

Títol: Caminador intel·ligent per a persones amb discapacitat

Volum: 1

Alumne: Francisco Bonnín Pascual

Director/Ponent: Antonio B. Martinez Velasco

Departament: ESAII

Data: 1 de febrer de 2008

DADES DEL PROJECTE

Títol del Projecte: Caminador intel·ligent per a persones amb discapacitat

Nom de l'estudiant: Francisco Bonnín Pascual

Titulació: Enginyeria Informàtica

Crèdits: 37,5

Director/Ponent: Antonio B. Martinez Velasco

Departament: ESAIL

MEMBRES DEL TRIBUNAL (*nom i signatura*)

President: Joan Climent Vilaró

Vocal: Carme Martin Escofet

Secretari: Antonio B. Martinez Velasco

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

Índex:

1	Introducció	9
2	Estat de l'Art	11
2.1	Caminadors Intel·ligents	11
2.2	Sensors de Força	18
2.2.1	Principis de funcionament.....	18
2.2.1.1	Afectació de la temperatura	19
2.2.2	Transductors comercials	20
2.2.3	Altres tipus de sensors de força	21
2.3	Amplificadors de Senyal.....	23
2.3.1	Amplificadors d'instrumentació	23
2.3.2	Condicionadors de Senyal Programables	25
3	Objectius del projecte i treball a realitzar.....	27
4	Mesura de Forces	29
4.1	Principis de l'extensometria.....	29
4.2	Usant el mètode d'elements finits	29
4.3	Disseny dels transductors pel i-Walker	30
4.3.1	Transductor per a la component X (manetes).....	32
4.3.2	Transductor per a les components Y i Z (manetes).....	34
4.3.3	Transductor per a la component Z (rodes posteriors)	35
5	Calibratge dels Sistemes de Mesura de Força.....	37
5.1	Mètode de calibratge	37
5.2	Calibratge dels transductors de les manetes	38
5.3	Calibratge dels transductors de les rodes.....	39
5.4	Dades dels transductors	40
5.4.1	Manetes	40
5.4.2	Rodes posteriors.....	41
6	Amplificació de la mesura	43
6.1	Escalat del guany	43
6.2	Ajust de l'"offset".....	45
6.3	Voltatge de referència	45
6.4	Excitació del sensor i linearització.....	45
6.5	ADC per al sensor de temperatura	46
6.6	EEPROM externa i coeficients de temperatura	46
6.7	Monitorització d'errors	47

6.8	Limitació sobre-escala i sota-escala	48
6.9	Interfície digital	48
6.10	Mode normal de funcionament	49
6.11	Pin de test	51
7	Disseny i implementació del sistema Microcontrolador	53
7.1	Interfície PGA – Microcontrolador (I ² C)	54
7.2	Microcontrolador	56
7.3	Disseny Funcional	58
7.4	Disseny de la PCB	60
8	Firmware del Microcontrolador	69
8.1	Codi comentat	71
9	Comunicacions (CAN)	81
9.1	Controller Area Network	81
9.2	Principals característiques de CAN	81
9.3	Protocol de comunicacions CAN	82
10	Interfície gràfica	87
11	Avaluació de Resultats	101
11.1	Metes assolides	101
11.2	Millores/Ampliacions	101
12	Valoració econòmica	103
13	Planificació seguida	105
14	Bibliografia	109
15	Agraïments	109
	Annex 1: Pont de Wheatstone	113
	Annex 2: Amplificadors Operacionals	115
	Comportament en corrent continua	116
	Llaç obert	116
	Llaç tancat	116
	Configuracions	117
	Comparador	117
	Seguidor	117
	Inversor	117
	No inversor	118
	Sumador inversor	118
	Restador	119

Integrador ideal.....	119
Derivador ideal.....	119



1 Introducció

El meu Projecte Final de Carrera forma part d'un projecte més gran en el que hi col·laboren diferents centres europeus mèdics i d'investigació. Basat en tecnologies assistencials, el projecte pretén millorar la vida diària de les persones amb discapacitats motrius i/o cognitives.

Aquest projecte europeu, de nom SHARE-it, té l'objectiu de desenvolupar un sistema escalable de sensors i tecnologia per a l'assistència de manera que pugui ser integrat modularment dins d'un entorn domèstic intel·ligent per a millorar l'autonomia individual. El sistema estarà dissenyat per a informar i assistir a l'usuari i als seus cuidadors mitjançant la monitorització i l'ajut a la mobilitat. Es pretén, aleshores, contribuir en el desenvolupament de la propera generació de dispositius per a l'assistència a la gent gran o a la gent amb discapacitats i augmentar en la mesura que sigui possible la seva autonomia.

Dins d'aquest marc, el meu Projecte Final de Carrera es basa en la col·laboració per al desenvolupament d'un caminador intel·ligent (i-Walker) que permeti millorar la mobilitat de les persones amb discapacitats motrius. Concretament es centra en l'estudi i incorporació de sensors de força a les manetes i al suport de les rodes posteriors d'un caminador de quatre rodes per tal de poder obtenir la magnitud i direcció de les forces que exerceix l'usuari a sobre de l'aparell.



Figura 1: caminador de sèrie



Al meu PFC també s'inclou el tractament de les dades obtingudes dels sensors de força així com la visualització d'aquestes en un entorn basat en realitat virtual a temps real. El marc interdisciplinari en el que es troba el meu projecte fa que sigui molt important ensenyar de manera gràfica i intuïtiva les dades que obtenim per a que puguin ser enteses per persones de diferents àmbits professionals, com poden ser els enginyers i els metges.

En el desenvolupament d'aquest projecte treballem conjuntament amb els departaments de Resistència de Materials i Estructures en l'Enginyeria (RMEE) i d'Enginyeria Mecànica (EM) liderats per Francesc Roure i Joaquim Agulló respectivament.

El RMEE s'encarrega del disseny i implementació dels sensors de força. La versió del caminador usat en aquest PFC consta de 8 sensors de força, tres en cada una de les manetes (eixos X, Y i Z) i un en cada una de les rodes posteriors per a captar la força normal al terra.



Figura 2: caminador amb els sensors de força dissenyats per Francesc Roure

Per altra banda, els professors Sergi de Haro i Joaquim Agulló, del departament d'Enginyeria Mecànica, s'han encarregat de l'estudi de la cinemàtica inversa i directa del caminador. D'aquesta manera podem saber com afecten les forces aplicades al caminador i quines forces haurien de ser aplicades per a obtenir la resposta desitjada.



2 Estat de l'Art

Aquí es presenta informació d'avenços fets fins al moment en caminadors intel·ligents, sensors de detecció i mesura de forces i amplificadors de senyal.

2.1 Caminadors Intel·ligents

En aquest primer apartat es parlen de diferents estudis realitzats que guarden algun tipus de relació amb els anomenats “Intelligent Walkers”; però en primer lloc farem una breu descripció del que és un caminador.

Un caminador o Rollator és un dispositiu destinat a la gent discapacitada que necessita ajuda per a mantenir el balanceig o l'estabilitat quan camina. Consisteix en una estructura dotada de dues manetes, on l'usuari es recolza, i rodes. El nombre de rodes depèn del model. En el mercat podem trobar:

- Caminadors amb dues rodes al davant i dues potes al darrera. Aquestes s'han d'aixecar o arrossegar per a avançar.
- Caminadors amb tres rodes, dues posteriors d'orientació fixa i una al davant en configuració “caster”.
- Caminadors amb quatre rodes, dues posteriors d'orientació fixa i dues al davant en configuració “caster”.



- 1.- Manetes
- 2.- Tub ajustable de les manetes
- 3.- Fre
- 4.- Maneta per ajustar l'altura
- 5.- Roda posterior
- 6.- Estructura
- 7.- Roda anterior
- 8.- Cistell
- 9.- Seient
- 10.- Safata
- 11.- Cable del fre

Figura 3: caminador de quatre rodes



Són bastants els projectes que s'han portat a terme amb la intenció de dotar a caminadors comercials, com el de la il·lustració anterior, de sensors i actuadors amb la intenció de fer que aquest dispositiu adquireixi certa “intel·ligència”. Tots aquests projectes han estat portats a terme abans del inici del nostre projecte i ens serviran per a veure fins on s’ha aconseguit arribar avui dia així com amb quins problemes es varen torbar els seu desenvolupadors i quines solucions varen proposar.

Comencem fent una classificació dels projectes que hem trobat en dos grans grups. Uns pretenen dotar al caminador d’habilitats per tal de realitzar tasques que puguin ajudar de manera activa a fomentar l’ús d’aquest dispositiu; i altres pretenen que el caminador sigui capaç de preveure la intenció i/o estat de l’usuari i ajudar de manera quasi imperceptible per a aquest quan es cregui convenient.

Dins del primer grup trobem projectes com el caminador augmentat robòticament de Jared Glover, David Holstius et al [2] desenvolupat a la Carnegie Mellon University, Pittsburgh. Aquest caminador disposa del “software” de navegació CARMEN, un odòmetre i un telèmetre làser; que fa servir per a situar-se dins de l’entorn pre-après i esquivar obstacles.



Figura 4: caminador augmentat de Carnegie Mellon



Fent servir aquestes habilitats el caminador és capaç d'aparcar-se sol quan no se'l necessita i tornar cap a l'usuari quan aquest el crida amb un comandament a distància per infraroigs. D'aquesta manera el caminador no molesta al pas de la gent quan no està essent usat.

Per altra banda, aquest caminador és capaç d'ajudar a l'usuari a orientar-se i a assolir el seu destí prèviament indicat mitjançant una interfície gràfica, en forma de senzill mapa, i uns botons.

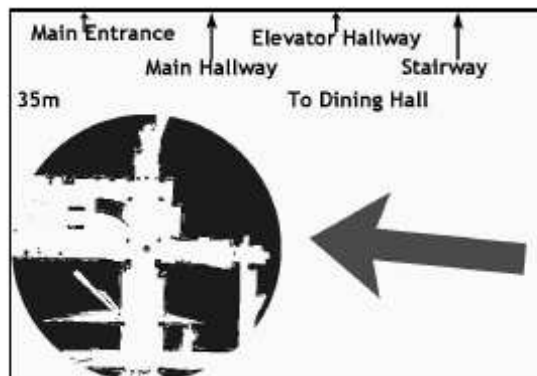


Figura 5: interfície gràfica desenvolupada

L'objectiu dels seus creadors és el de fer més còmode l'ús del caminador i d'aquesta manera fomentar-lo.

Dins del mateix grup trobem altres projectes com per exemple PEARL [3], un robot dissenyat per la mateixa universitat de Carnegie Mellon amb dos grans objectius: ajudar a les persones majors a recordar tasques diàries (com per exemple prendre una pastilla després de dinar) i guiar a través de l'entorn prèviament après pel dispositiu. Malgrat l'èxit d'aquest projecte, nosaltres no hi aprofundirem més ja que l'estructura física del robot difereix massa de la forma que té un caminador convencional.

L'altre grup definit engloba tots aquells projectes basats en caminadors capaços de romandre passius, com si d'un caminador normal es tractés, fins que detecten alguna situació definida i aleshores actuar de manera que l'usuari, a ser possible, no se n'adoni. Dins d'aquest grup trobem els projectes que més s'assemblen al que nosaltres volem acabar obtenint [1] [4] [5] [6].

La majoria d'aquest projectes es basen en la inferència de la intenció de l'usuari. En el disseny de Yasuhisa Hirata et al, d'una universitat japonesa [1], es proposa un sistema per a actualitzar els paràmetres usats en aquesta inferència, de manera que el caminador s'adapti a l'usuari que el fa servir,

les seves condicions diàries, l'entorn, etc. El principal paràmetre que es fa servir en aquesta proposta és el de la posició de l'usuari respecte al caminador. Per a captar-la s'usa un telèmetre làser que detecta la distància dels genolls al caminador. Suposant que el peu es troba a la mateixa distància, situen a l'usuari al punt mig dels dos peus.

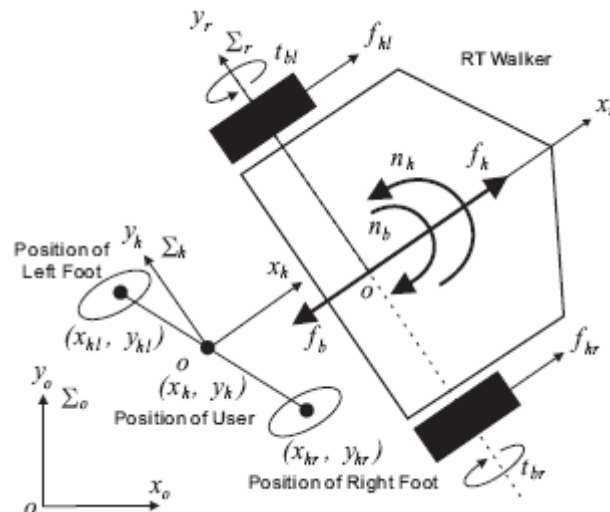


Figura 6: càlcul de la posició de l'usuari

Usant aquesta tecnologia els creadors d'aquest caminador són capaços de determinar l'estat de l'usuari i, en funció d'aquest, es tria una acció a realitzar.

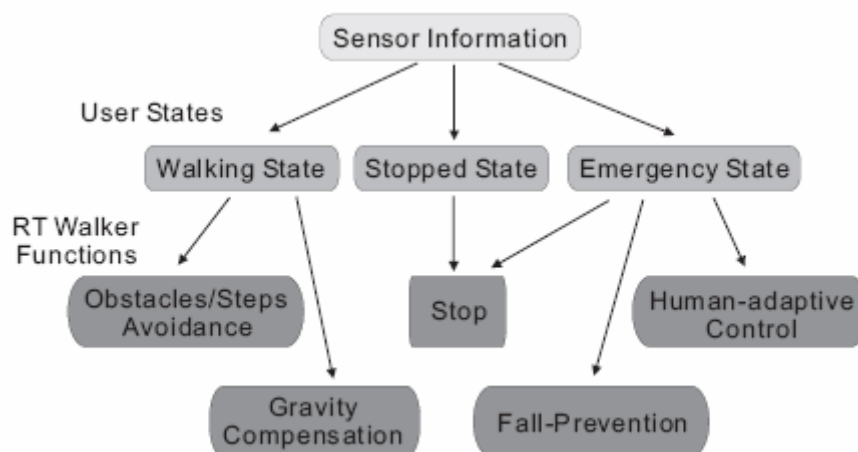


Figura 7: possibles accions en funció de l'estat de l'usuari

Per saber si l'usuari es troba a l'estat de parat només cal veure si la velocitat d'aquest i la del caminador no superen un cert llindar. Per altre banda, per saber si l'usuari es troba caminant o en estat d'emergència, es fa servir la distància entre aquest i el caminador.



