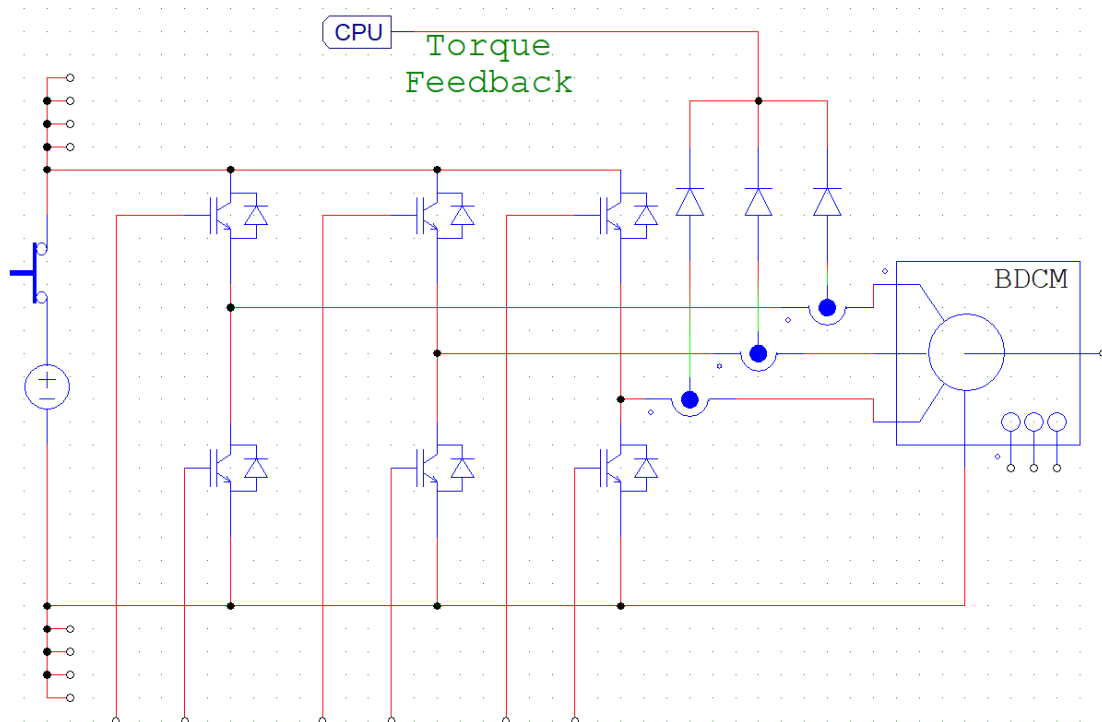


Fig. 3.7. Circuit electric del circuit de potencia (inversor trifàsic) [1] [10]

3.4.2. Motor i sensors de posició.

Els motors són un dels pocs components que s'han comprat ja prefabricats, a l'hora de seleccionar motor l'única cosa relevant per el disseny electronic és que continguin 3 sensors d'efecte HALL per poder realitzar un "sensored controll" (el motor seleccionat en té 3 d'integrats).

Mentre que els sensors HALL no requereixen cap classe de disseny (unicament cal connectar 3 cables a les sortides pertinents del motor), si que fa falta pensar com s'integrarà el sensor de posició i el seu control. Lo normal és decidir si es prefereix un encoder relatiu (crear una pols al passar punts determinats), un encoder absolut analogic (la tensió o corrent augmenta linealment de forma continua) o un encoder digital (N cables en paral·lel depenent de la precissió del encoder).

Degut a que el encoder absolut digital requeriria un gran nombre d'entrades digitals (es necessiten com a mínim 9 bits per tenir una precisió inferior a 1°), les alternatives plantejades eren un encoder absolut analogic, el problema principal del qual és llegir consignes lleugerament erronies (soroll i/o caigudes de tensió), o un encoder relatiu, el problema principal del qual és que complica lleugerament el control a realitzar (cal integrar un comptador en el microcontrolador i assegurar que no es saltarà cap pols). L'elecció final ha sigut el encoder relatiu, ja que en aquest cas no hauria de tenir gaires problemes (el microcontrolador hauria de poder gestionar el comptador mitjançant interrupcions).

Finalment, cal comentar que si durant l'etapa de testeig es veiés que s'ha de passar d'un encoder relatiu a un absolut digital, per evitar alguna classe de problema, unicament faria falta desmuntar l'aparell (no hauria de ser gaire difícil degut a com s'ha fet el disseny mecanic) i modificar un parell de linies de codi del script que s'executa en els microcontroladors.

3.4.3. Controlador digital PID.

El PID és l'única part que si o si fa falta realitzar digitalment, ja que les seves dades un cop mostregades s'hauran d'enviar cap a l'altre CPU en forma de consignes, a més a més de ser enregistrats en servidors per posteriorment fer-li un tractament de Big Data., Ja que no hi ha un circuit electronic com a tal, unicament fa falta comentar que les seves entrades seran:

- Valor analogic del feedback de corrent (intensitat "maxima"): Unica entrada analogica.

- Els 2 cables del encoder digital: El 1r s'utilitzara per crear rutines d'interrupcions (actualitzar el contador), mentre que l'estat del 2n (ja estava a 1 o encara esta a 0) indica el sentit de gir utilitzat per saber si cal augmentar o disminuir el contador.

- Una connexió en sèrie (mitjançant un USB) que connecta les 2 CPUs: Addicionalment faria falta una 2a connexió que envii la informació pertinent cap al servidor de Big Data (els detalls d'aquest servidor estan fora de l'abast d'aquest informe).

Tal com es comentarà en el proxím subapartat, les seves sortides depenent de com s'implement la descodificació dels sensors HALL (passar de 3 bits de posició a 6 possibles sortides PWM).

3.4.4. Descodificació dels sensors HALL.

Ja que s'està utilitzant un algoritme de control trapezoidal en tot moment s'activarà únicament 1 transistor superior (S1, S3 o S5) i 1 transistor inferior (S2, S4 o S6) que mai formarà part de la mateixa branca (no hi haura curtcircuit fins i tot durant un període de transició d'un estat a un altre). A efectes pràctics s'ha de fer una "AND" entre la senyal PWM i el "bit" que indica si una sortida ha de ser accionada o no.

Mentre que gran part del sistema oviament requereixen hardware (cada un sera un subcircuit analògic) i el PID s'ha de fer via software (s'ha de fer càlculs i enviar dades a l'altre CPU), la descodificació dels HALL teòricament podrien fer-se de les 2 formes. Encara que el pla original és controlar-lo de forma completament digital conjuntament amb el PID, hi ha 2 alternatives analògiques que s'han dissenyat com a "pla B" en cas que la solució completament digital provoqui problemes a l'hora d'implementar el sistema.

Alternativa 1: El controlador té una sortida PWM (indica el % de potencia) i un simple bit de direcció (indica si cal invertir el sentit del corrent girar de forma antihorària), els sensors HALL s'implementen de forma totalment analògica. Aquesta és la solució més simple ja que no caldria utilitzar els 3 sensors d'efecte HALL com a entrades del microcontrolador, simplement s'utilitzarien en un circuit analògic, que incorporaria les sortides del microcontrolador.

L'idea és tenir un desmultiplexor de 4 bits on l'entrada és la senyal PWM del microcontrolador, mentre que els 4 bits que decideixen quina de les 16 sortides connectar a l'entrada serien els 3 bits dels sensors HALL (l'ordre importa), juntament amb el bit de direcció (bit de major pes). Les 16 sortides del DEMUX es connectarien als 6 pins de sortida (les consignes dels 6

transistors del inversor trifasic), mitjançant 6 portes OR de 4 entrades, seguint la següent taula de la veritat (groc és una transistor que cal activar amb el PWM i gris és PWM que s'ha de mantenir tancat per evitar curtcircuit amb el transistor que si s'ha d'obrir):

Out	DIR	HALL C	HALL B	HALL A	UP 1	DOWN 1	UP 2	DOWN 2	UP 3	DOWN 3
0	0	0	0	0	Avisar que ha succeït un error + Tot a zero					
1	0	0	0	1	0	0	0	1	1	0
2	0	0	1	0	0	1	1	0	0	0
3	0	0	1	1	0	1	0	0	1	0
4	0	1	0	0	1	0	0	0	0	1
5	0	1	0	1	1	0	0	1	0	0
6	0	1	1	0	0	0	1	0	0	1
7	0	1	1	1	Avisar que ha succeït un error + Tot a zero					
8	1	0	0	0	Avisar que ha succeït un error + Tot a zero					
9	1	0	0	1	0	0	1	0	0	1
10	1	0	1	0	1	0	0	1	0	0
11	1	0	1	1	1	0	0	0	0	1
12	1	1	0	0	0	1	0	0	1	0
13	1	1	0	1	0	1	1	0	0	0
14	1	1	1	0	0	0	0	1	1	0
15	1	1	1	1	Avisar que ha succeït un error + Tot a zero					

Taula 3.1. Taula de la veritat per descodificar els 16 estats de 3 sensor HALL [1] [6]

Cal comentar que al canviar el bit de direcció les caselles grogues i grises s'inverteixen (per canviar de sentit de gir cal invertir la polaritat del corrent). Sent l'únic instant on si es podria tenir un petit pic de tensió (curtcircuit temporal). Un circuit electronic per implementar aquesta taula de la veritat mitjançant portes OR de 2 entrades (cada trio de portes OR de 2 entrades es pot substituir per una porta OR de 4 entrades) es pot observar a continuació:

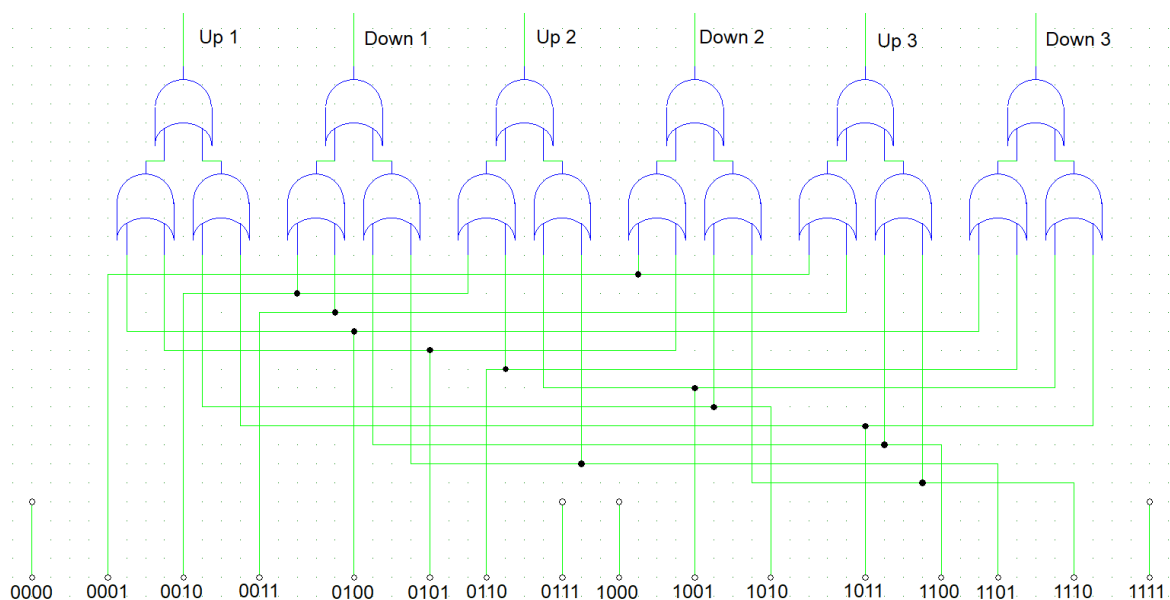


Fig. 3.8. Circuit utilitzat per implementar la logica de control de la Taula 3.1 [1] [10]

Aquesta alternativa únicament requereix 1 sortida PWM i 1 sortida digital per motor, a lo qual s'ha d'agregar les 2 entrades del encoder i l'entrada del sensor de corrent. Dit d'una altre manera si s'utilitza un únic microcontrolador per gestionar els 5 motors del controlador o manipulador faria falta 5 sortides PWM, 5 sortides digitals, 10 entrades digitals (en total 20 pins digitals, 5 dels quals han de poder generar senyals PWM) i 5 entrades analogiques. La majoria de microcontroladors de la família Arduino disposen d'aquest nombre pins (l'únic model relativament popular que no conté tants pins digitals és l'Arduino UNO).

Alternativa 2: De forma similar a l'alternativa anterior, el controlador PID té 2 sortides (la senyal PWM i el bit de direcció), la diferencia principal és que s'ha d'agregar 3 entrades i 3 sortides digitals addicionals. Basicament es digitalitza els 3 bits d'informació dels sensors HALL, es comprova si tots ells formen un dels estats prohibits (X000 o X111) i en cas negatiu s'actualitzaria les 3 sortides addicionals amb el valor de les 3 entrades addicionals.

Aquest pas intermitg permet modificar els 3 bits que entren al DEMUX en cas que es consideri oportu. Per exemple, si es creu que l'estat 000 pot succeir a baixes velocitats (inici del moviment o quan està apunt d'aturar-se) es pot programar que mantingui la sortida anterior (un dels 6 estats "legals") fins que les entrades deixin de ser 000 (mantenir un registre del ultim sector "correcte" mostreigat). Les connexions del DEMUX i la taula de la veritat de les seves sortides (i per tant el circuit de les portes OR) segueixen sent iguals, ja que l'únic canvi és que el microcontrolador en ocasions modificarà l'entrada del DEMUX (mantenir estat anterior).

Aquesta alternativa té el problema de requerir una mateixa invesió de hardware (circuit del DEMUX i portes OR) i ocupar 6 pins digitals addicionals respecte l'alternativa 1. Dit d'una altre manera si s'utilitza un únic microcontrolador per gestionar els 5 motors del controlador o manipulador faria falta 5 sortides PWM, 20 sortides digitals, 25 entrades digitals (en total 50 pins digitals, 5 dels quals han de poder generar senyals PWM) i 5 entrades analogiques. Degut a aixó s'hauria de dividir el control digital en diversos microcontroladors independents (requereixen masses pins) o utilitzar una Arduino DUE o MEGA (les uniques que plaques de la família Arduino que tenen 50 o més pins digitals).

A canvi d'aquets inconvenients, aquesta alternativa permet un control total mitjançant software de la relació entre sortida dels sensors HALL i entrada del DEMUX, lo qual permet fixar temporalment les entrades (ignorar canvi a algun estat "incorrecte) o modificar quin nombre del MUX s'assigna a cada conuinació de valors dels sensors HALL. Aixó ultim és molt util si

s'ha de canviar el motor i/o els seus sensors d'efecte HALL i el nou conjunt genera sortides lleugerament diferents, ja que permet modificar el software sense tenir que modificar el circuit analògic en si (menys cost monetari).

Alternativa 3 (la utilitzada): L'alternativa elegida és realitzar totes les decisions dins del microcontrolador i generar directament 6 senyals PWM en els pins del microcontrolador, per tal de que únicament s'hagi de fer una connexió directe entre el microcontrolador i el inversor.

El software que possible aquest control s'ha explicat en l'apartat 5.2.2. En quant a hardware (circuit electrònic) únicament fa falta saber que aquesta alternativa, a més de les 2 entrades digitals del encoder i l'entrada analògica del sensor de corrent, requereix de 3 entrades digitals pels sensors HALL i 6 sortides PWM per comandar els transistors de l'inversor (s'ha eliminat totes les sortides digitals). Per tant si es volgues utilitzar un microcontrolador per gestionar els 5 motors, faria falta 5 entrades analògiques, 25 entrades digitals i 30 sortides PWM (en total 60 pins digitals, 25 dels quals han de poder generar senyals PWM).

Cal mencionar que el gran nombre de pins PWM que fan falta per implementar aquest algoritme de control ha sigut una de les raons principals per dividir el control digital en diversos microcontroladors. Altres raons és simplificar el process d'agregar motors addicionals (ma del robot) i assegurar que es podra complir les especificacions temporals (menys calculs / segon).

En definitiva, s'utilitzarà un microcontrolador per controlar cada motor (si es veu que tot funciona de forma optima es pot plantejar passar a 1 controlador general mitjançant alguna de les 2 alternatives) i el "circuit electrònic" que descodifica els sensors HALL serà simplement connectar la sortida dels 3 sensors a 3 pins digitals (3 entrades del microcontrolador) i les 6 entrades de control de l'inveror a 6 pins digitals amb capacitat per crear ones PWM. Un dels millors microcontroladors per aquest nombre de pins, tenint en compte que cal agregar-li els 2 pins digitals de l'encoder i l'entrada analògica del sensor d'intenistat, és l'Arduino UNO.

4. Disseny de l'electrònica de l'aparell de control.

4.1. Diferències entre el manipulador i el seu controlador.

Mentre que optimitzar al màxim la carrega que el manipulador pot moure (augmentar la potència dels actuadors) sol ser una de les prioritats més importants d'un robot, en el cas de l'aparell de control la mida és una prioritat molt més important. No només pot arribar a ser contraproduent augmentar el parell màxim de les articulacions, ja que obliguen a aplicar més força al controlador per tal de moure'l, sinó que a l'estar unit al braç de l'usuari el pes, la mida i l'ergonomia són unes preocupacions molt més importants.

Degut a això, les alternatives hidràuliques queden pràcticament descartades ja que no és necessari el parell extra i tant el pes com la mida augmentarien de forma innecessària. Depenent de quin parell màxim es desitgi les millors alternatives serien un brushless, tal com s'ha comentat en el capítol anterior, o uns servomotors si no es vol tenir uns parells gaire elevats en el feedback de força (força que cada articulació genera al braç de l'usuari). En aquest disseny s'ha decidit utilitzar simplement motors brushless.

Un cop es té clar això, l'única diferència substancial a l'hora de dissenyar els circuits elèctrics del controlador respecte els del manipulador és que els actuadors estaran dispersos de manera una mica més oberta que en un manipulador (no fa falta tancar-los completament ja que sempre estaran separats del braç humà) amb lo qual els cables s'hauran d'agrupar de forma correcta per evitar que es trenquin. A més a més degut a voler minimitzar espai ocupat per el controlador, gran part dels circuits, plaques i xips molt probablement es col·loquin tan a prop de l'espalla com sigui possible per evitar que molestin o s'integraran dins del suport dels diferents motors.

Com a norma general, el disseny electrònic en si seria completament igual, variant únicament on es col·loquen els sensors (alguns fa falta col·locar-los de forma indirecta) i potencialment que classe de driver requerirà els actuadors en cas de no utilitzar els mateixos del manipulador (al ser exactament el mateix model de motors BLDC el circuit serà casi idèntic).

4.2. Calcul de la traducció de consignes.

Tal com s'explica en major detall en l'altre informe (el basat principalment en el disseny mecànic), degut a com s'ha realitzat el disseny geomètric del manipulador i el controlador era

important començar un procés per trobar unes equacions que permetessin traduir les consignes informàtiques de l'espatlla d'un aparell en consignes equivalents de l'altre aparell.

De forma molt resumida, tal com es comenta en l'informe del disseny mecànic, fa falta realitzar un càlcul que depen directament de l'estructura final de tant el manipulador com controlador. Concretament fa falta una matriu T del manipulador i una matriu T del controlador que:

- Comencin en un sistema de coordenades equivalent (mateixes direccions X, Y i Z).
- Continguin com a variable únicament les articulacions A1 i A2 del seu aparell.
- Acabin en un sistema de coordenades equivalents si es compleix alguna condició.