Observant l'evolució de la posició de l'usuari respecte als eixos x i y del caminador quan es troba caminant s'obtenen unes distribucions normals com les que es mostren a la figura següent:

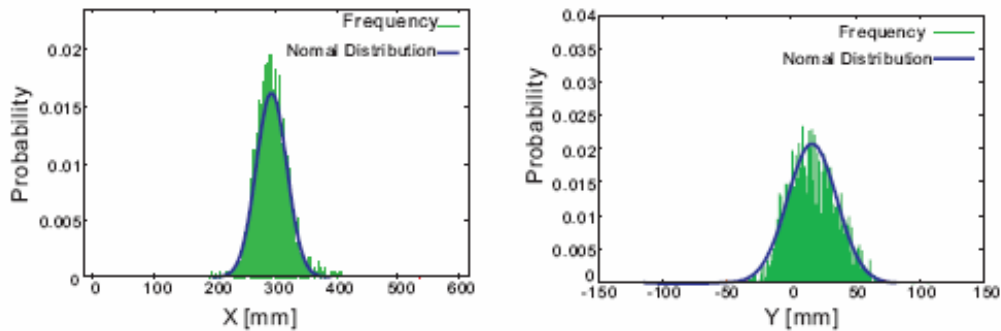


Figura 8: posició de l'usuari respecte els eixos X i Y

Integrant aquestes dues distribucions s'obté una distribució normal en 2 dimensions com la següent:

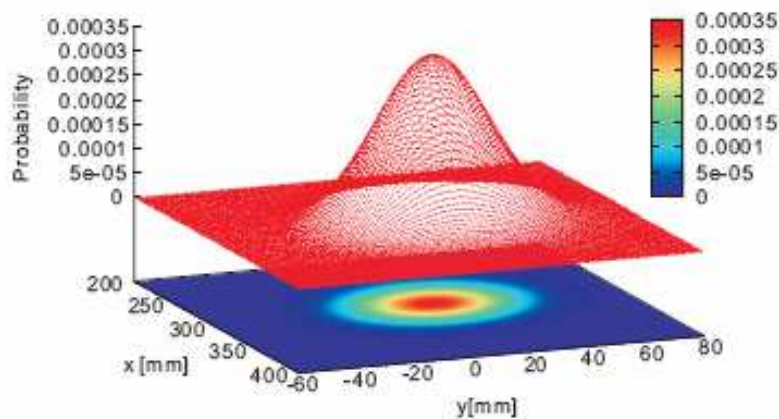


Figura 9: distribució normal de 2D de la posició de l'usuari

Aleshores, si la posició de l'usuari respecte al caminador es troba a dins de l'el·lipsoide l'usuari es troba en estat caminant, en cas contrari es troba en estat d'emergència.

Altres projectes existents fan servir un major nombre de sensors per tal d'inferir la voluntat i/o estat de l'usuari, al mateix temps que es centren més en com actuar quan es detecta una situació anòma/peril·losa [4] [5] [6]. Aquests projectes es basen en modificar la orientació de la roda davantera ("caster") d'un caminador de tres rodes per tal de modificar la direcció d'avanç de l'usuari. L'objectiu pot ser evitar una pèrdua d'estabilitat o un col·lisió.



Un sensor molt utilitzat en aquests projectes és el de força. Fent servir aquest dispositius es capten les forces que exerceix l'usuari a les manetes del caminador. Els resultats dels experiments realitzats revelen que l'intent de girar, està fortament relacionat amb el moment de gir total al voltant de l'eix vertical així com les forces laterals aplicades per l'usuari. D'aquesta manera es pot predir si l'usuari està intentant girar amb el caminador i canviar la orientació de la roda del davant (usant un motor incorporat ad hoc) per a tal de facilitar-ho.



Figura 10: caminador amb sensors 6-DoF

Usant la informació addicional d'altres tipus de sensors (senyors IR, sonar, etc.) es generen sistemes més o menys complexes per a controlar els actuadors integrats en el caminador. Per exemple es podrien bloquejar les rodes fent servir un fre mecànic accionat elèctricament quan es detectés una col·lisió imminent, o produir lleus correccions en la trajectòria per tal d'ajudar a desplaçar-se en un entorn més o menys dens.



És comú en alguns d'aquests projectes definir un control que té en compte la intenció inferida de l'usuari (per exemple girar a la dreta), determinada pels sensor de força de les manetes i/o per un històric del comportament de l'usuari que s'ha anat emmagatzemant; i la intenció de caminador (per exemple esquivar una columna), determinada a partir de la senyal de sensors que capten l'entorn.

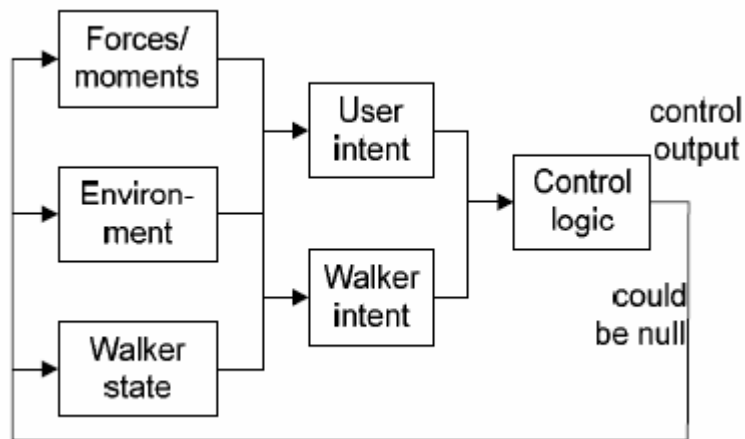


Figura 11: diagrama de blocs del control d'un caminador intel·ligent



2.2 Sensors de Força

A continuació s'expliquen que són i per a què serveixen els sensors de força o esforç.

2.2.1 Principis de funcionament

Un sensor de força és un dispositiu que permet mesurar les deformacions o esforços que es produeixen en un objecte quan se li aplica una força. Inventat per Edward E. Simmons i Arthur C. Ruge, el tipus més comú de sensor d'esforç està compost per una bobina plana de cable conductor introduïda a dins d'un aïllant elèctric.

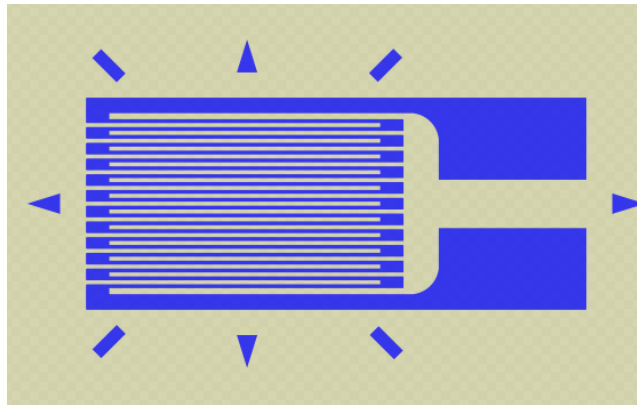


Figura 12: sensor d'esforç de Simmons i Ruge

Aquest dispositiu es fixat a l'objecte amb un adhesiu com per exemple el cyanocrilat. Quan l'objecte es deformat la bobina plana (galga) també es deforma, provocant un canvi en la seva resistència elèctrica. Aquest canvi en la resistència, normalment mesurat per un pont de Wheatstone (mirar annex 1), està relacionat amb l'esforç per una quantitat anomenada "Factor de la galga" (GF) de la forma següent:

$$GF = \frac{\Delta R / R_G}{\epsilon}$$

on R_G és la resistència de la galga sense deformat, ΔR és el canvi en la resistència provocat per l'esforç i ϵ és l'esforç.

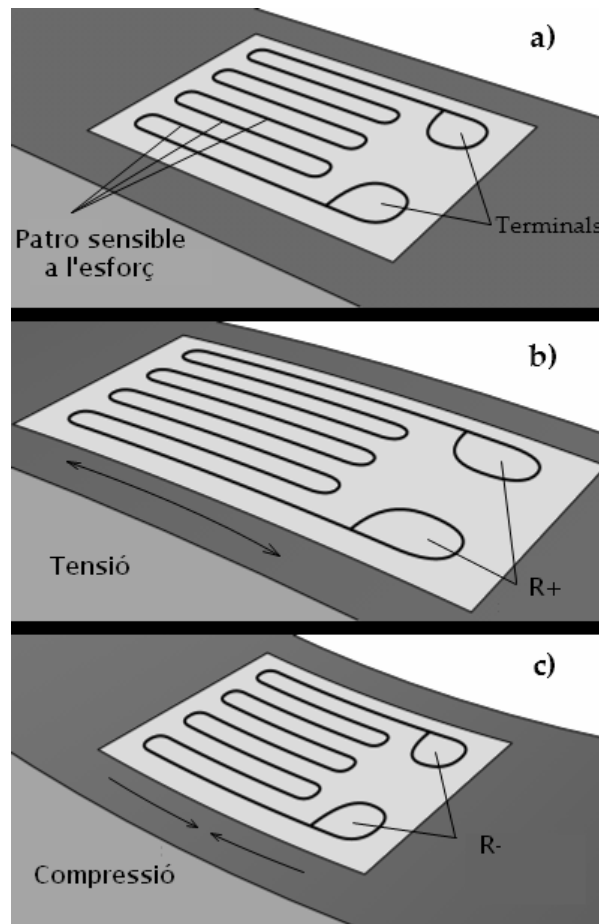


Figura 13: Variació de la resistència degut a la deformació

En funció de l'ús i de les condicions de funcionament del sensor de força es requereixen diferents configuracions de galgues. Trobem per exemple galgues que formen part de cèl·lules de càrrega que han de romandre estables per un període considerable d'anys, sinó dècades; i galgues usades per a mesurar la resposta d'un experiment dinàmic que només han d'estar adherides a l'objecte uns dies, ser alimentades per unes hores i operar menys d'un segon.

2.2.1.1 Afectació de la temperatura

Les variacions tèrmiques poden causar multitud d'efectes. L'objecte pot canviar de dimensions degut a la dilatació tèrmica, fet que pot fer variar la resistència de la galga i ser interpretat com un esforç.

Per evitar aquest fenomen es fabriquen galgues de materials que presenten la mateixa resistència elèctrica en un ampli rang de temperatures (per exemple aliatges de constantà). Per altra banda, algunes galgues



d'aliatges de Karma o constantà han estat dissenyades per a que la variació de la resistència de la galga degut al canvi en la temperatura d'aquesta cancel·li la variació de la resistència deguda a la dilatació del material sobre el qual esta muntada. Tenint en compte que diferents materials tenen diferents valors d'expansió tèrmica, per a la compensació tèrmica s'ha de seleccionar l'aliatge idoni per al material de l'objecte de test.

En el cas de les galgues que no “auto-compensen” les variacions de temperatura (com per exemple les galgues d'aliatges isoelàstics) és possible corregir-ho fent servir ponts de Wheatstone. Per a aconseguir-ho es necessari muntar un pont format per quatre galgues: dues col·locades a sobre de l'objecte i dues deixades a l'aire, sense suportar els esforços de l'objecte però a la mateixa temperatura que aquest i les galgues que porta aferrades. Per a més detalls consultar l'annex 1.

2.2.2 Transductors comercials

En funció del tipus d'esforç que es vol captar existeixen diferents tipus de transductors que incorporen els sensors de força. En el mercat podem trobar, per exemple, els models següents:

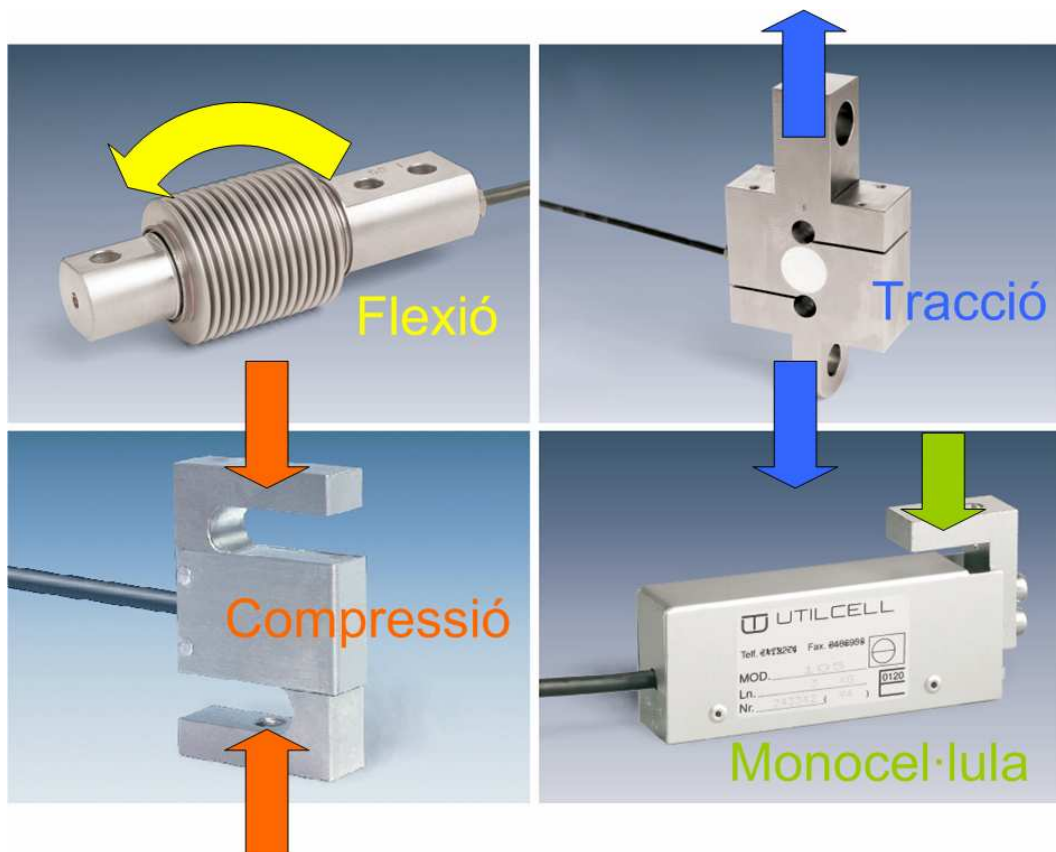


Figura 14: alguns tipus de transductors comercials



2.2.3 Altres tipus de sensors de força

Per a mesurar esforços petits les galgues de semiconductor, també anomenades galgues piezorresistives, són normalment preferides a les galgues metal·liques ja que presenten un “Factor de Galga” (GF) més elevat. Aquests dispositius tenen un altre principi de funcionament: les deformacions aplicades no varien la seva resistència elèctrica sinó que fan que es produeixin diferències de potencial en la seva superfície. Les galgues de semiconductor tendeixen a ser més cares, més sensibles als canvis de temperatura i més fràgils.