La següent figura mostra una comparació de com evolucionen els sistemes de coordenades (com giren els eixos) al llarg dels punts relacionats amb el gir de l'espatlla:

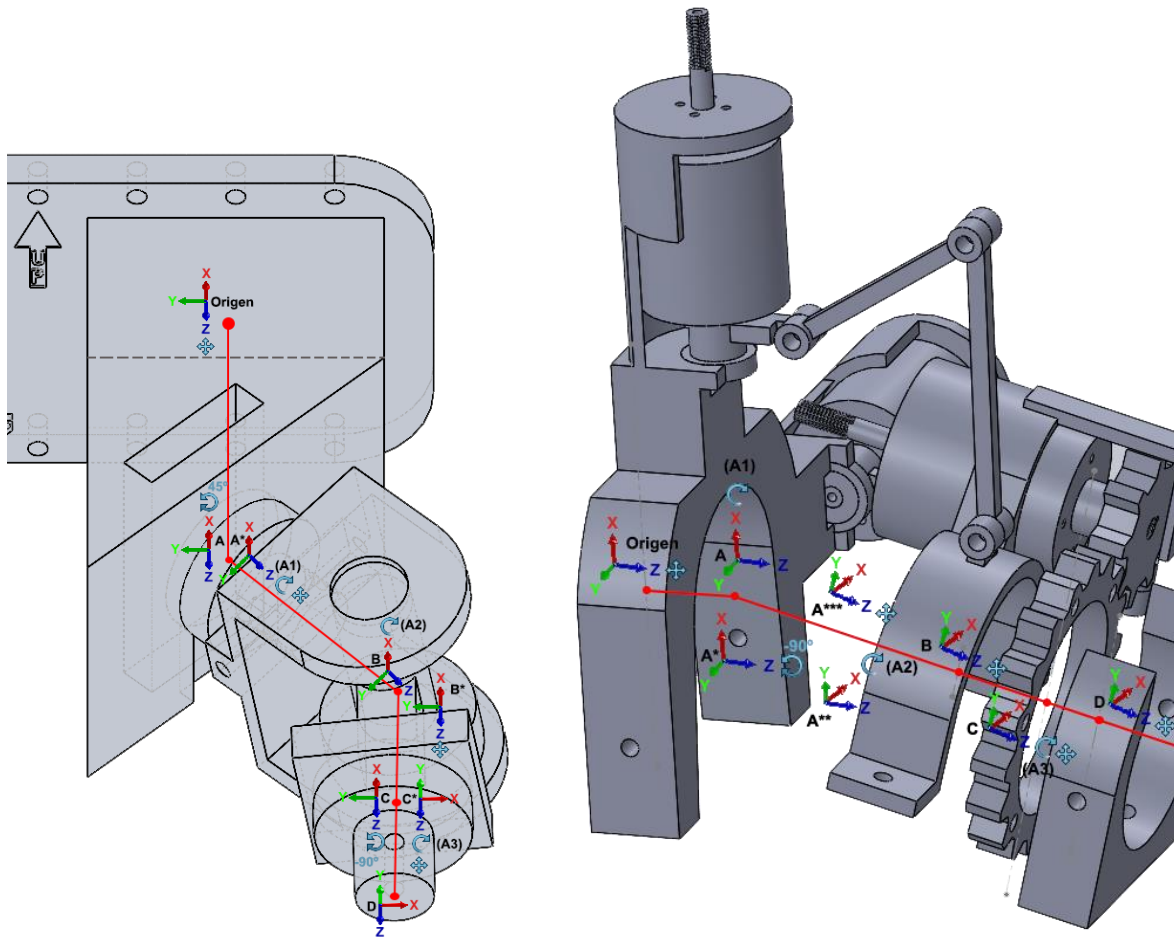


Fig. 4.1. Comparació de l'evolució dels sistemes de coordenades [1] [11] [15]

Ho primer que es pot veure és que si tots els eixos del controlador estan a 0° , tots els punts s'alinearàn en la direcció Z original. Per aconseguir algo similar en el manipulador s'ha de posar l'eix A2 en una posició de -45° (motiu per el qual la taula 3.1 tenia un "A2 - 45° ").

Un cop definit els angles "0°" del manipulador (els que aconseguixen que gran part del robot s'alinei amb la posició Z de l'origen), toca fer una calculadora que transformi els parametres de Denavit-Hartenberg en matrius 4x4. S'ha utilitzat tant MatLab com Excel per comprovar que la metodologia no afecta al resultat. Els resultats (i forma de calcular-los) són:



Fig. 4.2. Resultats fixos calculats mitjançant Excel (posició de repos) [1] [12]

```

1  A1m = 0
2  A2m = 0
3
4  A1c = 0
5  A2c = 0
6  L1 = 129
7
8  M12=MatriuT(0,0,0,45)
9  M23=MatriuT(A1m,120,0,0)
10 M34=MatriuT(0,0,0,A2m-45)
11 M45=MatriuT(0,80,0,0)
12 M56=MatriuT(-90,0,0,0)
13
14 C12=MatriuT(0,0,0,A1c)
15 C23=MatriuT(-90,0,0,0)
16 C34=MatriuT(0,0,0,A2c)
17 C45=MatriuT(0,L1,0,0)
18 C56=MatriuT(0,51.04,0,0)
19
20 M16=M12*M23*M34*M45*M56
21 C16=C12*C23*C34*C45*C56
22
23 function T=MatriuT(theta,d,a,alpha)
24 T=[cos(theta*pi()/180) -sin(theta*pi()/180) 0 a;
25   sin(theta*pi()/180)*cos(alpha*pi()/180) cos(theta*pi()/180)*cos(alpha*pi()/180) -sin(alpha*pi()/180) -sin(alpha*pi()/180)*d;
26   sin(theta*pi()/180)*sin(alpha*pi()/180) cos(theta*pi()/180)*sin(alpha*pi()/180) cos(alpha*pi()/180) cos(alpha*pi()/180)*d;
27   0 0 0 1];
28 end

```

Fig. 4.3. Resultats fixos calculats mitjançant MatLab (Posició de repos) [1] [13]

Un cop comprovat que les totes les matrius (sent especialment crítiques la M16 i C16) són identiques es pot modificar el script de MatLab per obtenir una equació on A1m, A2m, A1c i A2m són variables de tipus “sym”. El nou codi que incorpora les variables simbòliques és:

```

1  A1m = sym('A1m')
2  A2m = sym('A2m')
3
4  A1c = sym('A1c')
5  A2c = sym('A2c')
6  L1 = 129
7
8  M12=MatriuT(0,0,0,45)
9  M23=MatriuT(A1m,120,0,0)
10 M34=MatriuT(0,0,0,A2m-45)
11 M45=MatriuT(0,80,0,0)
12 M56=MatriuT(-90,0,0,0)
13
14 C12=MatriuT(0,0,0,A1c)
15 C23=MatriuT(-90,0,0,0)
16 C34=MatriuT(0,0,0,A2c)
17 C45=MatriuT(0,L1,0,0)
18 C56=MatriuT(0,51.04,0,0)
19
20 M16=M12*M23*M34*M45*M56
21 C16=C12*C23*C34*C45*C56
22
23 EspatllaManipulador=M16(1:3,1:3)
24 EspatllaControlador=C16(1:3,1:3)
25
26 function T=MatriuT(theta,d,a,alpha)
27 T=[cos(theta*pi()/180) -sin(theta*pi()/180) 0 a;
28   sin(theta*pi()/180)*cos(alpha*pi()/180) cos(theta*pi()/180)*cos(alpha*pi()/180) -sin(alpha*pi()/180) -sin(alpha*pi()/180)*d;
29   sin(theta*pi()/180)*sin(alpha*pi()/180) cos(theta*pi()/180)*sin(alpha*pi()/180) cos(alpha*pi()/180) cos(alpha*pi()/180)*d;
30   0 0 0 1];
31 end

```

Fig. 4.4. Codi de MatLab generalitzat [1] [13]

Idealment, arribat aquest punt simplement faria falta formular el següent sistema d'aquest estil on simplement es realitzi una igualtat entre les fileres de la matriu (cal igualar cada filera per aconseguir una mateixa orientació en cada eix de coordenades):

$$\begin{aligned}
 M16(1,1) + M16(1,2) + M16(1,3) &= C16(1,1) + C16(1,2) + C16(1,3) \\
 M16(2,1) + M16(2,2) + M16(2,3) &= C16(2,1) + C16(2,2) + C16(2,3) \\
 M16(3,1) + M16(3,2) + M16(3,3) &= C16(3,1) + C16(3,2) + C16(3,3)
 \end{aligned} \quad (4.1)$$

Desgraciadament, ja que s'estan multiplicant diverses matrius relativament complexes, gran part de les caselles de les 2 matrius contenen expressions molt llargues i difícils de resoldre.

La regió 3x3 important (matriu de rotació) de les matrius T resultants resumides (patrons repetitius s'han assignat a noves variables que cal calcular) es poden observar a continuació:

EspatllaManipulador =

$$\begin{pmatrix} \frac{4967757600021511 \sigma_8}{81129638414606681695789005144064} + \sigma_1 \cos(\sigma_9) & \sigma_8 - \frac{4967757600021511 \sigma_1 \cos(\sigma_9)}{81129638414606681695789005144064} & \sigma_1 \sin(\sigma_9) \\ \frac{4967757600021511 \sqrt{2} \sigma_1}{162259276829213363391578010288128} + \sigma_6 - \sigma_5 & \frac{\sqrt{2} \sigma_1}{2} - \sigma_4 + \sigma_3 & -\sigma_7 - \sigma_2 \\ \frac{4967757600021511 \sqrt{2} \sigma_1}{162259276829213363391578010288128} - \sigma_6 - \sigma_5 & \frac{\sqrt{2} \sigma_1}{2} + \sigma_4 + \sigma_3 & \sigma_7 - \sigma_2 \end{pmatrix}$$

where

$$\sigma_1 = \sin\left(\frac{\pi \text{ A1m}}{180}\right)$$

$$\sigma_2 = \frac{\sqrt{2} \sigma_8 \sin(\sigma_9)}{2}$$

$$\sigma_3 = \frac{4967757600021511 \sqrt{2} \sigma_8 \cos(\sigma_9)}{162259276829213363391578010288128}$$

$$\sigma_4 = \frac{4967757600021511 \sqrt{2} \sin(\sigma_9)}{162259276829213363391578010288128}$$

$$\sigma_5 = \frac{\sqrt{2} \sigma_8 \cos(\sigma_9)}{2}$$

$$\sigma_6 = \frac{\sqrt{2} \sin(\sigma_9)}{2}$$

$$\sigma_7 = \frac{\sqrt{2} \cos(\sigma_9)}{2}$$

$$\sigma_8 = \cos\left(\frac{\pi \text{ A1m}}{180}\right)$$

$$\sigma_9 = \frac{\pi (\text{A2m} - 45)}{180}$$

Fig. 4.5. Matriu de rotació “resumida” de l'espalla del manipulador [1] [13]

EspatllaControlador =

$$\begin{pmatrix} \frac{4967757600021511}{81129638414606681695789005144064} & \sigma_3 & -\sigma_1 \\ -\sigma_4 & \frac{4967757600021511 \sigma_4 \sigma_3}{81129638414606681695789005144064} - \sigma_2 \sigma_1 & -\frac{4967757600021511 \sigma_4 \sigma_1}{81129638414606681695789005144064} - \sigma_3 \sigma_2 \\ -\sigma_2 & \sigma_4 \sigma_1 + \frac{4967757600021511 \sigma_3 \sigma_2}{81129638414606681695789005144064} & \sigma_4 \sigma_3 - \frac{4967757600021511 \sigma_2 \sigma_1}{81129638414606681695789005144064} \end{pmatrix}$$

where

$$\sigma_1 = \sin\left(\frac{\pi A2c}{180}\right)$$

$$\sigma_2 = \sin\left(\frac{\pi A1c}{180}\right)$$

$$\sigma_3 = \cos\left(\frac{\pi A2c}{180}\right)$$

$$\sigma_4 = \cos\left(\frac{\pi A1c}{180}\right)$$

Fig. 4.6. Matriu de rotació “resumida” de l'espatlla del manipulador [1] [13]

Degut a l'enorme complexitat d'aquesta tasca, juntament al fet de que el controlador encara esta en fase de desenvolupament (la seva matriu de gir possiblement acabarà sent modificada), es va decidir deixar fora de l'abast d'aquest projecte la resolució total d'aquest calcul. Dit d'una altre manera, en aquest TFG únicament s'arribarà al calcul d'aquestes 2 matrius.

Per tant, per tal de resoldre aquest sistema d'equacions i implementar el resultat com una funció de python en el codi de les CPU (Raspberrys) farà falta esperar a un projecte posterior. Aquesta falta de resolució se “solucionarà” col·locant una línia de codi en la programació de la CPU que fagi una crida a una funció que, per motius evidents, encara no s'haurà dissenyat.

5. Disseny del feedback de control.

5.1. Disseny lògic del feedback a utilitzar.

Un dels primers apartats que s'han dissenyat ha sigut la lògica que utilitzaria el feedback del sistema que s'està dissenyant. Saber que cal mesurar i on s'utilitzarà és elemental per saber quina classe de disseny i/o programació s'utilitzarà. En aquest cas s'ha decidit utilitzar un flowchart dividit en les diferents etapes que es poden identificar al llarg del sistema:

- Interacció del controlador amb el mon físic (usuari).
- Circuit electrònic (analògic) dels drivers i sensors del controlador.
- Programació de la CPU del controlador (operacions i/o càlculs informàtics).
- Programació de la CPU del manipulador en si (operacions i/o càlculs informàtics).
- Circuit electrònic (analògic) dels drivers i sensors del manipulador en si.
- Interacció del manipulador amb el mon físic (accions del braç robòtic).

Adicionalment s'ha pintat cada bloc depenen de a quin subcircuit formen part (quina és la seva funció principal):

- Blanc: Formen part de 2 o més subcircuits (color genèric).
- Taronja: Sistemes de parada d'emergència. Tallar el flux d'electricitat per aturar-ho tot.
- Blau: Circuit de feedback que indica com el controlador s'ha de moure. Es basa en comparar els angles (posicions) del manipulador i el controlador, per tal de que el controlador imiti la posició del manipulador (aplicant una força al braç de l'usuari).
- Verd: Control del manipulador en si (braç robòtic). Es basa en utilitzar la mesura de parell que s'està aplicant cada articulació del controlador com a consignes de les articulacions equivalents del manipulador, per tant que la força que l'usuari aplica al controlador sigui la que el manipulador aplicarà al seu entorn (objectes amb els que potencialment interactuï).

La figura 5.1, que es pot observar a continuació, mostrà el flowchart que seguirà els diferents tipus d'informació (posicions, forces i consignes informàtiques) que el sistema ha de gestionar, seguint les divisions ja mencionades

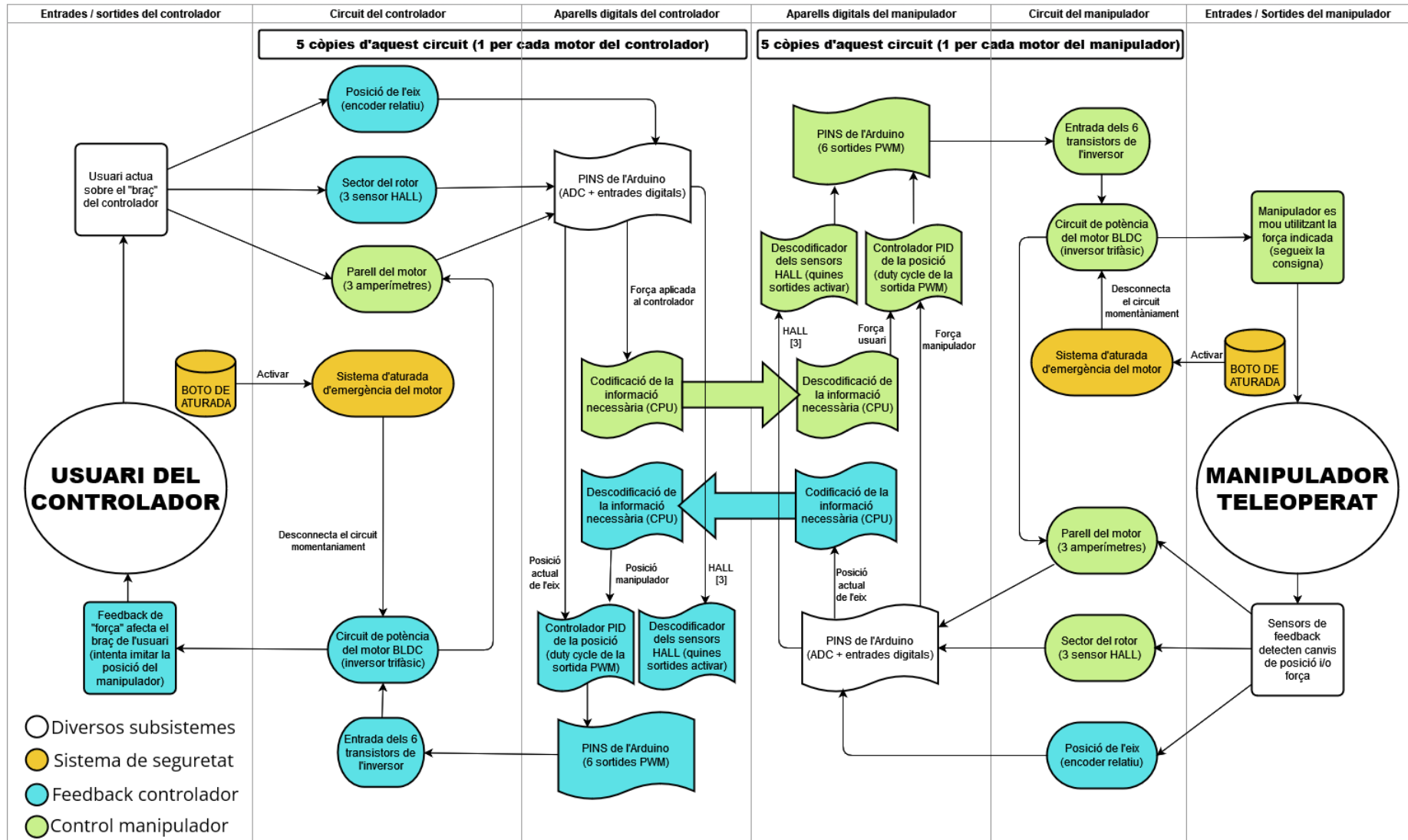


Fig..5.1. Diagrama flowchart dels diferents components del sistema [1] [8]

5.2. Programació dels aparells informatics.

Mentre que el disseny del 2n i 5e apartat de la Fig. 5.1 (els sensors i circuits dels actuadors) ja han sigut explicats en els capítols 3 i 4 respectivament, la informació relativa al 3r i 4t apartat del diagrama (la programació de tots els aparells de control digital) s'exposaran a continuació:

Originalment es volia minimitzar costos utilitzant únicament 2 microcontroladors (Arduinos) o microprocessadors (Raspberry), de manera que únicament fes falta una sola connexió entre elements de control. El problema d'aquesta classe de configuració és que pot requerir un circuit electronic més complex per tal d'evitar quedar-se sense pins d'entrada i/o de sortida. Tal com s'ha comentat en l'apartat 3.4.4, un control purament digital d'un motor requereix 6 sortides PWM, 3 entrades digitals pels sensors HALL, 2 entrades digitals pels encoders i una entrada analogica per el sensor d'intensitat (parell motor). Ja que si el control de 5 motors utilitant el metode mencionat en l'apartat 3.4.4 requereix 30 sortides PWM (el maxim d'una placa mitjanament gran, com l'Arduino DUE, solen ser uns 12 PWM), s'ha decidit dividir el control en diversos aparells.

Per aconseguir-ho, s'ha utilitzat una metodologia de IoT basada en 4 nivells diferents:

- 1) Sensors i actuadors: Circuits analogics explicats en l'apartat 3.4.
- 2) Edge Computer: Microcontroladors connectats als sensors i actuadors que fan la majoria dels calculs repetitius (controlador PID per generar la senyal PWM i seleccionar quines entrades del inversor cal activar). En total faran falta 10 Arduinos UNO (5 pel manipulador i 5 pel controlador), amb un codi practicamente identic que s'explicarà en els apartats 5.2.1 i 5.2.2.
- 3) Fog Computer: Microprocessadors (CPU) que serveixen com a capa superior del Edge Computer. La seva funció és realitzar calculs no tan repetitius o que requereixen dades de diversos motors, encarregar-se de gestionar totes les connexions en sèrie (redireccionar dades únicament al controlador digital que les necessita) i tenir una interfície amb l'usuari (mostrar errors i guardar els programes necessaris per reiniciar el sistema). S'utilitzarà 2 Raspberrys com a microprocessadors del manipulador i controlador, sent que els detalls de com s'implementaran es poden llegir en l'apartat 5.2.3.
- 4) Cloud Computing: Aquesta capa del sistema conté els servidors i ordinadors utilitzats per fer l'anàlisi de "Big Data" amb les dades que el sistema generarà. Al quedar fora de l'abast del projecte (s'ha d'implementar posteriorment) no s'explicarà en gaire detall.

5.2.1. Programació del PID del Edge Computer.

El microcontrolador del manipulador (braç robòtic) i del controlador han de realitzar funcions bastant similars (únicament varia on s'utilitzen certes variables). De forma resumida el manipulador ha de realitzar 5 procediments de forma periòdica:

1) Obtenir dades del sensor de posició (l'entrada pot variar depenent del tipus d'encoder) i del sensor de parell (via un ADC), per conèixer els valors actuals (posicions i forces) del manipulador. Cal recordar que el parell es calcularà amb un sensor d'intensitat.

2) Descodificar i llegir el conjunt de dades que s'envien des de la CPU del controlador (passant primer per la CPU pròpia), per conèixer els valors actuals de forces aplicades en el controlador.