En les mesures biològiques, especialment flux sanguini i inflamació de teixits, s’usa una variant anomenada galga de mercuri-en-cautxú. Aquest tipus de sensor està format per una petita quantitat de mercuri líquid a dins d’un petit tub de cautxú que es posa al voltant de la cama o del dit gros del peu. La inflamació d’aquesta part del cos fa que el tub s’estiri, fent-lo més llarg i prim i incrementant la seva resistència elèctrica.

Existeixen galgues senzilles de tipus mecànic usades sobretot en enginyeria civil per a mesurar moviments d’edificis, fonaments i altres estructures. A l’exemple de la Figura 15, les dues meitats del dispositiu estan fortament fixades a la paret d’una construcció, una a cada part del crui.

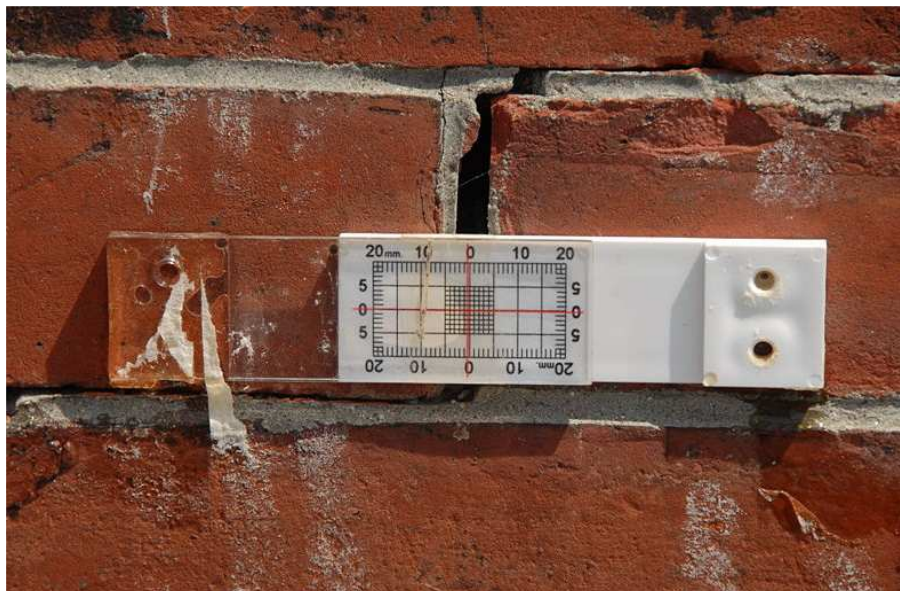


Figura 15: sensor d’esforç mecànic

Les línees vermelles de referència estan a la part transparent i el reixat negre està dibuixat a la part blanca i opaca. Aquest dispositiu ens permet



monitoritzar moviments verticals i horitzontals al mateix temps. A la figura es veu com les dues meitats del crui s'han separat uns 0.3 mm en l'eix horitzontal (i res en el vertical) des de la instal·lació dels dispositiu.



2.3 Amplificadors de Senyal

En aquest apartat s'expliquen diferents mètodes per a amplificar un senyal d'entrada.

2.3.1 Amplificadors d'instrumentació

Un amplificador d'instrumentació és un dispositiu creat a partir d'amplificadors operacionals (per a més informació sobre amplificadors operacionals consulta l'annex 2). Està dissenyat per a tenir una alta impedància d'entrada i un alt rebuig al mode comú (CMRR). Es pot construir a partir de components discrets o es pot trobar encapsulat (per exemple el INA114). L'operació que realitza és la resta de les seves entrades multiplicada per un factor.

La figura següent il·lustra l'estructura d'un amplificador d'instrumentació:

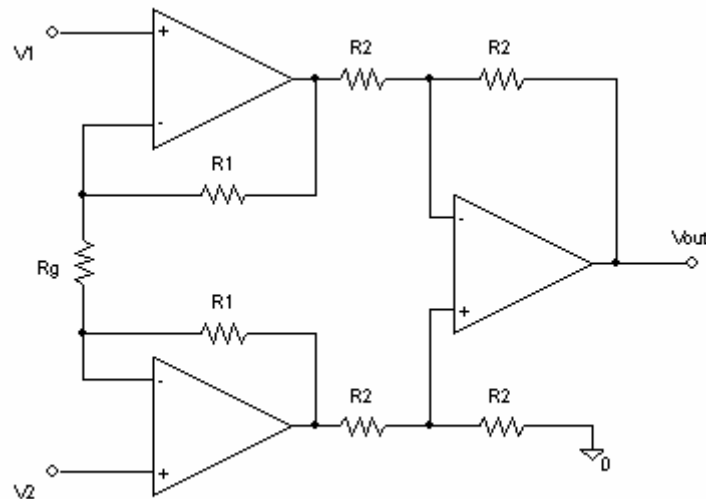


Figura 16: amplificador d'instrumentació

Donat que existeix una realimentació negativa es pot considerar un curtcircuit virtual entre les dues entrades inversora i no inversora (símbols - i + respectivament) dels dos operacionals. Per aquest motiu es tindran les tensions en els terminals i per tant en els extrems de la resistència R_g . D'aquesta manera per R_g circularà una corrent:

$$I_g = (V_2 - V_1) \left(\frac{1}{R_g} \right)$$



Degut a l'alta impedància d'entrada del A.O., aquesta corrent serà la mateixa que travessa les resistències R_1 . Per tant, la tensió que cau en tota la branca formada per R_g , R_1 i R_1 és:

$$V_{\text{int ermitja}} = \frac{(V_2 - V_1)}{R_g} (R_g + 2R_1) = (V_2 - V_1) \left(\frac{R_g}{R_g} + \frac{2R_1}{R_g} \right)$$

Simplificat:

$$V_{\text{int ermitja}} = (V_2 - V_1) \left(1 + \frac{2R_1}{R_g} \right)$$

que és la diferència de tensió entre la sortida immediata dels dos A.O.s (just abans de les R_2).

Donat que la resta del circuit és un restador de guany la unitat, la seva sortida serà exactament la diferència de tensió de la seva entrada (sense afegir guany), la qual s'acaba de definir com:

$$V_{\text{out}} = (V_2 - V_1) \left(1 + \frac{2R_1}{R_g} \right)$$

En cas de voler encapsular tot el circuit integrat s'ha de deixar fora la resistència R_g per a poder controlar el guany.



Figura 17: amplificador d'instrumentació INA114

Es clar que el màxim guany d'aquest tipus d'amplificador ve donat pel valor de les resistències R_1 i R_g . Si volem aconseguir un guany major s'ha d'escollir una R_g petita o fer composicions amb diversos amplificadors.

Entre les aplicacions d'un amplificador d'instrumentació trobem:

- Condicionar la sortida d'un pont de Wheatstone (annex 1)
- Amplificar senyals elèctriques biològiques (per exemple en electrocardiogrames)



- Com a part de circuits per a proporcionar alimentació a corrent constant
- Fonts d'alimentació

2.3.2 Condicionadors de Senyal Programables

Un condicionador de senyal programable és un circuit analògic que permet ajustar la senyal de sortida d'un sensor de pont sencer (pont de Wheatstone).

La senyal analògica del sensor és amplificada pel dispositiu que a més permet una calibratge digital per ajustar el zero (offset), el rang de treball (span) i corregir errors de linearització. Els valors d'aquest calibratge es guarden en registres o en memòria externa i són periòdicament consultats pel dispositiu.

Per a calibrar el dispositiu per a un nou sensor cal connectar aquests i configurar els registres que contenen els valors de l'amplificació que es vol realitzar, i si es desitja, l'"offset" i la linearització.

Alguns condicionadors de senyal de més altes prestacions permeten diferents nivells d'amplificació i de correcció de l'"offset". D'aquesta manera es pot tenir per exemple un nivell tosc ("coarse") d'amplificació i "offset" i un nivell fi ("fine") que depèn de la temperatura del sensor. Així un sensor de temperatura connectat al sensor de pont ens permet corregir derives d'amplificació i ajust de l'"offset". Tenim aleshores un amplificador autocompensat.

Altres prestacions dels condicionadors de senyal programables són la limitació per dalt i per baix de la senyal de sortida (ja amplificada) o la selecció de la tensió d'excitació del sensor.



Figura 18: exemple de condicionador de senyal programable



3 Objectius del projecte i treball a realitzar

L'objectiu principal d'aquest projecte final de carrera és dotar un caminador de sensors de força i electròniques necessàries per a captar les forces a les quals es sotmès. Concretament, les forces que volem captar són les que exerceix l'usuari a les manetes del caminador i la força que fan les rodes posteriors al terra (força normal).

El disseny i implementació dels sensors de força corre a càrrec del departament de Resistència de Materials i Estructures a l'Enginyeria de la UPC.

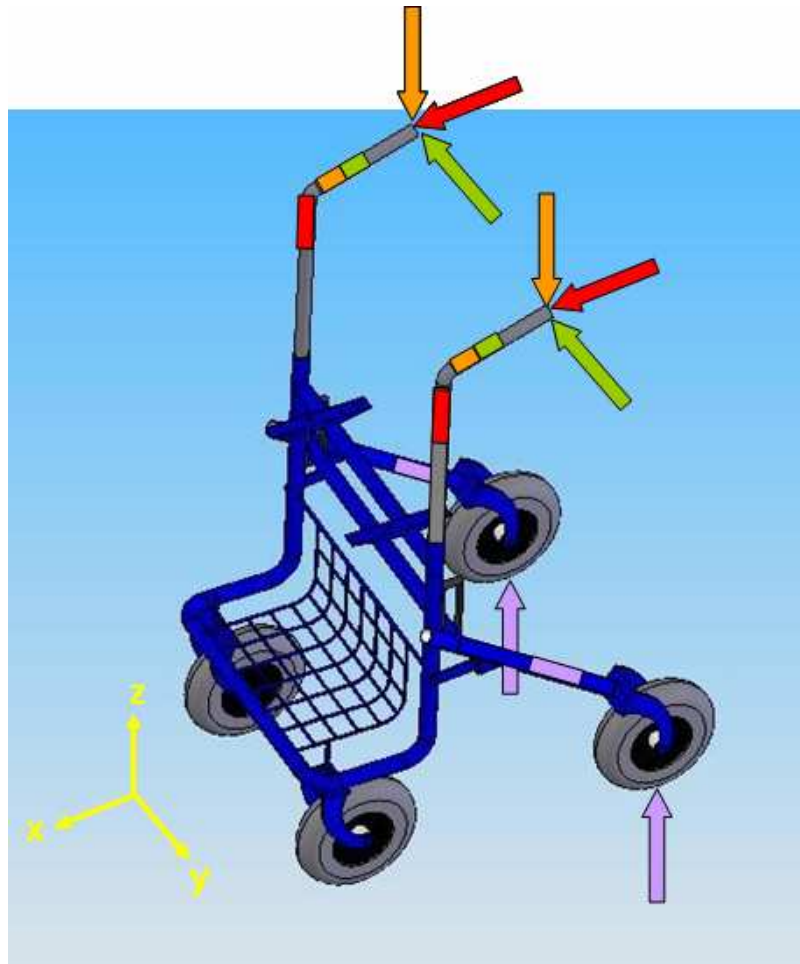


Figura 19: situació dels sensors de força al caminador



Amb els sensors implementats obtenim un senyal analògic proporcional a la força exercida. Les forces captades a cada una de les manetes del caminador són tres (eixos X, Y i Z). Això fa que tinguem un total de 8 forces (dues manetes + dues rodes posteriors).

El senyal de cada un dels sensors de força ha de ser condicionat (possiblement amplificat) i digitalitzat per a que pugui ser comunicat a un computador sense necessitat d'usar una tarja d'adquisició especial. Aquesta comunicació és un altre dels objectius del PFC. El μC integrat en les electròniques de captació ha de publicar en un bus de comunicacions CAN la dada ja digitalitzada.

El darrer dels objectius del projecte és el de realitzar una representació realista en temps real de les dades rebudes per CAN al computador. Aquestes dades seran prèviament tractades per tal d'obtenir el valor de la força en Newtons i/o kiloponds.



4 Mesura de Forces

Com ja s'ha mencionat, els sensors de força de que disposa el caminador han estat dissenyats i implementats pel departament de Resistència de Materials i Estructures en l'Enginyeria. Veurem ara en que es basa el funcionament d'aquests sensors.

4.1 Principis de l'extensometria

L'extensometria és una tècnica experimental per a la mesura d'esforços i deformacions basada en el canvi de la conductivitat elèctrica d'un material quan se li aplica una força. Degut a la recent introducció del mètode d'elements finits, aquesta tècnica és menys utilitzada per la seva complexitat encara que sigui molt més exacta.

És experimentalment útil en la mesura d'esforços vibracionals i detecció de ressonàncies a altes freqüències (la resposta en freqüència d'una galga extensomètrica és d'uns 100 kHz) on els algorismes de simulació per elements finits no ofereixen resultats fiables (aquests solen començar a mostrar imprecisions amb sistemes complexos que vibren a més de 50 Hz).

En les seves múltiples variants permet determinar estats tensionals, unidireccionals o complets, mesurar deformacions a alta temperatura (fins uns 800 °C amb bandes soldables), controlar obra civil (galgues de formigó) i fabricar acceleròmetres extremadament sensibles capaços de mesurar camps continus.

El seu ús requereix un amplificador analògic d'elevades prestacions degut a les dèbils senyals que es generen (sovint inferiors a 1mV) i un filtre "antialiasing" (per a sistemes amb registre digital del senyal).

4.2 Usant el mètode d'elements finits

Aquest mètode numèric ens permet aproximar solucions d'equacions diferencials parcials. Es basa en dividir el cos, estructura o domini (medi continu) -sobre el qual estan definides certes equacions integrals que caracteritzen el comportament físic del problema- en una sèrie de subdominis que no intersequen entre sí anomenats "elements finits". El conjunt d'elements finits forma una partició del domini també coneguda com a



discretització. Dins de cada element es distingeixen una sèrie de punts representatius anomenats nodes. Dos nodes són adjacents si pertanyen al mateix element finit; a més un node situat sobre la frontera d'un element finit pot pertànyer a diferents elements finits. El conjunt de nodes considerant les relacions d'adjacència s'anomena malla.

Els càlculs es realitzen sobre una malla o discretització creada a partir del domini amb programes especials anomenats generadors de malles, en una etapa prèvia als càlculs anomenada pre-procés. D'acord amb aquestes relacions d'adjacència o connectivitat es relaciona el valor del conjunt de variables incògnites definides en cada node i denominades graus de llibertat. El conjunt de relacions entre el valor d'una determinada variable entre els nodes es pot escriure en forma de sistema d'equacions lineals (o linealitzades). La matriu d'aquest sistema d'equacions s'anomena matriu de rigidesa del sistema. El nombre d'equacions d'aquest sistema és proporcional al nombre de nodes.

En l'estudi de les forces que actuen sobre un cos, el mètode dels elements finits es típicament programat computacionalment per a calcular el camp de desplaçaments i, posteriorment, mitjançant relacions cinemàtiques i constitutives, les deformacions i tensions.

Aquest mètode és molt utilitzat en aquesta i altres practiques de l'enginyeria quan es té la impossibilitat pràctica de trobar una solució analítica. Gracies a la propietat de la convergència, si es consideren particions d'elements finits cada cop més fines, la solució calculada convergeix ràpidament cap a la solució exacta del sistema d'equacions.

4.3 Disseny dels transductors pel i-Walker

Els transductors implementats per al "intelligent Walker" permeten obtenir una completa informació de la interacció de l'usuari amb el caminador. La major part d'aquesta interacció es fa mitjançant les manetes. Una altra data important és la força que exerceixen les rodes del caminador a sobre del terra.



Les forces que es fan a sobre de les manetes poden ser en qualsevol direcció, així que és necessari separar-les en cada una de les components espacials segons el sistema de referència següent:



Figura 20: sistema de referència

Les forces en l'eix X són captades per un transductor muntat en la secció vertical de cada una de les manetes. Les forces en els eixos Y i Z són captades per un transductor muntat en la barra horitzontal de les manetes. Les forces a les rodes només poden ser verticals, així que només tenen component Z. Per a mesurar la magnitud d'aquesta component a cada una de les rodes posteriors s'ha instal·lat un transductor en la barra que les subjecte (veure Figura 19).

Els transductors implementats per a la mesura de les forces aconsegueixen les següents condicions:

- Es respecte la geometria funcional original del caminador
- Introdueixen canvis mínims en l'estructura
- Permeten la mesura simultània i per separat de les 8 components de força
- Tenen resposta dinàmica i estàtica



4.3.1 Transductor per a la component X (manetes)

El disseny del cos del transductor es pot veure a la figura següent:

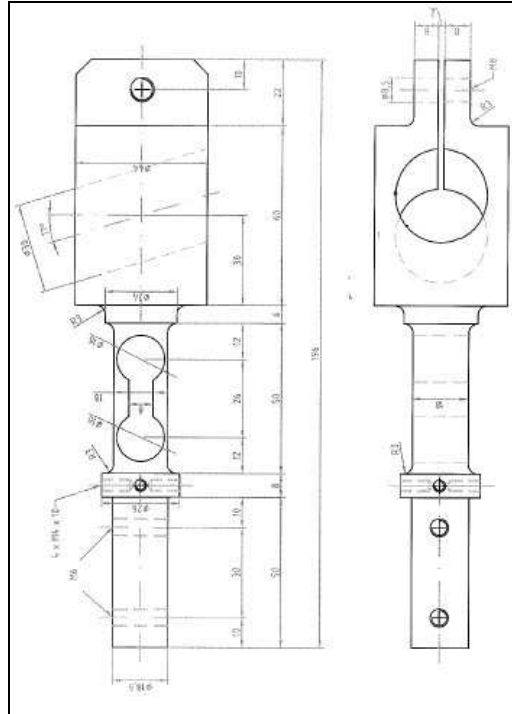


Figura 21: transductor per a la component X

Quan una força en l'eix X actua, el cos del transductor es doblega. Les galgues estensomètriques han estat col·locades en els punts on les tensions són més elevades.

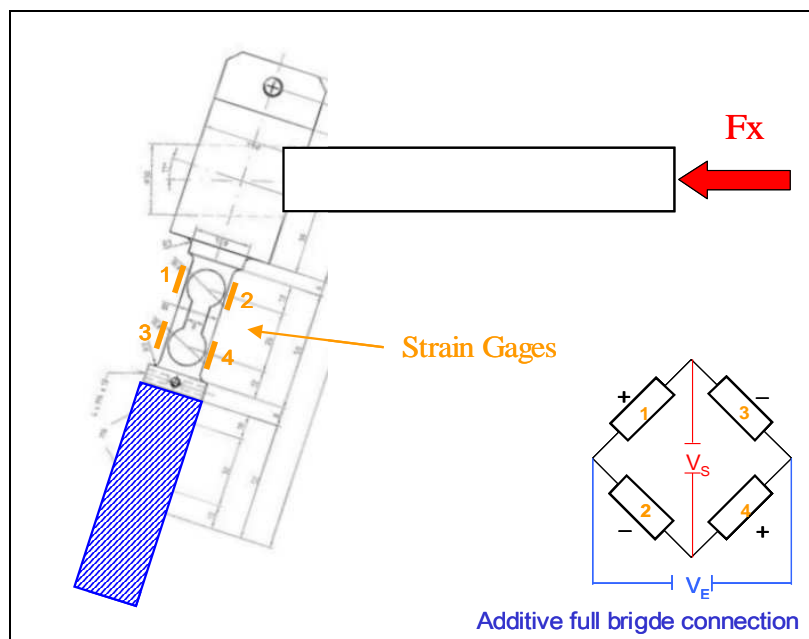


Figura 22: col·locació de les galgues extensomètriques



Per a optimitzar el disseny del transductor s'ha creat i analitzat un model d'elements finits del seu cos.

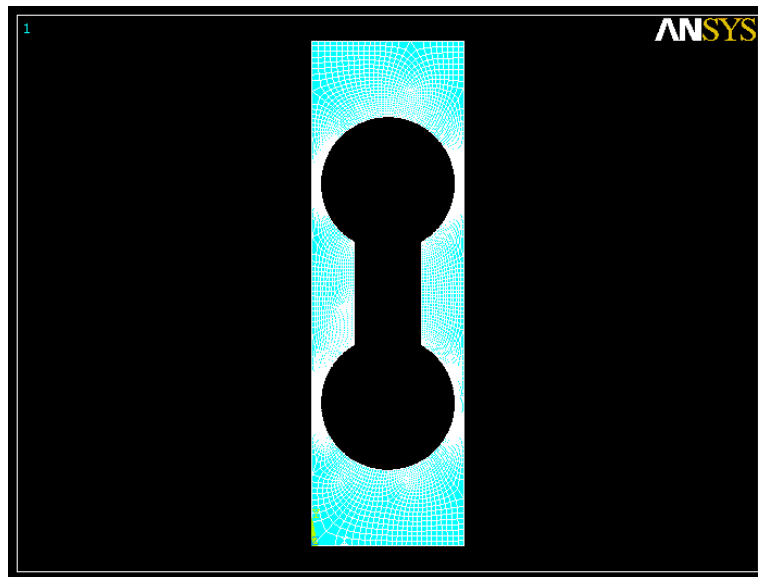


Figura 23: model d'elements finits

La de formació i la distribució de les tensions obtingudes es mostren a la figura següent:

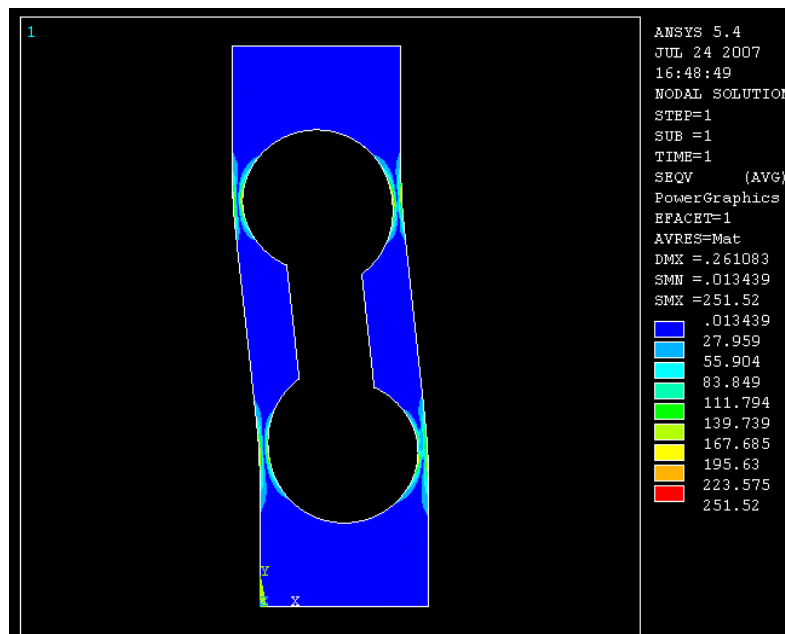


Figura 24: tensions de Von Mises

Les galgues extensomètriques s'han connectat en “additive full bridge”: el senyal de cada una de les quatre galgues contribueix amb el mateix signe a la sortida del pont.

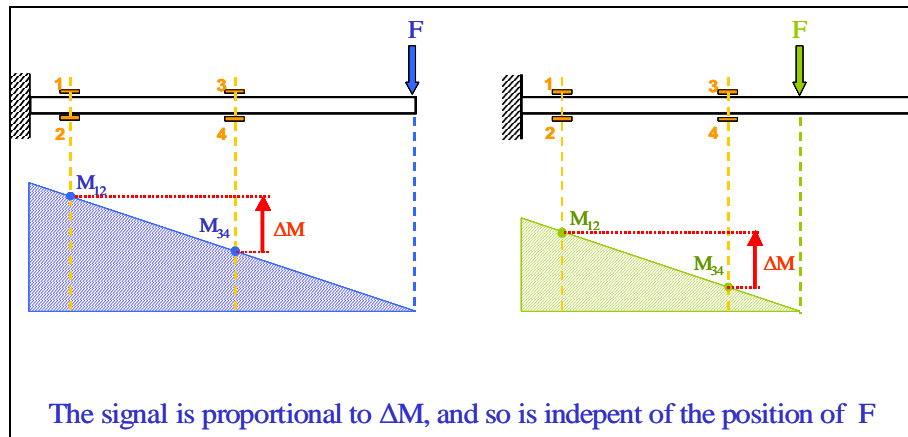


Figura 27: independència de la posició de la força F

Per a mesurar la força en Y les galgues extensomètriques s'ha col·locat en el pla horitzontal, i per a mesurar la força en Z, en el pla vertical.

4.3.3 Transductor per a la component Z (rodes posteriors)

El disseny del transductor per a mesurar la força Z en les rodes posteriors en pot veure a la figura següent:

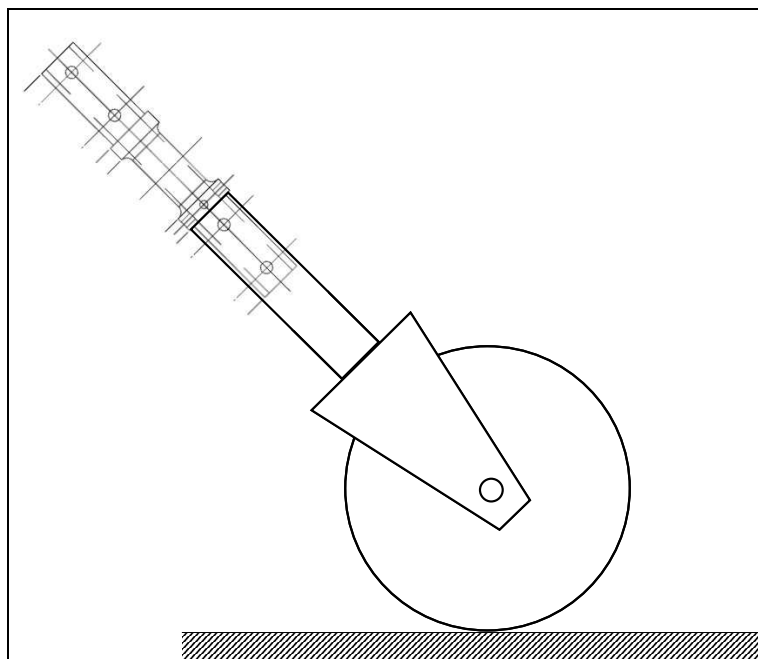


Figura 28: transductor per a la component Z



El cos esta format per una tub cilíndric que substitueix una part del tub original de suport de cada una de les rodes posteriors. Quan una força Z (la reacció del terra a sobre de la roda) actua, el cos del transductor es doblega.

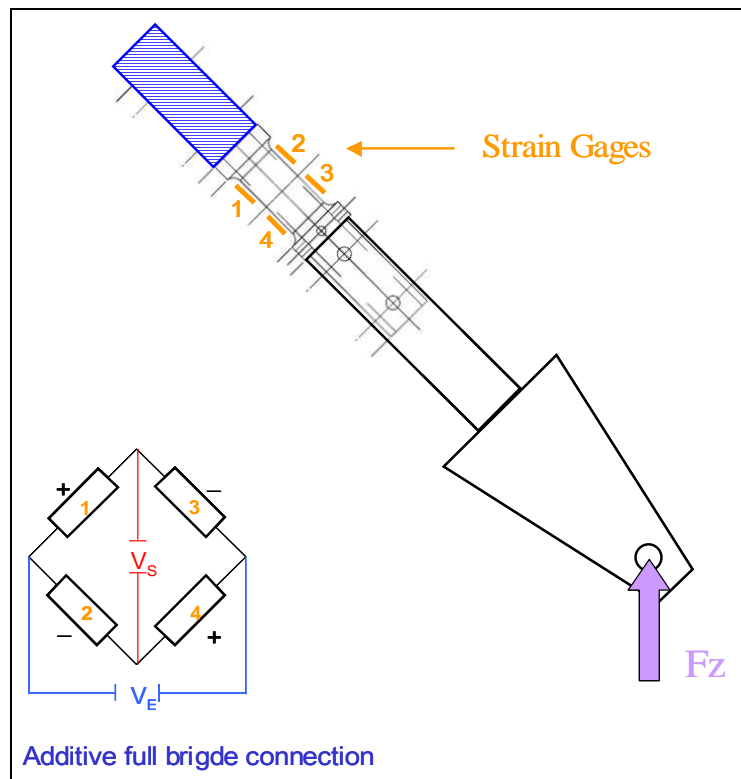


Figura 29: col·locació de les galgues extensomètriques

Les galgues s'ha connectat en forma de "additive full bridge".



5 Calibratge dels Sistemes de Mesura de Força

El calibratge dels sensors de força l'ha portat a terme la gent del mateix departament que s'ha encarregat del seu disseny i implementació. A continuació veurem com ho han fet així com els resultats que han obtingut.