3) Utilitzar el valor de la força actual en cada articulació del controlador per saber si cal augmentar, reduir o mantenir les consignes del manipulador. Aquestes consignes, que s'utilitzaran en el següent període, es calcularan amb un controlador digital tipus PD (un petit error estàtic en la força realitzada és acceptable) que utilitza la diferència entre els 2 valors de les forces actuals. El valor obtingut serà dividit en una variable booleana que indica el sentit de gir (negatiu = 1) i un valor absolut limitat entre 0 ~ 255 (és un percentatge on 255 és el 100%).

4) Codificar i enviar el conjunt de dades que la CPU del controlador necessita (angles actuals del manipulador). Actualment, únicament farà falta enviar els valors actuals dels angles actuals de cada articulació del manipulador (les obtingudes via el ADC 1). En el cas del controlador les dades enviades són les relatives als moments en cada articulació (els angles serveixen per el càlcul de l'error de l'apartat "3").

5) Enviar els valors de les consignes del punt 3 cap a l'exterior. El sentit pot ser una simple sortida digital, mentre que el percentatge s'utilitzarà per definir el duty cycle d'una senyal PWM i per tant requereix com a mínim 1 sortida digital-PWM. En total només faria falta utilitzar 2 sortides digitals (una normal i una que accepti senyals PWM).

Si es transforma l'algoritme anterior en algun tipus de pseudo-codi, es pot arribar a la conclusió que el codi més simple possible ha de estar dividit en 4 parts elementals:

1) Inicialització del microcontrolador: Abans de les tasques periòdiques cal declarar totes les variables a utilitzar durant el cicle del loop(), a més de configurar els registres interns del microcontrolador per tal de que entrades (ja siguin analògiques, digitals), sortides (PWM i digital amb el signe), interrupcions (ja siguin de hardware o periòdiques) i connexions en sèrie

(connexió amb la seva CPU) funcionin de forma correcte. Tot aquesta part del codi s'executarà únicament durant el “període 0” (a l'encendre el sistema).

2) Iniciar un cicle de mostreig: Cada un temps determinat (temps de mostreig del controlador digital) s'ha de poder sortir del loop on s'estava esperant. Això es pot fer iniciant una interrupció periodica (que canvi un boolea que termini el while d'espera) o fent que el loop d'espera estigui consultant constantment el rellotge intern del microcontrolador (per exemple la funció “millis()” d'Arduino permet saber els microsegons que han passat fins que s'ha reiniciat el programa). Un cop surt de l'estat “d'esperar un nou cicle”, el primer pas és llegir els valors de totes les entrades (punts 1 i 2), descodificant els valors rebuts de la CPU en cas que sigui necessari i actualitzant els valors de les variables corresponents.

La descodificació d'informació enviada per un port serie pot variar una mica depenent de que s'utilitzi i del seu llenguatge de programació, però en el cas de l'Arduino si només es vol llegir strings com si fossin valors enters (int sense decimals), únicament cal la següent línia:

```
>> AltreConsigna = Serial.readString().toInt(); //Un string sense nombres és igual a 0
```

Si el ADC està configurat de forma correcte obtenir les altres entrades simplement utilitzar la línia de codi que el manual de la placa indiqui (com a molt pot ser necessari restar un offset i multiplicar per un factor que escali el valor). Per exemple en una Arduino ADC1 podria ser:

```
>> InputA1 = (analogRead(A1)-2048)*(1/2048); //Entrada: (INPUT - OFFSET) * ESCALAR
```

3) Operacions matemàtiques: Un cop es tenen totes les dades rellevants, toca fer el 2n grup d'operacions del cicle de mostreig (calcul de la nova consigna). El càlcul del valor que la consigna ha de tenir durant el següent període es pot realitzar utilitzant les equacions 5.1 i 5.2, que es poden observar a continuació:

$$error_{Força\ manipulator}(k) = \varepsilon_{Fm}(k) = F_{controlador}(k) - F_{manipulador}(k) \quad (5.1)$$

$$Consigna_{manipulador}(k+1) = K_p * \varepsilon_{Fm}(k) + [T_d / T_{mostreig}] * [\varepsilon_{Fm}(k) - \varepsilon_{Fm}(k-5)] \quad (5.2)$$

Cal recordar que la K_p i T_d són constants que caldria optimitzar de forma experimental un cop es proves el funcionament dels nostres motors (ja sigui optimitzant o fent un calcul de MatLab mitjançant un model aproximat amb els parametres del motor).

Tal com s'ha comentat anteriorment la consigna serà dividida en un boolea (dir) que representi el signe i un percentatge en valor absolut (DutyCycle). En Arduino això es pot fer així:

```
>> dir = (Consigna_manipulador < 0) // Signe negatiu = 1
```

```
>> DutyCycle = min(abs(PID), 255) // El 255 equival a un dutyCycle del 100%
```

4) Finalment, un cop fet tots els càlculs necessaris, el final del cicle consisteix en enviar els valors a l'exterior utilitzant els pins digitals en el cas de la consigna del manipulador (senyal binaria + senyal PWM) o el bus que l'uneix a la CPU en el cas dels angles actuals de les articulacions del manipulador. Els 3 procediments es poden realitzar de la següent forma:

```
>> analogWrite(PinPWM, DutyCycle); // Senyal PWM que s'enviarà al circuit dels HALLS
```

```
>> digitalWrite(PinSigne, dir); // Bit que indica el signe
```

```
>> Serial.println(Enviar); // Enviar com a String el valor de l'angle
```

Finalment, cal comentar que degut a que el temps de mostreig òptim ve indicat per la part més lenta del sistema, al tenir interacció humana, un temps de mostreig de 5 ms (500 Hz) hauria de ser suficient. Cal recordar que baixar encara més el temps de mostreig pot arribar a causar problemes en diversos elements: La connexió en sèrie pot tenir algun element que no soporta processar tants bits / segon (caldrà baixar el baud rate, juntament amb el període de mostreig), la freqüència del les sortides PWM (la qua esta fixada i modificar-la pot provocar problemes) ha de ser superior a la freqüència de mostreig i sobretot el calcul de temps de mostreig molt petits (menys d'un milisegon) s'ha de realitzar amb la funció `micros()`.

Aixó ultim és un problema ja que si es vol mesurar el temps manualment dins del loop s'ha d'utilitzar la funció "`millis()`" (funció òptima per mostreigs de 3 ms o superiors) o la funció "`micros()`", que permet períodes 1000 vegades més petits, a canvi de que sempre tindrà un overflow al cap de poc més de 70 minuts (`millis` tardà 50 dies al tenir 3 xifres menys). Degut a aixó ultim s'ha utilitzat la funció `millis()`, ja que al ser un aparell orientat a l'investigació i/o manipulació puntual d'algun element, es suposa que sempre s'acabarà reiniciant com a mínim 1 vegada per dia (rang de 1 a 16 hores de funcionament abans d'apagar el sistema durant la nit).

5.2.2. Programació d'altres funcions del Edge Computer.

Un cop explicat la part principal del programa (el calcul del controlador PID), cal mencionar que per algun moti o altre s'ha tingut que incorporar 3 funcionalitats addicionals que cal tenir en compte. Aquestes 3 funcionalitat són:

1) Comptador que rep les senyals del encoder: La majoria d'encoders moderns donen un valor relatiu. Basicament tenen un nombre determinat de marques al llarg de la circumferència (360 per exemple) i cada vegada que passen per un 2 pins de sortida generen 2 pulsos amb un petit delay (el delay entre la sortida A i B permet saber el sentit de gir). En cas de tenir aquesta classe de sensor fa falta un element que serveixi de comptador per tal mantenir el registre de la posició actual (sumi +1 o -1 cada vegada que li arribi un pols depenent del sentit de gir).

Ja que aquests pulsos són completament "aleatoris" (no tenen un període fix) i saltar-se uns quants pot tenir conseqüències nefastes (es va generant un error estàtic bastant difícil de detectar), s'ha utilitzat una interrupció de hardware, de manera que es sumi +1 o -1 (depen de l'estat de la sortida B en aquell moment) a una variable cada vegada que es detecti un flanc de pujada en la sortida A. Aquesta és l'única interrupció del programa (sempre està disponible).

2) Incorporar un "Protocol HOME": Mentre es mantingui activat un pulsador (una entrada digital) no li arribi consignes al manipulador i el PID canvi a una consigna de posició = 0° constant. Basicament si es manté pressionat aquell botó el braç robòtic del manipulador intentarà tornar a una "posició de repos" on l'angle és igual a 0° (angles que tenia al encendre el sistema).

Aquesta protocol de retornar a la posició de repos (posició HOME) és pensat per evitar que el sistema es desconnecti en una posició "incorrecta" que desincronitzi els encoders del controlador i manipulador (al ser encoders relatius els 0° es generen al reiniciar el sistema). Al tenir una forma fàcil de que torni a la posició inicial, és molt més fàcil desconnectar el sistema (simplement es pressiona, s'espera a que torni a la posició designada i finalment es desconnecta la font d'energia del sistema de control per tal d'apagar-lo).

3) Incorporar el "descodificador dels sensors HALL": Apart de calcular el duty cycle (% de potència que entrarà al motor), al tenir un motor BLDC fa falta decidir en tot moment quins 2 transistors cal tenir activats (quina branca serà l'entrada i quina serà la sortida) depenent de l'estat dels sensors HALL. Encara que es podria realitzar possiblement a nivell analògic, primer es vol intentar incorporar-ho al programa informàtic (minimitzar cost de components).

Primer de tot, cal recordar que l'angle del motor (sensors HALL) no te perquè coincidir amb l'angle de l'eix (encoder), ja que lo més normal és que hi hagi una caixa de canvis per minimitzar la velocitat i augmentar el parell motor (l'angle de l'encoder serà una fracció de la suma de revolucions realitzades), motiu per el qual cal mesurar-los per separat. Els sensors HALL únicament serviran per determinar en quin dels 6 sectors està el rotor del motor.

Independentment dels detalls de quin bit és quin HALL o quan cada bit està a ON/OFF, l'idea d'aquesta part del codi és que utilitzant els 3 bits dels sensors HALL i el bit “dir” (signe del resultat del PID) es generin 12 estats potencials (6 per gir horari i 6 per gir antihorari). Un cop es té una expressió if (o case) que separi aquests 12 casos simplement fa falta que en cada un d'ells únicament 2 sortides PWM estiguin actives, seguint el dutyCycle calculat en el PID.