5.1 Mètode de calibratge

Tots els transductors de han estat calibrats fent servir pesos per a generar les forces. Els pesos han estat prèviament controlats per a aconseguir una precisió més gran del 0.1%.

Els transductors han estat connectats a amplificadors de dos canals. D'aquesta manera tots els senyals s'ha mesurat sempre de manera independent i simultània. Les galgues extensomètriques han estat excitades a 5V de corrent continua.



Figura 30: banc de treball usat per al calibratge dels sensors



5.2 Calibratge dels transductors de les manetes

Els transductors per a la component X han estat calibrats només en un sentit: captació de la força d'empenta del caminador. En l'altre sentit el comportament es suposa simètric. Els transductors han estat calibrats fins a 100 N de força.

Els transductors per a les components Y i Z han estat calibrats aplicant el pes en els plans verticals i horitzontals respectivament. Aquests transductors han estat calibrats en els dos sentits (positiu i negatiu), fins a 200 N. La independència del senyal de sortida del punt en la qual s'aplica la força ha estat confirmat experimentalment.



Figura 31: calibratge dels sensors de les manetes



5.3 Calibratge dels transductors de les rodes

Els transductors per a captar la força que es fa a les rodes han estat calibrats un cop muntats en el caminador. El caminador té quatre rodes i aquestes disposen de pneumàtics de goma. Aquests dos fets fan que s'introdueixi una incertesa en la força aplicada en cada una de les rodes. Després d'una sèrie d'intents experimentals, els resultats més consistents s'han obtingut amb el muntatge següent: la part del davant del caminador es aguantada per un únic punt central amb una unió esfèrica, les rodes posteriors són extrems i el caminador es suspèn pels eixos de les rodes amb una barra horitzontal que disposa del sensor de força.

Aquests transductors han estat calibrats només en el sentit positiu, fins a 220 N.



Figura 32: muntatge usat pel calibratge del sensor de les rodes



5.4 Dades dels transductors

5.4.1 Manetes

Els transductors de les manetes han estat calibrats per a determinar la seva sensibilitat i linealitat. En la taula següent es resumeixen les dades de la seva capacitat i linealitat:

Component	Nominal force (N)	Linearity error (%)
F1x	100	< 1
F2x	100	< 1
F1y	200	< 1
F2y	200	< 1
F1z	200	< 1
F2z	200	< 1

Els senyals dels tres transductors en cada una de les manetes (L_i) estan relacionats amb les tres forces que hi actuen (F_i) per una matriu G .

$$\{L\} = [G] \cdot \{F\}$$

$$\begin{Bmatrix} L_x \\ L_y \\ L_z \end{Bmatrix} = \begin{bmatrix} G_{xx} & G_{xy} & G_{xz} \\ G_{yx} & G_{yy} & G_{yz} \\ G_{zx} & G_{zy} & G_{zz} \end{bmatrix} \times \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}$$

Els components de la matriu G s'obtenen experimentalment en el calibratge dels sensors. Aleshores, invertint aquesta matriu, la matriu K resultant ens permet calcular les tres forces (F_i) a partir dels tres senyals dels transductors (L_i).

$$[K] = [G]^{-1}$$

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix} \times \begin{Bmatrix} L_x \\ L_y \\ L_z \end{Bmatrix}$$



Idealment, la matriu K hauria de ser una matriu diagonal, de manera que cada un dels transductors fos totalment insensible a les forces creuades.

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \begin{bmatrix} K_{xx} & 0 & 0 \\ 0 & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{bmatrix} \times \begin{Bmatrix} L_x \\ L_y \\ L_z \end{Bmatrix}$$

En els transductors de les manetes del i-Walker la matriu K es quasi diagonal. Això significa que alguns càlculs han de ser realitzats pel computador per a separar els tres components de la força.

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \begin{bmatrix} 49,371 & 0 & -89,830 \\ 0 & 290,452 & 0 \\ 0 & 0 & 286,006 \end{bmatrix} \times \begin{Bmatrix} L_x \\ L_y \\ L_z \end{Bmatrix}$$

$$(N) \qquad \left(\frac{N}{mV/V} \right) \qquad (mV/V)$$

5.4.2 Rodes posteriors

Les dades de la capacitat, sensibilitat i linealitat dels transductors de les rodes posteriors es resumeixen en la taula següent:

Component	Nominal force (N)	Sensitivity (mV/V)	Linearity error (%)
F3z	200	1,166	< 0,6
F4z	200	1,134	< 0,6

La lectura que ens dona és el valor en milivolts -per cada volt d'excitació- que dona el sensor quan es sotmès a una càrrega de 200 N. També ens indica un error en la linealitat d'ambdós sensors menor al 0.6%.





6 Amplificació de la mesura

Els sensors de força que fem servir són del tipus “pont sencer” (Weatstone full-bridge). Com ja hem vist, la senyal de sortida d’aquestes galgues ha de ser amplificada si volem tenir resolució en la mesura. Per a aquesta amplificació fem servir un condicionador de senyal programable que ens permetrà configurar el senyal amb el guany desitjat i corregir l’“offset” en cas de que sigui necessari. El model escollit és el PGA309.

El PGA309 és un condicionador de senyal complet que proporciona excitació del pont, ajust del rang i “offset” inicials, ajust del rang i “offset” en funció de la temperatura, mesura de la temperatura interna o externa, limitació de la sortida per sobre i per sota, detecció d’errors i calibratge digital. La Figura 33 mostra els diagrama de blocs bàsic del PGA309.

6.1 Escalat del guany

El cor del PGA309 es l’amplificador d’entrada de guany programable (Front-End PGA). El guany total del Front-End PGA més l’amplificador de sortida (Output Amplifier) pot ser ajustat des de 2.7V/V fins 1152V/V. La polaritat de les entrades pot ser canviada usant un multiplexor per tal de connectar fàcilment sensors de pont amb polaritat de sortida desconeguda.

El Front-End PGA proporciona el guany tosc (“coarse”) inicial usant un amplificador d’instrumentació. El guany fi (“fine”) s’aconsegueix amb un convertidor digital-analògic de guany de 16 bits (Gain DAC). El Gain DAC usa com a entrada les dades guardades en una taula de compensació de temperatura accedida per un convertidor digital-analògic de temperatura (Temp ADC). Amb l’objectiu de compensar les no-linealitats de segon ordre, el guany fi es establert fent servir els coeficients guardats a una memòria externa no volàtil (EEPROM).

A continuació del guany fi ve la etapa de l’amplificador de sortida (Output Amplifier) que proporciona un guany programable addicional. Les dues connexions principals de l’amplificador de sortida, V_{FB} i V_{SJ} , són tretes del PGA309 per més flexibilitat de les aplicacions. Aquestes connexions permeten obtenir un voltatge del senyal condicionat molt acurat, controlar els pics de sobrevoltatge i filtrar RFI/EMI.

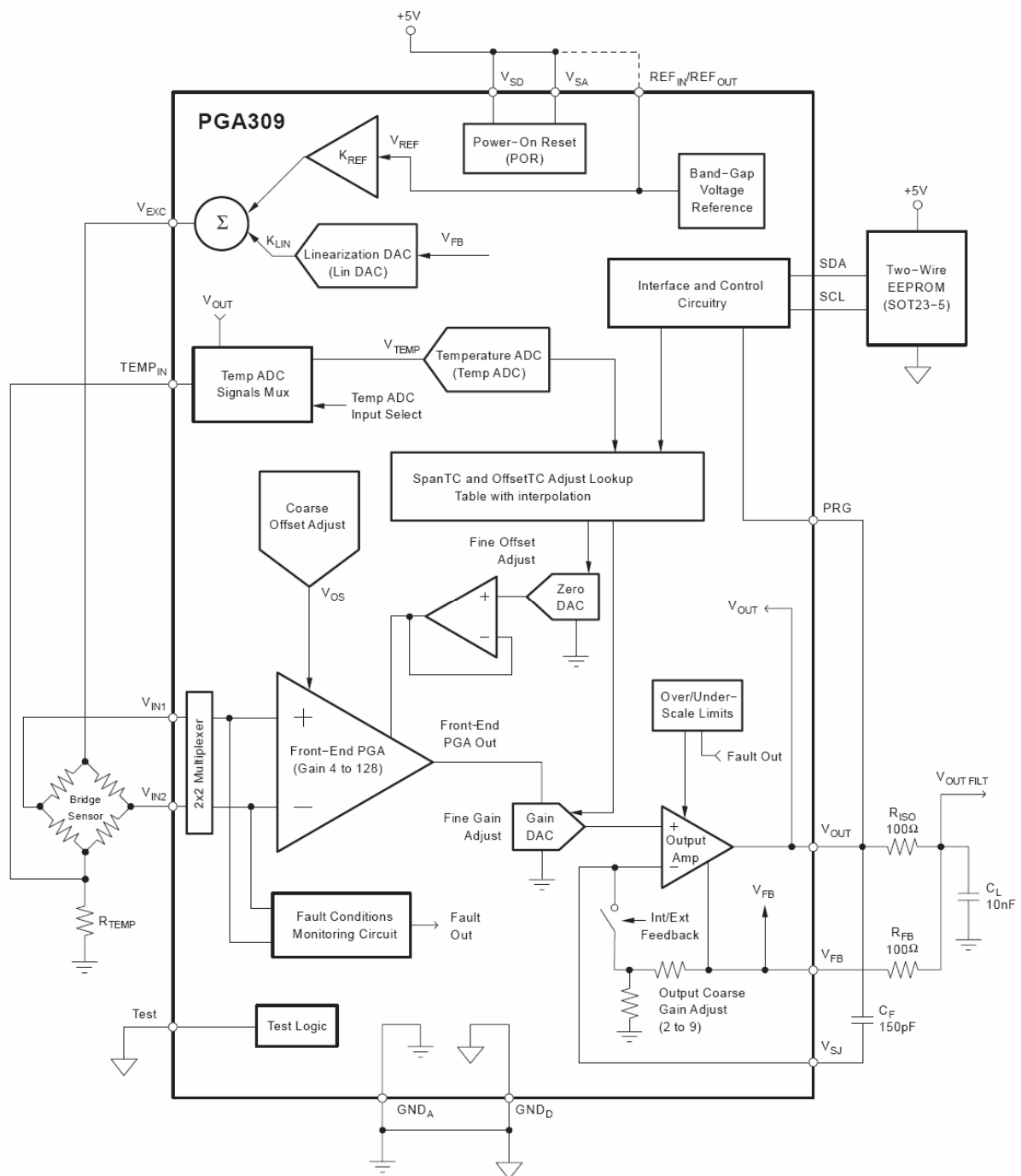


Figura 33: diagrama de blocs del PGA309



6.2 Ajust de l'offset

L'ajust de l'offset del sensor es fa en dues fases. L'ajust tosc té aproximadament $\pm 60\text{mV}$ d'ajust per a un voltatge de referència (V_{REF}) de 5V. L'ajust fi es realitza amb un convertidor digital-analògic de 16 bits (Zero DAC) que suma la senyal a la sortida del Front-End PGA. De manera anàloga al Gain DAC, els valors digitals d'entrada al Zero-DAC estan emmagatzemats a la taula de compensació de temperatura guardada en la EEPROM externa i adreçada pel TempADC. El rang programable del Zero DAC és de 0V a V_{REF} , amb un rang de sortida de 0.1V a $V_{\text{SA}} - 0.1\text{V}$.

6.3 Voltatge de referència

El PGA309 proporciona un voltatge de referència d'alta precisió i baixa oscil·lació (2.5V o 4.096V) que pot ser usat per a circuits externs usant els pins $\text{REF}_{\text{IN}}/\text{REF}_{\text{OUT}}$. Aquesta mateixa referència és usada pel convertidor de l'offset tosc, Zero DAC, limitador de sobre/sota escala i excitació de sensor/linearització mitjançant el pin V_{EXC} . Quan la referència externa es deshabilitada, les pins $\text{REF}_{\text{IN}}/\text{REF}_{\text{OUT}}$ han de ser connectats a una referència externa o a V_{SA} per a sistemes escalats radiomètricament.

6.4 Excitació del sensor i linearització

Un circuit dedicat amb un convertidor digital/analògic de 8 bits (7 bits + signe) es contingut en el PGA309 per tal de proporcionar voltatge d'excitació al sensor i linealitzar. Aquest bloc escala el voltatge de referència i suma a aquest una porció de la sortida del PGA309 per tal de compensar les no-linealitats que presenten molts sensors al llarg del seu rang de treball. Sensors que no requereixin linealització poden ser connectats directament a l'alimentació (V_{SA}) o al pin V_{EXC} amb el DAC de linealització (Lin DAC) a zero.



6.5 ADC per al sensor de temperatura

El guany i “offset” fins són ajustats fent servir un circuit dedicat a la mesura de la temperatura. Es poden seleccionar entre els modes de detecció de temperatura interna o externa. La temperatura pot ser mesurada d’una de les següents maneres:

- Canvi de la impedància del pont (sensor del corrent d’excitació, en la part positiva o negativa del pont), per a sensors amb un alt coeficient de resistència a la temperatura ($TCR > 0.1\% / ^\circ C$).
- Temperatura del xip PGA309, quan el xip està suficientment pròxim al sensor.
- Diode extern, thermistor o RTD col·locat en la membrana del sensor. Una font interna de $7\mu A$ de corrent ha de ser habilitada per tal de excitar aquests tipus de sensor de temperatura.

La senyal de la temperatura és digitalitzada fent servir el Temp ADC. La sortida d’aquest convertidor és usada pel circuit de control digital per llegir les dades de la “Lookup Table” guardada en la EEPROM externa, i establir la sortida del Gain DAC i el Zero DAC segons els valors calibrats en funció de la temperatura.

Una funció addicional proporcionada pel Temp ADC es l’habilitat de llegir el pin V_{OUT} pel seu multiplexor d’entrada. Això proporciona flexibilitat per a una sortida digital mitjançant les interfícies de comunicacions del xip, així com també proporciona la possibilitat de que un microcontrolador extern dugui a terme el calibratge en temps real del PGA309.

6.6 EEPROM externa i coeficients de temperatura

El PGA309 usa una EEPROM externa de connexió Two-Wire, estàndard en el món de la indústria. Es necessària una EEPROM mínima de 1 K-bit quan s’usen tots el 17 coeficients de temperatura. Una EEPROM més gran pot ser usada per proporcionar espai a l’usuari per tal d’emmagatzemar el nombre de sèrie, el codi del lot i altres dades.