A efectes practics això signifia que s'ha de posar totes les sortides a 0 i acte seguit reactivar les 2 sortides PWM de l'estat actual amb el duty cycle calculat durant l'últim cicle. Gran part d'aquest proces s'ha separat en forma de funció secundaria que la funció principal crida cada vegada que cal modificar les sortides.

Ja que el canvi de sector generalment serà bastant rapid (pot superar el periode de mostreig), s'ha col·locat una versió reduïda d'aquesta funcionalitat dins del while, per tal d'actualitzar les sortides mentre s'espera l'inici del proxim cicle si es detecta un canvi en els sensors HALL.

Cal recordar que la taula de la veritat utilitzada per traduir inputs (bits del sensor HALL) en outputs (quins transistors activar) és la següent:

Angle rotor (HALL[3])	HALL C	HALL B	HALL A	Sortides (dir = 0)	Sortides (dir = 1)
Qualsevol	0	0	0	Mantenir anterior	Mantenir anterior
330 – 30	0	0	1	Up 3 + Down 2	Up 2 + Down 3
210 – 270	0	1	0	Up 2 + Down 1	Up 1 + Down 2
270 – 330	0	1	1	Up 3 + Down 1	Up 1 + Down 3
90 – 150	1	0	0	Up 1+ Down 3	Up 3 + Down 1
30 – 90	1	0	1	Up 1+ Down 2	Up 2 + Down 1
150 – 210	1	1	0	Up 2 + Down 3	Up 3 + Down 2
Cap	1	1	1	ERROR! (tot OFF)	ERROR! (tot OFF)

Taula 5.1. Taula de la veritat utilitzat per en el programa de l'Arduino [1] [6]

Cal mencionar que al ser un control purament digital, en cas de tenir-se que modificar alguna cosa, simplement faria falta modificar quins pins s'està activant en cada cas. En canvi, en un circuit analogic podria fer falta dissenyar i comprar una nova placa PCB on s'arregli el problema (tardaria més temps i tindria un sobrecoast superior).

És important mencionar que el codi complet per implementar el Edge Computer en una Arduino UNO (o placa equivalent) ha sigut adjuntat en els annexos d'aquest TFG.

5.2.3. Progrmació de la Fog Computer (CPU).

Tal com s'ha comentat anteriorment, s'ha dividit els 2 controladors digitals en diversos microcontroladors de “edge computing” (una Arduino per cada articulació), que realitzen els calculs mencionats en els capítols 5.2.1 i 5.2.2, i una CPU de “fog computing” que s'encarrega de una serie de processos que cal tenir centralitzats. Una llista dels processos més importants que cal realitzar en aquesta CPU són:

1) Unir les connexions en sèrie: Cada CPU serveix de nexa entre els diferents microcontroladors (Arduinos) del seu aparell i la CPU de l'altre aparell. Cal utilitzar algun aparell amb bastants ports USB que pugui soportar el bit-rate de 6 o més connexions (les 5 Arduinos i enviar l'informació cap a l'altre CPU). Això és fa a nivell de Hardware (simplement s'ha de tenir suficients ports USB per connectar-se amb tot lo important).

Degut a aquesta necessitat farà falta instal·lar un element a la Raspberry per augmentar el nombre de ports USB fins a tenir-ne 6 o més. Pel disseny d'aquest TFG un HUB de 4 sortides (8 en total) és suficient: Els primer 5 ports serviran per connectar les 5 Arduinos, el 6e port per comunicar-se amb l'altre Raspberry, el 7e port per el Hub en si i el 8e per connectar un ratolí a la Raspberry amb el qual interactuar amb ella i les Arduinos que controla. El model seleccionat és el “Sabrent 4-Port USB 3.0 Data Hub” que és pot comprar via Amazon:



Fig. 5.2. Fotografia del Data Hub utilitzat per obtenir els ports USB 3.0 addicionals [8]

2) Un cop llegit els valors que li arriben de les Arduinos, la Raspberry els ha d'unir tots i potencialment modificar. Algunes modificacions potencialment útils són reescalar valors de les forces si no es vol tenir que aplicar tanta força al controlador (CPU del manipulador) i fer una operació matemàtica per traduir les consignes de l'espallta (ambes CPUs), les quals utilitzen 2 sistemes de coordenades lleugerament diferents.

3) Un cop totes les dades s'han unit i adaptat, cal enviar-les a l'altre CPU. Lo més fàcil és enviar-ho en un sol missatge com una llista de valors separats amb algun caràcter (un espai o un “ ; ” són mètodes bastant típics).

4) “Desxifrar” el missatge que l'altre CPU ha enviat (separar-lo en les 5 consignes) i enviar cada consigna de forma individual al microcontrolador (Arduino) que té assignat.

5) Addicionalment, un cop implementat la resta d'aparells del sistema de manipulació teleoperada que han quedat fora de l'abast d'aquest informe, aquestes CPUs s'encarregaran d'enviar la informació enregistrada cap un servidor (ja sigui un aparell físic o una màquina virtual de Cloud Computing).

Tots aquests processos han sigut integrats en un mateix script de Python que s'executarà de forma ininterrompuda (cal executar-lo mitjançant el ratolí + pantalla que es connectarà a cada Raspberry). Aquest script de Python conté els següents elements:

Setup (part del codi executada al reiniciar-se):

1) Importar llibreries utilitzades: Llibreria “datetime” per calcular el temps en cas de fer delays o temporitzadors (per enviar informació de forma periòdica) i llibreria “serial” per simplificar la lectura i escriptura d'informació mitjançant els ports USB (connexió en sèrie).

2) Iniciar la connexió sèrie amb els 6 ports USB (5 Arduinos + l'altre CPU): Aquest apartat és l'únic apartat susceptible de canvis ja que el nom que cada port USB rep depèn lleugerament del Hardware (especialment els 4 ports USB externs) i del sistema operatiu.

3) Inicialitzar totes les dades: Python únicament requereix inicialitzar llista de valors (grups de 5 variables a llegir i/o enviar), constants (temps de mostreig) o que es consulten immediatament (valor del temporitzador ha de tenir en el 1r cicle de mostreig).

3.5) Addicionalment en la CPU del manipulador s'ha d'introduir el valor del “Multiplicador” de força. Aquest valor s'introdueix amb el teclat al reiniciar la CPU del manipulador (ja que no hi ha USB extres s'utilitzarà una app de Linux que genera un teclat virtual). Aquest, juntament amb pulsar el botó de “RUN”, és l'únic pas on l'usuari ha de interaccionar si o si amb la Raspberry per tal de inicialitzar el programa.

En cas que no es desitges interaccionar amb la Raspberry, hi hauria un 2n script que directament col·loca un valor per defecte =1 on únicament s'ha de donar-lo al boto de RUN i ja està.

Loop infinit (part del codi que s'executa constantment):

4) Revisa si algun dels 5 ports connectats a l'Arduino té informació pendent d'analitzar.

4.1A) Si hi ha informació de tipoc "numeric" (es un int) simplement es reescriu un registre intern on es guarda les ultimes dades rebudes de cada port. En el cas de la CPU del manipulador addicionalment es multiplica aquest valor per el "Multiplicador" (és l'única lloc on s'utilitza).

4.1B) Si hi ha informació no numerica (principalment string) simplement fa un print() a la pantalla de la Raspberry, per llegir que ha enviat (s'ha programat per enviar un parell de strings a mode d'avis / error).

5) Revisa si el port de l'altre CPU té informació pendent d'analitzar.

5.1) Si hi ha informació se suposa que és numerica (no hi ha cap protocol que envies paraules), es divideix en 5 nombres (l'informació entre CPUs és un string amb 5 nombres separats amb un espai " ") i es sobrescriu la llista amb els 5 valors anteriors.

5.2) Addicionalment, és realitza un calcul per passar del eix de cordenades de l'altre l'aparell a l'eix de cordenades propi (la CPU del manipulador passa de les coordenades del controlador a les propies), per traduir l'informació que l'altre CPU ha enviat a la CPU que fa aquest calcul. Aixó unicament es fa amb els 2 valors de l'espantilla ja que l'espantilla del manipulador i del controlador tenen eixos lleugerament diferents degut a com s'ha fet el disseny mecanic.

5.3) Un cop llegits i modificats els 5 valors que l'altre CPU li ha enviat, s'envia cada un dels 5 valor a l'Arduino pertinent (Valor N → Arduino que controla el motor N).

6) Finalment, es comprova si la data actual ja ha passat el temps on començava el proxim periode de mostreig. Si tot funciona de forma correcta, aixó voldrà dir que ha pasat aproximadament un Tmostreig de l'ultima vegada que s'ha entrat a aquest IF.

6.1) En cas afirmatiu, es suposa que ja és hora d'enviar el proxim paquet de dades a l'altre CPU i es prepara un string que contingui els 5 valors que s'han anat actualitzant en el punt "4.1A)". Tal com s'ha comentat anteriorment, les dades enviades entre CPUs són strings (conjunt de characters) que nomes contenen els 5 nombres separats amb un espai entre si. Una combinació valida de characters podria ser "4 109 69 131 0".

Al igual que amb el microcontrolador (Arduino), s'adjuntara el codi complet en l'anex d'aquest treball (val la pena revisar-lo en cas de que no quedi clar com s'ha realitzat)

6. Impacte mediambiental.

Tal com es va comentar durant l'apartat d'objectius, el manipulador teleoperat dissenyat al llarg d'aquest informe teòricament s'utilitzaria exclusivament per treballs d'investigació. Dit d'una altre manera, no es pretén fabricar-ne gaires (molt possiblement es fabricaria 1 o 2 models funcionals) i encara menys vendre'ls al mercat. Degut a això, és fàcil d'imaginar que l'impacte mediambiental de tot aquest projecte serà pràcticament nul (al fabricar-se únicament 1 o 2 unitats, casi no es consumiran matèries primeres ni electricitat).

El punt anterior molt probablement ja seria suficient per obtenir qualsevol classe de permís mediambiental que fos necessari demanar a alguna entitat reguladora (al consumir tant pocs materials tindrem un impacte mínim), però a més a més es tindran les següents precaucions:

- Tota peça de plàstic obtinguda via impressió 3D es realitzaria amb materials coneguts per poder-se reciclar (transformar en fils per imprimir peces addicionals). D'aquesta manera qualsevol peça inservible, ja fos degut a un error d'impressió, del disseny (dimensions lleugerament errònies) o durant el muntatge (trencar-se alguna part fràgil), es podria reutilitzar a l'hora d'imprimir el seu substitut. Això també s'aplicaria a peces que es canviessin durant un període de manteniment, període on s'intentaria salvar totes les peces que caldria canviar per tal de ser reutilitzades.
- Es realitzaria un sistema electrònic que no requerís estar connectat innecessàriament les 24h del dia, per tal de que quan no fos necessari tota la maquinària es pogués desactivar i no gastes electricitat innecessària (LEDs, CPUs, fonts d'alimentació, etc.)
- Es vigilaria que els components electrònics segueixin les normatives vigents de consum òptim d'electricitat.
- En cas de que un cop construït es decidís comercialitzar el disseny, venent-lo per exemple a altres laboratoris, molt probablement es realitzaria un projecte per veure que millores addicionals es podrien aplicar al disseny per minimitzar encara més l'impacte mediambiental.

Degut a tots els punts anteriors, podem assegurar que el projecte tindrà un impacte mediambiental molt reduït, lo qual ens assegurarà una alta viabilitat mediambiental.

7. Propietat intel·lectual i copyright.

En general aquest informe s'ha realitzat tenint en ment 3 principis:

- Maximitzar el nombre d'imatges pròpies (dissenys i/o fotografies realitzades per el propi estudiant). Totes les imatges d'aquest estil estaran marcades amb un [0].
- S'ha intentat utilitzar, sempre que fos possible, Wikimedia Commons. Aquesta web és una base de dades especialitzada en emmagatzemar audiovisuals de Domini Públic i/o que estan sota la llicència Creative Commons (CC). S'ha decidit utilitzar aquesta pagina en concret degut a ser molt popular gracies a formar part de la mateixa fundació que Wikipedia.
- Tota imatge tindrà un link en l'apartat de Referències (en general a la seva pagina de Wikimedia Commons).

En quant a utilitzar arxius informàtics de tercers (tots ells publicats de forma gratuïta a alguna pagina pertinent), aquest TFG únicament ha utilitzat 2:

- Una plantilla d'Excel per generar els diagrames de Gantt. Cal comentar que la plantilla es pot descarregar des de un fòrum oficial per compartir plantilles d'Excel.
- Un model 3D d'un braç humà, per tal de poder comparar-lo amb els diferents dissenys mecànics realitzats.

Finalment les webs i/o eines utilitzades per generar les diferents imatges d'aquest TFG són:

- Eina de dibuix "Paint", per dibuixar esbossos relativament simples per il·lustrar conceptes que són més fàcils d'explicar de forma visual. Aquesta eina ve preinstal·lada en Windows 10.
- Excel per generar el diagrama de Gantt, utilitzant la plantilla ja mencionada. S'ha utilitzat la llicència d'Excel 365 entregada per el centre.
- SolidWorks per generar la majoria de models 3D observats al llarg de l'informe. S'ha utilitzat la clau entregada a alumnes durant el curs 2021-2022.
- Web "Visual Paradigm Online" que permet generar de forma fàcil e intuïtiva un gran nombre de gràfics, tals com el diagrama de flowchart de la fig. 5.1, de forma totalment gratuïta.
- MatLab (i Excel) per fer els calculs relacionats amb les matrius de rotació del apartat 4.6.
- Web "I Love PDF" una pagina web util per, entre altres coses, combinar i ordenar PDFs.

8. Perspectiva de gènere e igualtat.

En els últims anys la preocupació per tal de que informes i/o experiments científics tinguin en compte les diferències inherents entre homes i dones, ja sigui física o mentalment, ha augmentat bastant. En el cas d'aquest informe les diferències físiques entre homes i dones són especialment evidents en 2 apartats:

1) Especificacions tècniques ideals: És ben sabut que les dones solen tenir més flexibilitat mentre que els homes solen ser més forts. Ja que el manipulador intenta imitar un braç humà, lo ideal seria tenir en compte les limitacions que generalment sofriran els 2 sexes:

- Els homes tindran un rang de moviment inferior (angles màxims i mínims de cada articulació).
- Les dones tindran un parell màxim inferior (carrega màxima), lo qual a més a més pot afectar la facilitat amb la qual podran utilitzar el controlador.

2) Interacció amb el controlador: El controlador es basa en forçar una posició al braç de l'usuari, degut a això s'ha de tenir compte de no aplicar-li gaire força. En cas de permetre que s'apliqui massa força l'usuari certs usuaris podrien ser incapaços de moure el controlador (i per tant el manipulador), sent que en el pitjor cas el feedback de força podria acabar lesionant el braç de l'usuari (massa força, anar fora del rang

3) Dimensions del controlador: El punt més variable entre diferents usuaris serà el diàmetre i la longitud del braç. Cada persona tindrà un braç amb una mida lleugerament diferent, motiu per el qual és important tenir en compte això durant el disseny per tal de que en teoria qualsevol usuari pogués equipar-se el controlador independentment de la forma del seu braç.

De forma resumida, degut als fets ja mencionats, s'ha tingut en compte els següents punts per tal de garantir que aquest disseny sigui inclusiu independentment de la seva forma física, del seu sexe (diferències entre homes i dones), de petites discapacitats que alguns usuaris pugin arribar a tenir (lesions en l'espatlla i/o colze) i sobretot de la longitud del seu braç:

- Reduir la força màxima que el feedback de força, tenint especial compte les forces màximes que com a mínim un 99% de la població pot realitzar de forma fàcil, per facilitar l'ús del controlador a la vegada que reduïm enormement el risc de lesionar-se.
- Dividir el controlador en 3 segments (espatlla, colze i canell) per tal de poder-se adaptar a la longitud del braç de l'usuari. Fent que la unió entre el controlador i el braç de l'usuari sigui adaptable, per tal de no dependre del diàmetre del braç.

9. Conclusions.

Com a conclusió d'aquest informe es pot comentar que al llarg de l'elaboració d'aquest TFG s'ha trobat un gran nombre d'obstacles que ha fet falta solucionar d'alguna forma. Mentre que gran part d'aquests obstacles es consideren completament solucionats, es interessant comentar que alguns elements del disseny (especialment relacionat amb l'espalla i/o el controlador) caldria acabar de polir en un projecte posterior, un cop s'hagi pogut comprovar que elements funcionen de forma correcte i quins no (mai se pot estar 100% segur que res fallarà).

Encara que el concepte de que "el informe no es el final del disseny" pot arribar a ser un procediment discutible, es considera que en robòtica tenir macro projectes iteratius (dissenyar, simular, construir, revisar i tornar a dissenyar amb els nous coneixements) és el millor protocol a seguir per obtenir un resultat òptim. Sent que al ser una única persona i no poder construir cap prototip físic (està fora de l'abast degut al cost de tots els materials) aquest procés iteratiu s'hauria de continuar (i eventualment completar) en projectes posteriors d'optimització.

En conclusió, aquesta memòria de com s'ha dissenyat un braç robòtic (manipulador) i un controlador ha tingut uns resultats bastant òptims, sent que l'únic inconvenient inesperat que si o si s'hauria de solucionar en un projecte posterior és acabar el càlcul de l'apartat 4.2.

10. Referències.

[1] Elaboració pròpia de l'escriptor d'aquest informe (Marc Coll Coll). Claudàtors addicionals indiquen que la figura utilitza recursos addicionals (imatges o programes de tercers).

Les imatges extretes d'internet per l'elaboració d'alguna taula o figura (hagin sigut modificades o no) han sigut obtingudes en les següents pàgines web:

[2] <https://commons.wikimedia.org>, unió de diverses imatges d'aquesta web. Links individuals no disponibles.

[3] https://commons.wikimedia.org/wiki/File:Human_arm_bones_diagram.svg, Mariana Ruiz Villarreal. "File:Human arm bones diagram.svg", 3 de gener, 2007.

[4] <https://www.researchgate.net/publication/311099028>, Leal-Naranjo, José & Ceccarelli, Marco & Torres San Miguel, Christopher. "Mechanical Design of a Prosthetic Human Arm and its Dynamic Simulation", Novembre de 2017. *La taula 2.1 s'ha extret de la pàgina 484.*

[5] <https://docs.arduino.cc/hardware/duemilanove>, Especificacions tècniques de l'ARDUINO DUE (pàgina web oficial), consultat per últim cop el 20/4/2022. *Fig. 3.1 extreta del botó "PINOUT".*

[6] https://www.researchgate.net/figure/Schematic-diagram-of-a-three-phase-BLDC-motor-drive_fig1_289559604, Tashakori, A. & Ektesabi, Mehran. "Position Sensors Fault Tolerant Control System in BLDC Motors", Febrer de 2014.

[7] <https://www.posterus.sk/?p=2589>, Posterus. Speed control of BLDC motor based on hall-effect sensors, 4 de Novembre de 2009. *La figura 3.6 es basa en una de les seves imatges.*

[8] <https://www.amazon.com/dp/B00JX1ZS5O?tag=aboutcom02lifewire-20&linkCode=ogi&th=1&psc=1&ascsubtag=4142295%7Cn87822dc4db034d14a7f633bb04e0f011%7CB00JX1ZS5O>, Amazon. *Punt de venda del USB Hub seleccionat per el disseny electronic. Una de les seves imatges s'ha utilitzat com a figura 5.2.*

Recursos digitals (models 3D, programes de dibuix, etc.) utilitzats per elaborar diverses figures i/o plànols es poden observar a continuació:

- [9] <https://online.visual-paradigm.com/es/>, Visual Paradigm Online. *Servei web utilitzat per facilitar la creació de diagrames 2D de flux d'informació i/o de circuits elèctrics.*
- [10] <https://powersimtech.com/products/psim/capabilities-applications/>, PowerSIM. *Eina informàtica utilitzada per crear la majoria de circuits electrònics i simular-los.*
- [11] <https://www.solidworks.com/sw/support/downloads.htm>, Solidwors. *Eina informàtica per crear models 3D professionals, transformar-los en plànols i executar simulacions estàtiques per determinar un factor de seguretat aproximat*
- [12] <https://www.microsoft.com/es-es/microsoft-365/excel?rtc=1>, Microsoft Excel. *Eina informàtica per realitzar i visualitzar càlculs semiautomàtics o crear petites taules.*
- [13] <https://es.mathworks.com/products/matlab.html>, MatLab. *Eina matemàtica per realitzar càlculs complexos de forma semiautomatitzada.*
- [14] <https://templates.office.com/es-es/Diagrama-de-Gantt-simple-TM16400962>, Vertex42. *Plantilla d'Excel per facilitar la creació dels diagrames de Gantt.*
- [15] <https://www.adobe.com/la/products/photoshop.html> Adobe Photoshop. *Eina de dibuix per unir diferents assets en una mateixa imatge i/o eliminar fons de colors.*
- [16] <https://www.ilovepdf.com/es>, I Love PDF. *Servei web utilitzat per unir PDFs.*

Pàgines web consultades, únicament per tal d'obtenir informació durant el procés d'investigació, es poden trobar a continuació. Cal comentar que moltes fons són redundants per evitar biaix (utilitzen dades i/o arguments bastant similars):

- [17] <https://garage.hp.com/us/en/generation-impact.html>, HP. “Generation Impact”, consultat per últim cop el 20/4/2022.
- [18] <https://www.youtube.com/c/IanDavisFpd/playlists>, Ian Davis. Consultat per últim cop el 20/4/2022. *Vídeos fabricant i/o demostrant el funcionament de mans prostètiques*
- [19] <http://virtualrealitytimes.com/2017/02/28/list-of-full-body-virtual-reality-haptic-suits/>, VirtualReality Times. “List of Full Body Virtual Reality Haptic Suits”, 28 de febrer de 2017.