La primera part de la EEPROM externa conté les dades de configuració per al PGA309:

- Registre 3: Control de referència i linealització



- Registre 4: “Offset” TOSC del PGA i guany del Front-End PGA i Output Amplifier
- Registre 5: Configuració del PGA i límits de sobre/sota escala
- Registre 6: Control del Temp ADC

Aquesta secció de la EEPROM conté el seu propi “checksum” (Checksum1).

La segona part de la EEPROM externa conté fins a 17 índexs de temperatura i els corresponents coeficients per al Zero DAC i el Gain DAC; i conté el seu propi “checksum” (Checksum2). La lògica del PGA309 conté un algorisme d’interpolació per ajustar els valors dels DACs entre els índexs de temperatura guardats. D’aquesta manera s’aconsegueix una compensació de la temperatura a partir dels 17 índexs de temperatura i els seus coeficients associats.

Si el Checksum1 i/o Checksum2 són incorrectes la sortida dels PGA309 es manté en alta impedància.

6.7 Monitorització d’errors

Per a detectar errors del sensor o curtcircuits, un conjunt de quatre comparadors estan connectats a les entrades del Front-End PGA. Si alguna de les entrades es troba a menys de 100mV de terra o de V_{EXC} , o es viola el CMR de l’entrada del Front-End PGA, aleshores el comparador corresponent activa un flag d’error que causa que la sortida del PGA309 es situï a menys de 100mV de V_{SA} o terra, en funció de la configuració d’alarma establerta (Registre 5 - Configuració del PGA i limitació de sobre/sota escala). Aquest valor estarà o bé per sobre del nivell de sobre escala o bé per sota del nivell de sota escala. La condició d’error pot ser llegit en format digital al Registre 8 - Registre de l’estat de l’alarma. Si es deshabilita el limitador de sobre/sota escala, la sortida del PGA309 es situarà a menys de 100mV de V_{SA} o terra, depenent de la configuració de l’alarma.

Hi ha cinc comparadors més per a ajudar a detectar petites violacions a l’entrada del PGA309 que podrien provocar sortides a V_{OUT} interpretades com a vàlides. Aquests són especialment útils durant el calibratge inicial i es configuren mitjançant el Registre 5 - Configuració del PGA i limitació de sobre/sota escala. El seu estat també pot ser llegit al Registre 8 - Registre de l’estat de l’alarma.



6.8 Limitació sobre-escala i sota-escala

El circuit de limitació de la sortida combinat amb el de detecció de falles proporcionen una manera de fer una diagnosi del sistema. Una sortida típica del sensor condicionat pot ser escalada entre el 10% i el 90% del rang de ADC del sistema quan el sensor opera normalment. Si el sensor de pressió es troba per sota del 4%, es considera a “infrapressionat”; i si es troba per sobre del 96%, es considera “sobrepessionat”.

El circuit de limitació del PGA309 pot ser programat individualment per limitar a sota-escala i a sobre-escala. Des de un punt de vista de diagnosi, el rang del 10% al 90% del ADC es considera normal, <4% es considera “infrapressió”, i >96% es considera “sobrepessió”. Si s’usa el sistema de detecció de falles, una detecció provocarà que la sortida del PGA309 es satura positiva o negativament. Si el flag de fallada es programat com a “alt”, >97% del rang de ADC es considerarà fallada; i si es programat com a “baix”, <3% del rang de ADC es considerarà fallada. Aleshores el programari pot ser usat per a distingir entre “sobrepessió” i “infrapressió”, que indica procés fora de control o fallada del sensor.

6.9 Interfície digital

Hi ha dues interfícies digitals en el PGA309. El pin PRG usa una interfície “One-Wire” compatible amb UART, que pot funcionar entre 4.8 Kbits/s i 38.4 Kbits/s. Els pins SDA i SCL junts conformen una interfície estàndard “Two-wire” (I²C) amb freqüències de rellotge entre 1 kHz i 400 kHz. La EEPROM externa usa aquesta segona interfície. Les comunicacions amb els registres interns del PGA309, així com amb la EEPROM externa, per a programar-los o consultar-los es poden realitzar mitjançant qualsevol de les dues interfícies.

És possible connectar el pin PRG a V_{OUT} per tal de dur a terme el calibratge del sensor fent una realimentació. En aquest mode l’amplificador de sortida del PGA309 ha de ser habilitat per a la seva configuració i després deshabilitat per a permetre la compartició del pin PRG amb la connexió a V_{OUT}.

La interfície “Two-Wire” (I²C) té un mecanisme de comptador per a prevenir col·lisions al bus. Quan el controlador “master” del PGA309 vol iniciar una comunicació per aquesta interfície i la línia SDA (dades) està ocupada s’inicia un comptador que espera la seva alliberació. Quan el



comptador acaba s'intenta alliberar el bus emetent polsos per la línia SCL ("clock") encara que el bus encara no estigui en estat "idle". (Per a més detalls d'aquest protocol consultar l'apartat 7.1 (Interfície PGA - Microcontrolador (I²C))).

6.10 Mode normal de funcionament

Quan la tensió d'alimentació supera els 2.2 V s'esborra el valor de tots els registres i es deixen amb el seu valor inicial. Després d'aproximadament 25 ms el PGA309 intenta llegir les dades de configuració (registres 3 a 6) emmagatzemats a la primera part de la EEPROM. Si la EEPROM conté la marca correcta (0x5449, "TI" a l'alfabet ASCII) a les adreces 0 i 1, el PGA continua llegint la EEPROM. En cas contrari, el PGA309 esperarà un segon abans d'intentar-ho de nou. Si el PGA309 detecta que no hi ha resposta de la EEPROM i el bus "Two-Wire" (I²C) no es troba ocupat (SDA = 1 i SCL = 1) aleshores el PGA espera un segon i ho intenta de nou. Si el bus es troba ocupat el PGA intenta alliberar-lo tal i com s'ha explicat a l'apartat anterior i espera 25 ms abans d'intentar llegir la EEPROM de nou. Si la configuració guardada a la EEPROM es llegeix sense problemes (i el Checksum1 és vàlid) el PGA309 consulta al Temp ADC la mesura de la temperatura (si així esta configurat). El resultat de 16 bits triga aproximadament 125 ms a completar la conversió. Un cop finalitzada, el PGA309 comença la lectura de la segona part de la EEPROM ("Lookup Table"), a partir de la posició 16, per tal de calcular els valors de la del Gain i Zero DACs usant l'algorisme d'interpolació. Un cop s'ha llegit tota la taula es determina si el Checksum2 és correcte. Si és així es calculen els valors del Gain i Zero DACs i s'habilita l'amplificador de sortida (V_{OUT}). Un cop arribat a aquest punt el procés torna a començar repetint-se indefinidament.

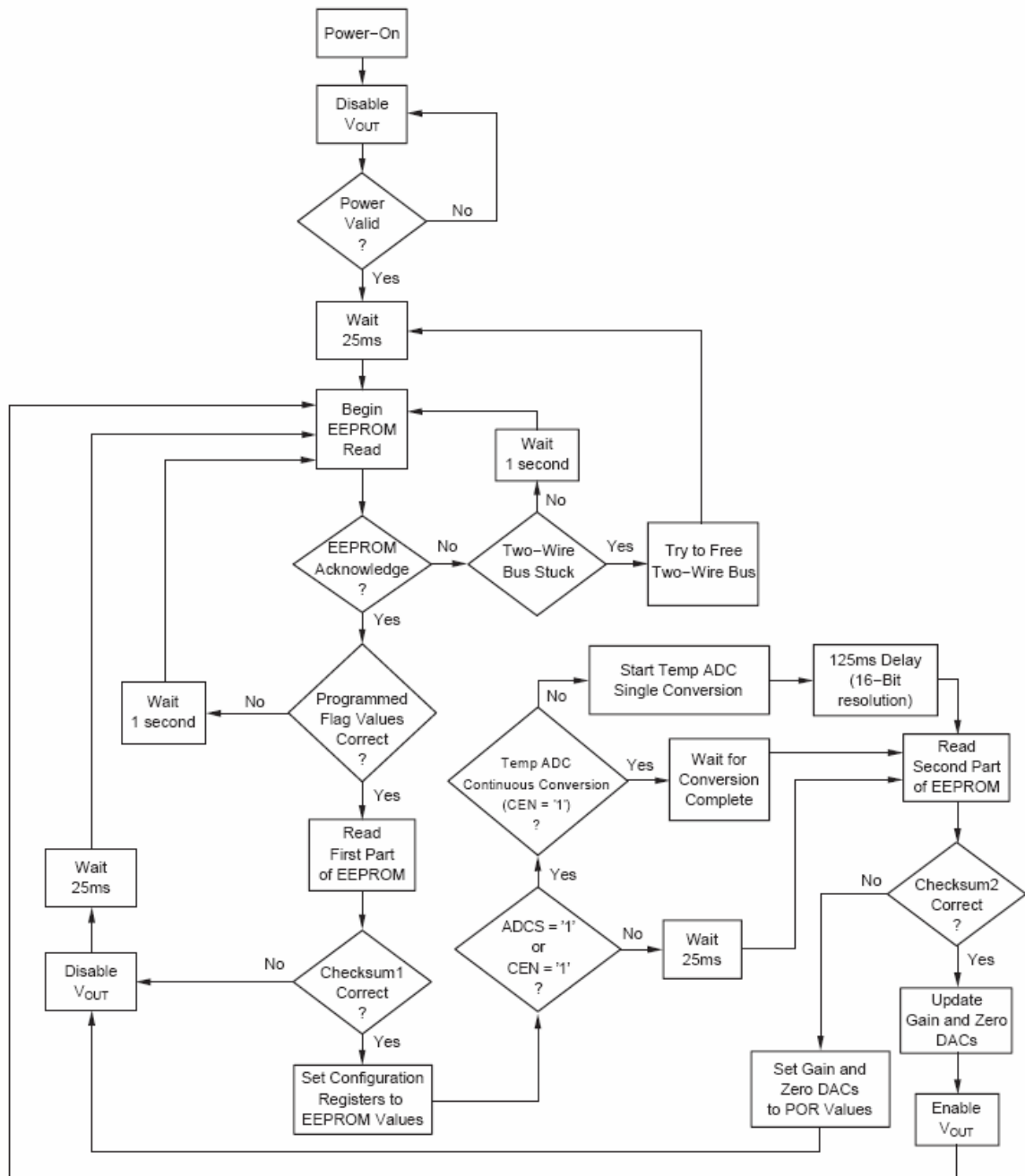


Figura 34: màquina d'estats del funcionament del PGA309



6.11 Pin de test

El PGA309 té un pin que permet activar el mode de test (pin 9). Posant un '1' lògic al pin es para la màquina d'estats del PGA, mantenint habilitada la sortida V_{OUT} . Aquest mode es pot usar per fer el diagnòstic de la configuració inicial ja que la EEPROM externa deixa de ser llegida i la configuració escrita en els registres del PGA no es sobreescrita. En el mode normal de funcionament el pin de test s'ha de connectar a terra ('0' lògic). Quan es posa un '1' lògic al pin de test passa el següent:

- La màquina d'estats abans d'escrita es para i torna al seu estat inicial. Tota transacció amb la EEPROM es interrompuda i el bus I^2C és alliberat.
- S'habilita la sortida del PGA309 (V_{OUT}).
- Tots els registres interns conserven el seu valor actual.
- Un controlador extern pot modificar qualsevol dels registres del PGA309 que permetin la seva escriptura usant la interfície "One-Wire" o "Two-Wire".



7 Disseny i implementació del sistema Microcontrolador

Per al nostre sistema farem que un microcontrolador emuli la EEPROM. D'aquesta manera, quan el PGA309 demani la seva configuració, el μC contestarà com si de la EEPROM es tractés. Això ens permet canviar fàcilment la configuració del PGA simplement canviant el programa del μC , a més d'estalviar la EEPROM.



Figura 35: emulació de la EEPROM amb un μ Controlador

Després de la configuració inicial (en cas de que tot hagi anat bé) el PGA309 habilita la seva sortida amb l'amplificació del senyal provinent de la galga. Aquesta, encara en format analògic la digitalitzarem usant un convertidor A/D de 16 bits, aconseguint així una alta resolució. Per tal d'aprofitar la interfície I²C del μC escollim com a convertidor el ADS1100. Aquest xip publica la dada ja digitalitzada en dues trames de 8 bits usant el protocol I²C. L'esquema resultant és el següent:

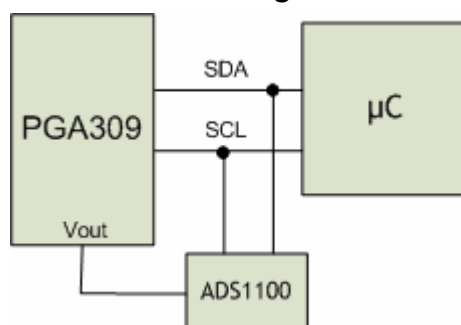


Figura 36: connexió del convertidor A/D

El fet d'usar el bus I²C per a configurar el PGA i per a llegir el A/D ens pot provocar algun problema de col·lisions al bus. A més cal tenir present que, segons em vist al diagrama d'estats del funcionament del PGA309, un cop s'ha portat a terme una lectura completa de la EEPROM (emulada pel μC) es torna a començar una nova lectura, alliberant el bus I²C un període molt curt de temps. De moment deixem aquest problema per més endavant.



7.1 Interfície PGA – Microcontrolador (I²C)

Dissenyat per Philips, I²C és un sistema d'intercanvi d'informació que, mitjançant tan sols dos cables, permet a microcontroladors i perifèrics de sistemes encastats (Embedded Systems) interactuar entre sí a velocitats relativament lentes (100 o 400 kbit/s, encara que també existeix una versió d'alta velocitat de 3.4 Mbit/s). Generalitzant, permet comunicar circuits integrats que normalment resideixen al mateix circuit imprès (PCB).

La principal característica de I²C ve donada pel fet d'usar tan sols dos fils per a transmetre la informació: per un van les dades i per l'altre la senyal de rellotge que serveix per a sincronitzar-los. També es necessita una tercera línia que s'usa com a referència (massa). Com aquest bus es sol fer servir per a comunicar circuits de la mateixa placa que a més comparteixen la mateixa massa, aquesta tercera línia sol ser innecessària.

Les línies són:

- SDA (System Data): viatgen les dades entre els dispositius.
- SCL (System Clock): transiten els polsos de rellotge per a sincronitzar el sistema.
- GND: massa

Les dues primeres son línies de drenador obert, i per tant necessiten resistències de pull-up.

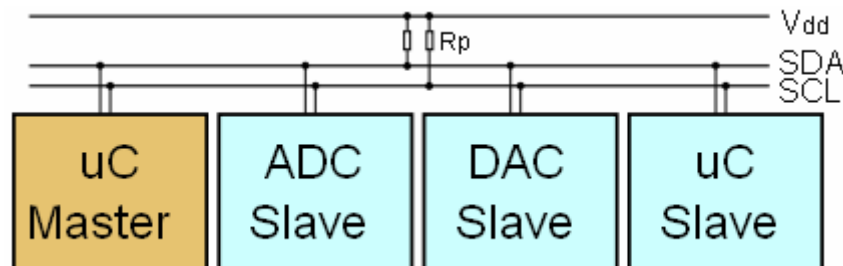


Figura 37: exemple de dispositius connectats a un bus I²C

Els dispositius connectats al bus I²C tenen un adreça única per a cada un d'ells. També poden ser mestres (master) o esclaus (slave). El dispositiu mestre és qui inicia la transferència i a més genera la senyal de rellotge. No és necessari que el mestre sigui sempre el mateix dispositiu sinó que aquesta característica pot anar passant per diferents dispositius que tinguin aquesta capacitat. Això fa que el bus I²C sigui un bus multimestre.



El diagrama temporal en una transacció I²C té la forma següent:

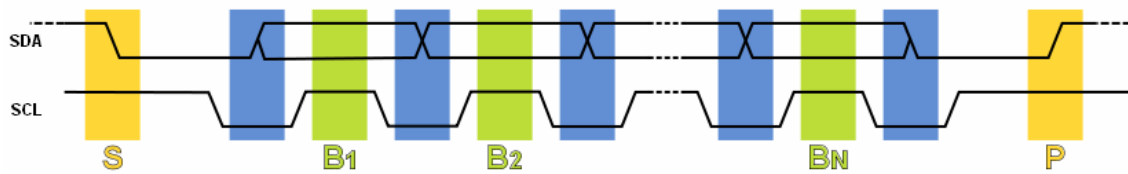


Figura 38: diagrama temporal d'una transacció I²C

- El bus està lliure (idle) quan SDA i SCL estan en estat lògic alt.
- Quan el bus està lliure qualsevol dispositiu el pot ocupar com a mestre.
- El mestre comença una comunicació enviant un patró anomenat “start condition” (S). D'aquesta manera alerta als dispositius esclaus i es posen a espera per a una transacció.
- El mestre es dirigeix al dispositiu amb qui vol parlar, enviant un byte que conté els set bits que componen l'adreça del dispositiu esclau amb qui es vol comunicar, i un octau bit de menor pes que correspon a l'operació desitjada (L/E), lectura = 1 (rebre de l'esclau) i escriptura = 0 (enviar a l'esclau).
- L'adreça enviada és comparada per cada un dels esclaus del bus amb la seva pròpia adreça. Si coincideixen l'esclau es considera adreçat com a esclau-transmissor o esclau-receptor depenent del bit L/E.
- L'esclau respon enviant un bit de ACK (“acknowledge”) que indica al dispositiu mestre que l'esclau ha reconegut la sol·licitud i està en condicions de comunicar-se.
- A continuació comença l'intercanvi d'informació entre els dispositius.
- El mestre envia l'adreça del registre intern del dispositiu que desitja llegir o escriure.
- L'esclau respon amb un altre bit de ACK.
- Ara el mestre pot començar a llegir o escriure bytes de dades. Tots els bytes de dades han de constar de 8 bits, el nombre màxim de bytes que poden ser enviats en una transacció no està restringit, essent l'esclau qui fixa aquesta quantitat en funció de les seves característiques.
- Cada byte llegit/escrit pel mestre ha de ser obligatòriament reconegut per un bit de ACK del dispositiu mestre/esclau.
- Es repeteixen els dos passos anteriors fins a finalitzar la comunicació entre el mestre i l'esclau.



- Encara que el mestre sempre és qui controla l'estat de la línia de rellotge, un esclau de baixa velocitat o que hagi d'aturar la transferència de dades per efectuar una altra funció, pot forçar la línia de rellotge a nivell baix. Això fa que el mestre entri en estat d'espera i deixi de transmetre informació, esperant a que l'esclau estigui preparat per continuar la transferència en el punt on ha estat parada.
- Quan la comunicació finalitza, el mestre transmet una "stop condition"(P) per a alliberar el bus.
- Després de la "stop condition" és obligatori que el bus estigui "idle" durant uns microsegons.

7.2 Microcontrolador

Com a μC per a la nostre electrònica escollim un model del fabricant Microchip® amb el que el departament està familiaritzat. Entre els dispositius de 16 bits d'aquest fabricant trobem dos grans grups: microcontroladors PIC24 (MCU) i controladors de senyal digitals dsPIC® (DSC). Totes les famílies de dispositius de 16 bits de Microchip® presenten compatibilitat de "pinout", programari, perifèrics i entorns de desenvolupament.

Els Microcontroladors PIC24 i els Controladors Digitals de Senyal dsPIC® estan governats per una màquina de 16 bits d'arquitectura "modified-Harvard" RISC. A més, els dsPIC® DSC combinen els avantatges de les altes prestacions d'un microcontrolador de 16 bits (MCU) amb l'alta velocitat de càlcul del processador de senyal digital (DSP) per a obtenir una xip d'alt nivell d'integració per a dissenys de sistemes encastats ("embedded systems").

Les característiques principals d'aquests dispositius són:

- Rang de tensió:
 - 2.5 a 5.5 V per a la família dsPIC30F
 - 3 a 3.6 V per a les famílies PIC24F, PIC24H i dsPIC33F
- Temperatura de treball:
 - Industrial (-40° fins +85°C)
 - Estès (-40° fins +125°C)
- Velocitat de treball:
 - 16 MIPS per a la família PIC24F
 - 30 MIPS per a la família dsPIC30F
 - 40 MIPS per a les famílies PIC24H i dsPIC33F



- Encapsulats de 18 pins PDIPs, 28 pins QFN i 100 pins TQFPs
- Sistema de manegament de l'alimentació flexible i d'altres prestacions
- Programació de la memòria Flash
- EEPROM de dades en la família dsPIC30F
- Memòria de dades SRAM
- Estructura d'interrupcions flexible
- Processat de les excepcions i interrupcions de programari ("software traps")

Altres prestacions d'aquests dispositius venen donades pels perifèrics que integren:

- Diferents possibilitats de rellotge: cristall, ressonador, RC de baix consum, Fast RC integrat al xip, amb o sense PLL
- Mòduls de comunicacions:
 - SPI™, I²C™, UART, CAN 2.0B (estàndard i millorat), "Codec Interface" (I²S i AC97)
- Comptadors
 - Captació d'entrades
 - Comparador de sortides i PWM estàndard
- Rellotge de temps real i calendari (RTCC)
- Comprovació de redundància cíclica (CRC)
- Control de motors per PWM
- Control d'alimentació per PWM
- Interfície de codificador de quadratura ("Quadrature Encoder")
- Convertidors analògic-digital (precissió de ± 1 LSB)
 - Fins a 2 A/Ds per xip
 - De 10 bits a 1 Msps
 - De 12 bits a 500 ksps
- Comparadors analògics
- Port paral·lel

De les quatre famílies disponibles ens quedem amb la dsPIC30F ja que integra el controlador de senyal digital (DSC) i entre els seus models trobem totes les característiques que necessitem (I²C, CAN, etc.).

El model dsPIC30F40XX és el més simple que ja integra CAN. La necessitat que tenim en el nostre sistema d'usar CAN (per a publicar les dades de les forces) i I²C (per a configurar els PGAs i llegir els A/Ds) de manera simultània,



fa que descartem el model dsPIC30F4012. Aquest dispositiu presenta multiplexats als mateixos pins els perifèrics de CAN i I²C (a més dels pins de programació), fet que impossibilitaria el seu ús concurrent.

D'aquesta manera acabem concretant la nostre elecció i ens decanem per usar el dsPIC30F4011. Cal remarcar que aquest dispositiu disposa d'un únic convertidor A/D de només 10 bits. Per tal de tenir una major resolució en la digitalització hem decidit usar un convertidor extern (ADS1100) de 16 bits. A més, aquest disseny no ens limita el nombre de digitalitzacions simultànies que podem realitzar, fent un disseny amb múltiples convertidors connectats a un mateix µC.

7.3 Disseny Funcional

Tal i com em vist abans, el fet de voler usar el bus I²C per a dues coses diferents, tenint en compte que per a cada una d'elles el master ha de ser un dispositiu distint, ens pot provocar que hi hagi col·lisions al bus. Aquest fenomen provocaria dues coses:

- El PGA309 no pogués fer una lectura satisfactòria de la seva configuració i per tant no habilités la seva sortida (V_{OUT}). El valor analògic que el A/D rep com a entrada no és vàlid.
- El µC no pogués fer-se master del bus I²C quan el PGA el manté ocupat i per tant no pot llegir el A/D. Si aconseguís fer-ho quan el PGA309 alliberés el bus després d'una configuració satisfactòria no el podria tenir ocupat molt de temps ja que es correria el perill de que el PGA el tornés a reclamar i el trobés ocupat. Si això passés el PGA no es podria reconfigurar i estaríem al punt anterior.

Per tal de solucionar aquest problema fem que el PGA309 condicioni el senyal del sensor amb la configuració guardada als registres interns i que només actualitzi els valors d'aquests quan al µC li “doni permís”. Això ho podem aconseguir fent servir el pin de test del PGA309. Connectant aquest pin a una sortida digital del µC podem fer que aquest darrer indiqui al PGA309 quan el bus I²C està lliure i així el PGA pugui actualitzar el valor dels seus registres sense perill de col·lisions. D'aquesta manera aconseguim que:

- el PGA309 no deixa mai de condicionar el senyal provinent del sensor de força (no deshabilita mai la seva sortida).
- no es produeixen mai col·lisions al bus I²C ja que el µC ha de donar permís al PGA per ser master.



Donat que volem captar 8 forces exercides a sobre del caminador necessitem vuit parelles PGA309/ADS1100. Si situem cada una d'aquestes parelles una placa de circuit imprès diferent ens veiem obligats a posar un total de vuit dsPICs (un per placa), a més de multiplicar també per vuit el nombre de xips destinats a les comunicacions del μC amb l'entorn (CAN, USB, RS-232, etc). Per aquesta raó, i per reduir el volum total de les electròniques, optem per col·locar a sobre d'un mateix circuit imprès els dispositius necessaris per a condicionar el senyal de quatre sensors de força.

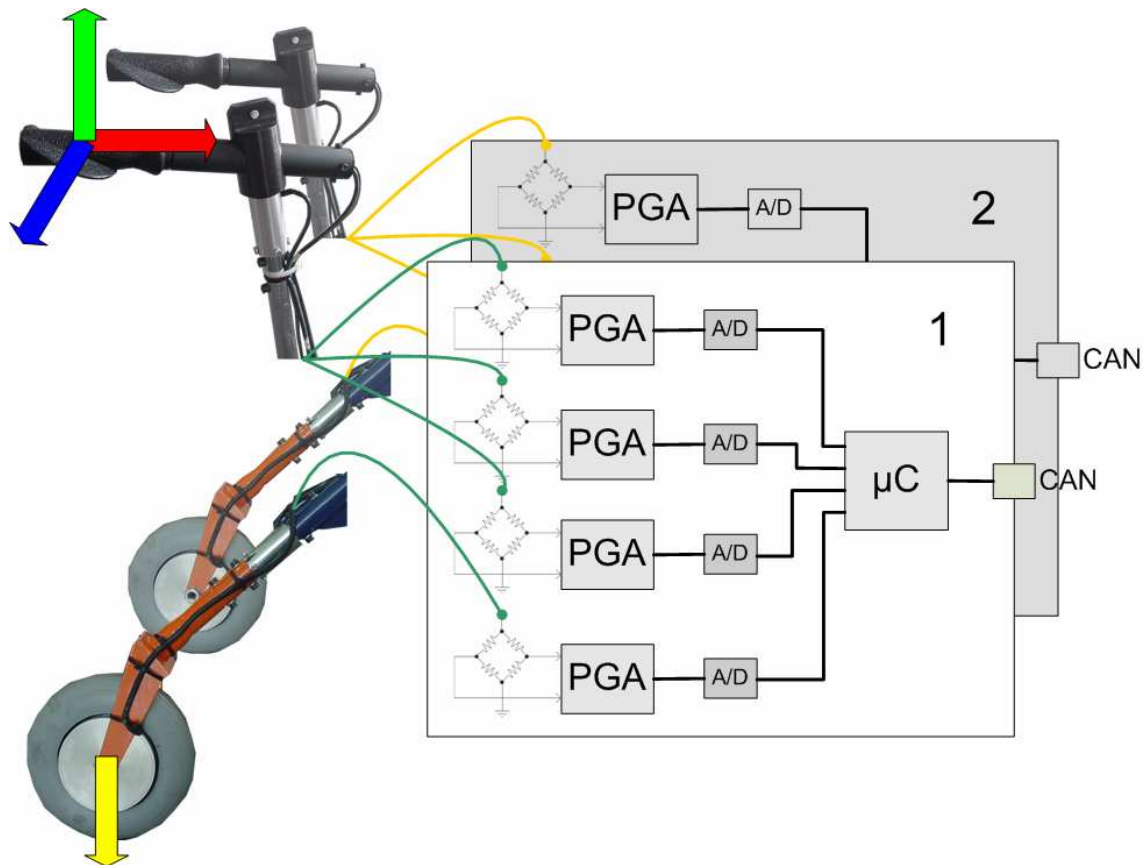


Figura 39: disseny funcional de les electròniques

D'aquesta manera un mateix dsPIC es comunica amb quatre PGAs i quatre A/Ds. Aquest fet no ens provoca cap tipus de problema de col·lisions al bus ja que el μC segueix indicant a les PGAs quan poden ser masters de I^2C , i per altra banda cada un dels quatre A/Ds té una adreça diferent.

Podem dir que el dsPIC juga diferents papers:

- Esclau de I^2C : quan respon a les peticions de configuració d'alguna PGA a la que prèviament ha donat permís per a ser mestre. En aquest cas el μC emula la EEPROM.
- Mestre de I^2C : quan llegeix als registres dels A/Ds per a obtenir els valors digitalitzats condicionats pels PGAs.



- “Mestre de mestres” de I²C: quan un cop ha alliberat el bus, permet a un (i només un) PGA ser mestre per a que aquest es reconfiguri. Aleshores el micro passa a ser esclau de I²C.