- [20] <http://www.cyberglovesystems.com/>, CyberGlove Systems. Catàleg productes relacionats amb datagloves i/o feedbacks de control, consultat per últim cop el 20/4/2022.
- [21] <http://www.oepm.es/es/index.html>, Oficina Española de Patentes y Marcas. Consultat per últim cop el 20/4/2022.
- [22] <https://www.itson.mx/publicaciones/rieeyc/Documents/v4/art2junio08.pdf>, RIEE&C (Revista de Ingeniería Eléctrica, Electrónica y Computación), Vol. 4 No. 1, Juliol de 2008.
- [23] <https://forum.arduino.cc/t/example-of-driving-adc-and-dac-from-timer-for-regular-sampling/199849>, MarkT. “Example of driving ADC and DAC from timer for regular sampling”, consultat per últim cop el 20/4/2022. *Codi per ARDUINO DUE*.
- [24] <https://www.youtube.com/watch?v=uKhxcOLQhKk>, Adam Savage’s Tested. “Realistic Robot Arm: Meet the Modular Prosthetic Limb!”, 28 de Juny de 2017. *Exposició d’un braç robòtic durant una convenció tecnològica*.
- [25] https://www.mutuauniversal.net/flippingbooks/04/data/downloads/04_ergonomia.pdf, Mutua Universal. “Prevención de riesgos laborales para PYME | Ergonomía”, Octubre 2017.
- [26] https://istas.net/sites/default/files/2019-12/M4_MetodosEvaluaci%C3%B3nErgo.pdf, ISTAS (Instituto Sindical de Trabajo, Ambiente y Salud). “Métodos de Evaluación Ergonómica | Módulo 4”, consultat per últim cop el 20/4/2022.
- [27] <https://www.healthline.com/human-body-maps/arm>, Jill Seladi Schulman. “Arm”, actualitzat el 27 d’Agost de 2018.
- [28] <https://doi.org/10.1007/s00221-012-3235-0>, Xu, Y., Terekhov, A.V., Latash, M.L. et al. “Forces and moments generated by the human arm: variability and control”. *Exp Brain Res* 223, 159–175 (28 de Setembre de 2012).
- [29] <https://www.carotron.com/applications/csag-d/>, Carotron. “Motor Torque & Speed Control | Motor Control Basics: Drive Operating Modes”, consultat per últim cop el 20/4/2022.
- [30] <https://www.researchgate.net/publication/326681369>, Leal-Naranjo, José & Torres San miguel, Christopher & Ceccarelli, Marco & Rostro-Gonzalez, Horacio. “Mechanical Design and Assessment of a Low-Cost 7-DOF Prosthetic Arm for Shoulder Disarticulation”, Setembre de 2018.
- [31] <https://www.solomotorcontrollers.com/blog/how-to-control-speed-or-torque-of-your-brushless-motor-with-arduino-using-solo-closed-loop/>, SOLO Team (SoloMotorControllers).

“How to controll speed or torque of your brushless motor with arduino using SOLO (Closed-loop)”, consultat per últim cop el 20/4/2022.

[32] <https://www.iiste.org/Journals/index.php/JETP/article/viewFile/1192/1113>, Journal of Energy Technologies and Policy. ISSN 2225-0573 (Online) Vol.2, No.1, 2011.

[33] <https://www.roboticstomorrow.com/article/2021/10/how-to-select-the-best-motor-for-a-jointed-arm-robot/17618>, Rozum Robotics. “How to select the best motor for a jointed arm robot”, 21 d'Octubre de 2021.

[34] <https://www.thomasnet.com/articles/machinery-tools-supplies/dc-motors-vs-servo-motors/>, Christian Cavello. “DC motors vs. Servo motors – What's the difference?”, consultat per últim cop el 20/4/2022.

[35] <https://www.elprocus.com/motors-used-in-industrial-robotics/>, diversos experts. “Experts Outreach for motors used in industrial robotics | Expert Out reach”, consultat per últim cop el 20/4/2022.

[36] <https://www.evsint.com/types-of-robot-joints/>, EVS TECH CO. “Types of robot joints: A detailed guide”, consultat per últim cop el 20/4/2022.

[37] <https://www.youtube.com/c/Skyentific/videos>, Skyentific. Consultat per últim cop el 20/4/2022. *Canal d'enginyeria especialitzat en desenvolupament de projectes de robòtica.*

[38] <https://patents.google.com/patent/US3945053>, Benny M. Hillberry & Allen S. Hall Jr. “US3945053A - Rolling contact prosthetic knee joint”, va expirar el 23 de març de 1993.

[39] <https://www.omch.co/absolute-rotary-encoder/>, OMCH. Absolute Rotary Encoder, consultat per últim cop el 15/6/2022.

[40] <https://www.youtube.com/watch?v=szgVUfyX8JM>, Texas Instruments. Introduction to InstaSPIN™ – BLDC Motor Control Solution, consultat per últim cop el 15/6/2022.

[41] <https://www.youtube.com/watch?v=teeMdFaykPE>, Texas Instruments. Sensored vs. sensorless control, consultat per últim cop el 15/6/2022.

[42] <https://www.youtube.com/watch?v=Sww1mek5rHU>, DroneBot Workshop. Arduino Uno to ATmega328 - Shrinking your Arduino Projects, consultat per últim cop el 15/6/2022.

- [43] <https://www.directindustry.com/prod/renishaw/product-5200-1336851.html>, Direct Industry. Encoder absolut rotatiu que es pot comprar si es vol canviar el tipu d'entrada, consultat per últim cop el 15/6/2022.
- [44] <https://github.com/vedderb/bldc-hardware>, Vedderb GitHub. VESC-Open Source ESC, consultat per últim cop el 15/6/2022.
- [45] https://www.youtube.com/watch?v=8swJ_Bnsgl4, GreatScott. Here is why MOSFET drivers are sometimes essential! || MOSFET Driver Part 1 (Driver, Bootstrapping), consultat per últim cop el 15/6/2022.
- [46] <https://www.youtube.com/watch?v=35YuILUlfGs>, GreatScott. From Idea to Schematic to PCB - How to do it easily!. Consultat per últim cop el 15/6/2022.
- [47] <https://jlcpcb.com/>, JLCPCB. Pagina web per comprar plaques PCBs de disseny propi, consultat per últim cop el 15/6/2022.
- [48] <https://dronebotworkshop.com/rotary-encoders-arduino/>, DroneBot Workshop. Using Rotary Encoders with Arduino, consultat per últim cop el 15/6/2022.
- [49] <https://www.youtube.com/watch?v=cdiZUszYLiA>, Texas Instruments. Field Oriented Control of Permanent Magnet Motors, consultat per últim cop el 15/6/2022.
- [50] <https://www.ti.com/lit/ug/tidu248/tidu248.pdf>, Texas Instruments. Phase-Shifted Full Bridge DC/DC Power Converter Design, consultat per últim cop el 15/6/2022.
- [51] <https://www.youtube.com/watch?v=InzXA7mWBWE>, Texas Instruments. Comparison of commutation methods, consultat per últim cop el 15/6/2022.
- [52] <https://www.wellpcb.com/types-of-arduino.html>, WellPCB. Types of Arduino Boards: A Comprehensive Guide, consultat per últim cop el 15/6/2022.
- [53] https://create.arduino.cc/projecthub/eliott3005/pwm-and-direction-control-of-a-dc-motor-via-bluetooth-ced466?ref=search&ref_id=pwm&offset=1 eliott3005 (Arduino Project Hub). PWM and direction control of a DC motor via Bluetooth, consultat el 15/6/2022.
- [54] <https://microcontrollerslab.com/arduino-pwm-tutorial-generate-fix-and-variable-frequency-signal/>. Arduino PWM Tutorial: Generate Fix and Variable Frequency and Duty Cycle Signal, consultat per últim cop el 15/6/2022.

- [55] <https://arduinogetstarted.com/tutorials/arduino-serial-monitor>, Arduino Get Started. Arduino – Serial Monitor, consultat per últim cop el 15/6/2022.
- [56] <https://www.thomasnet.com/articles/instruments-controls/all-about-electrical-sensors/>, Thomas (a xometry company). All About Electrical Sensors, consultat el 15/6/2022.
- [57] <https://www.lifewire.com/best-usb-hubs-4142295>, Lifewire. Here Are the Best USB Hubs, Tested by Our Experts, consultat per últim cop el 15/6/2022.
- [58] <https://www.youtube.com/watch?v=P9XsGbLoRog&t=502s>, Creative Think. Making High Torque Big Brushless Dc Motor, consultat per últim cop el 15/6/2022.
- [59] <https://alienpowersystem.com/shop/brushless-motors/80mm/c80100-sensored-outrunner-brushless-motor-50kv-7000w/>, Alien Power System. Motor seleccionat com a actuator (amb paràmetres electrònics), consultat per últim cop el 15/6/2022.
- [60] <https://www.rls.si/eng/lm10-linear-and-rotary-magnetic-encoder-system?partNumbers=MR045C028B064A00&qty=1>, RLS. *Catalag i punt de compra d'encoders.*
- [61] https://www.amazon.com/-/es/Interruptor-corriente-miniatura-normalmente-detectable/dp/B086YT49H7/ref=sr_1_7?_mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&&crid=FX5255H4VOC&&keywords=Arduino+high+power+current+sensor&&qid=1655294041&&rnid=2941120011&&s=industrial&&sprefix=arduino+high+power+current+senso,aps,151&&sr=1-7&&th=1&language=es_US¤cy=EUR, Amazon. *Sensor de corrent seleccionat.*