L'esquema resultant és el següent:

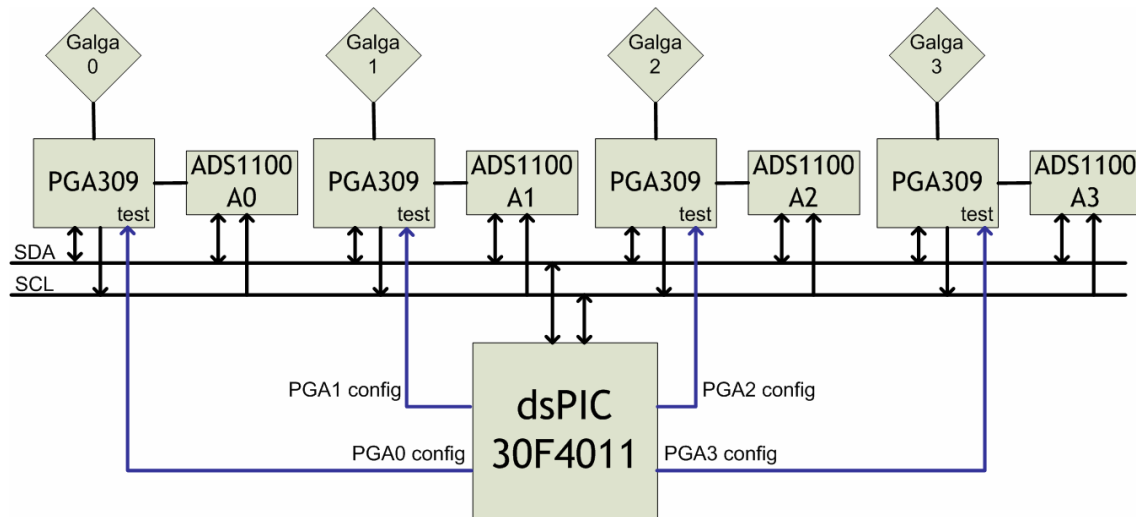


Figura 40: esquema de l'electrònica de captació

7.4 Disseny de la PCB

Per al disseny del circuit integrat fem servir Orcad. Aquest entorn de desenvolupament consta de diferents programes dels quals nosaltres en fem servir dos: Orcad Capture i Orcad Layout.

Orcad Capture ens permet definir tots els components electrònics que fem servir, i els hi posa un identificador a cada un d'ells. També permet descriure totes les connexions entre els pins dels components, definint així les diferents xarxes. El resultat final és l'esquema elèctric complet de la nostra electrònica.

A la Figura 41 es mostren les connexions d'un PGA309. El mateix esquema es repeteix per a cada un dels altres tres amplificadors, cada un connectat al pin del dsPIC corresponent (control del PGA usant el pin TEST), així com a la seva galga i al seu digitalitzador. El valor de les resistències indicades com a “R” depèn de les galgues que hi vulguem connectar i serveixen per a controlar la intensitat de corrent que aquesta consumirà.

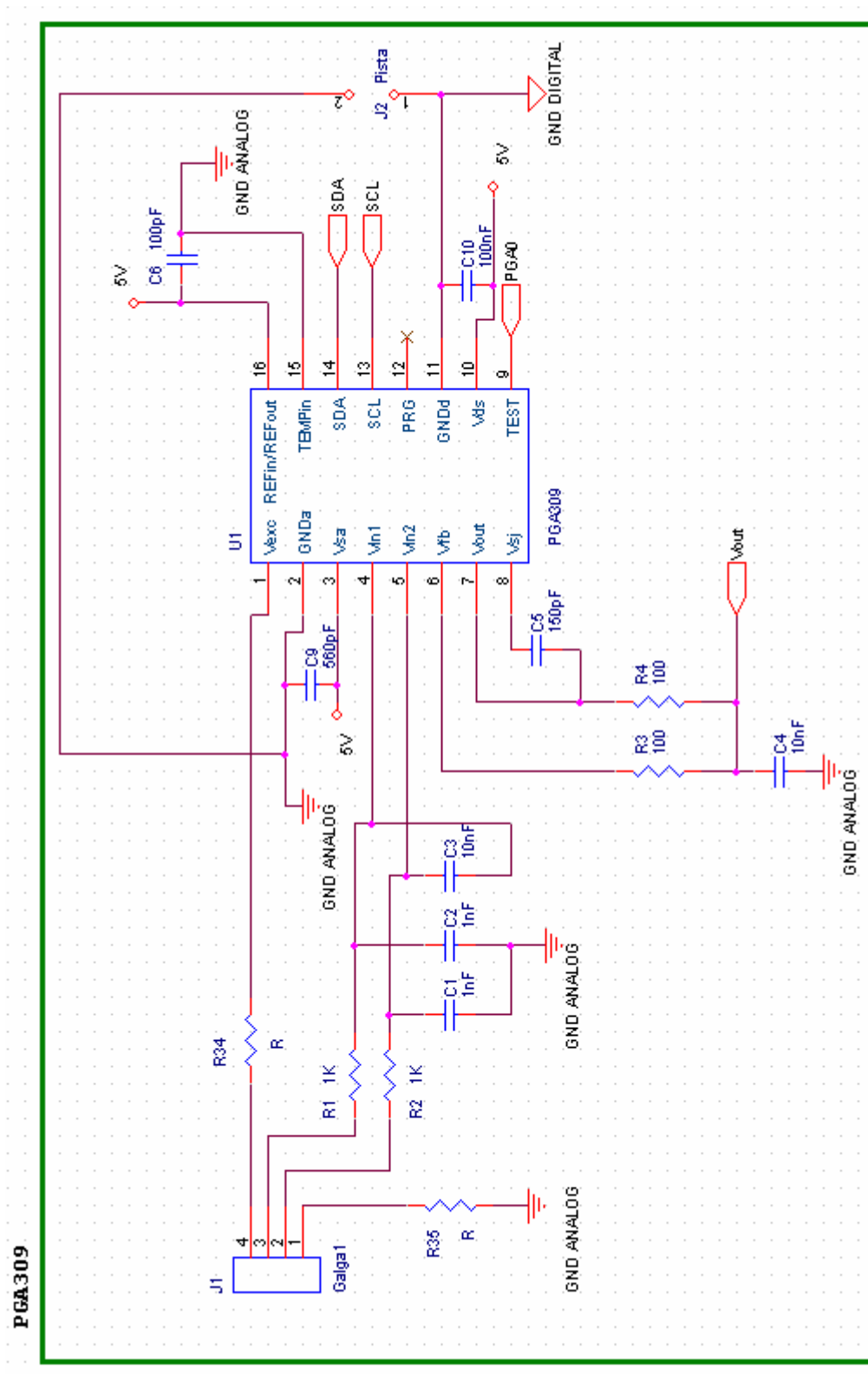


Figura 41: connexions d'un PGA309



La figura següent il·lustra les connexions necessàries per a cada un dels convertidors de senyal ADS1100:

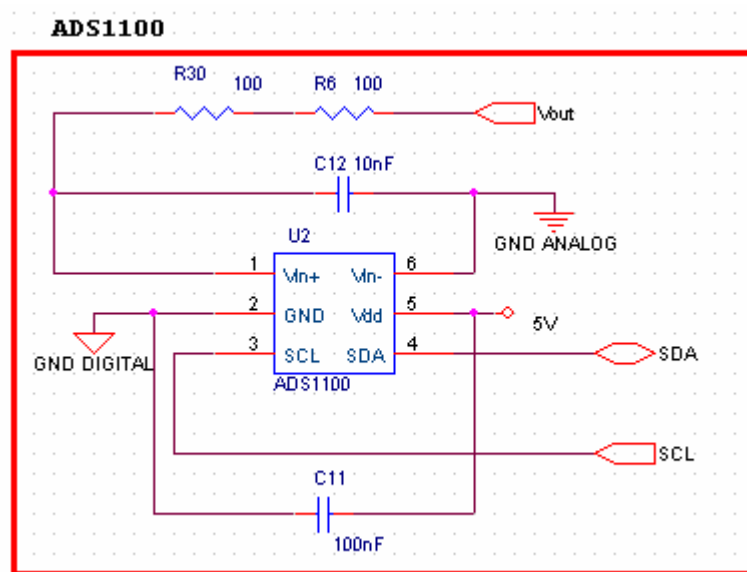


Figura 42: connexions d'un ADS1100

Com s'ha vist en apartats anteriors, el bus I²C necessita resistències de “pull-up” per funcionar. Per altra banda, sabem que el model de µC que hem escollit multiplexa els pins de I²C amb els pins de programació i que alguns programadors (com el ICD 2 de Microchip®) tenen problemes -ahora de programar microcontroladors- causats per la col·locació de resistències de “pull-up”. Per aquest motiu usem dos “jumpers” per tal de poder desconnectar les resistències alhora de programar. L'esquema és el següent:

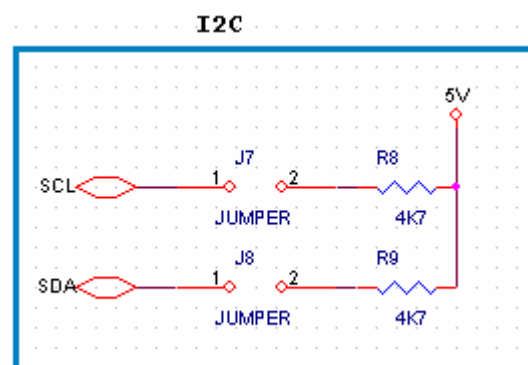


Figura 43: resistències de “pull-up” amb “jumpers”



A la figura següent es mostren les connexions del dsPIC30F4011 amb el cristall de 8 MHz, interruptor de reset, pins de test dels PGAs, bus I²C, controlador de CAN i controlador USB.

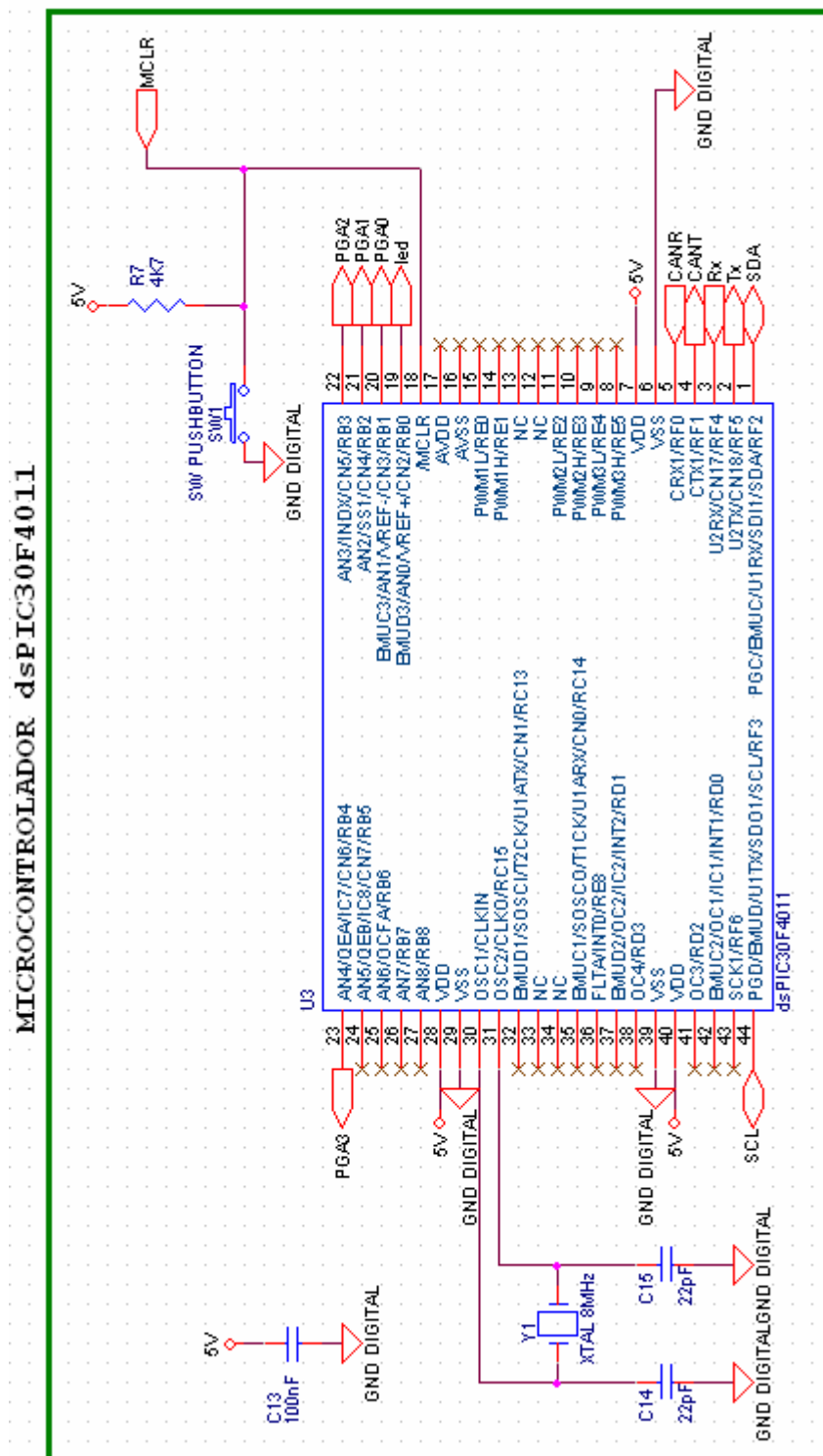


Figura 44: connexions del dsPIC30F4011

Per tal d'ajudar en el debugat del programa usem un led que connectem a un dels pins del dsPIC i que podrem encendre i apagar a voluntat. La figura següent mostra la connexió d'aquest led a la xarxa de 5V i al dsPIC. També mostra el connector de programació.

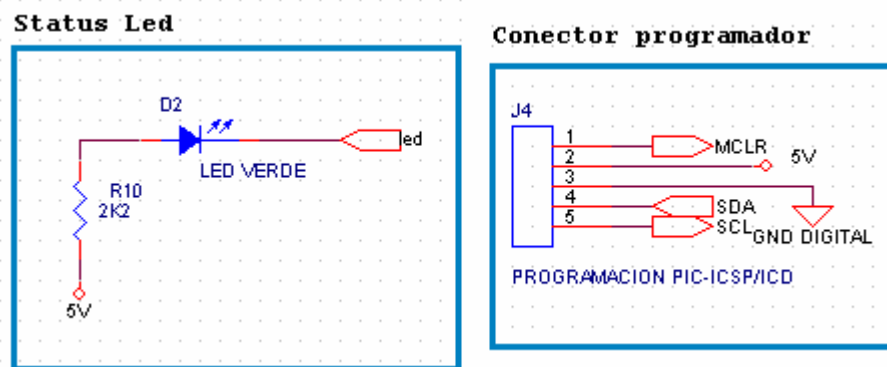


Figura 45: led i connector de programació

Per tal de comunicar el dsPIC amb el bus CAN fem servir el xip controlador 82C250.

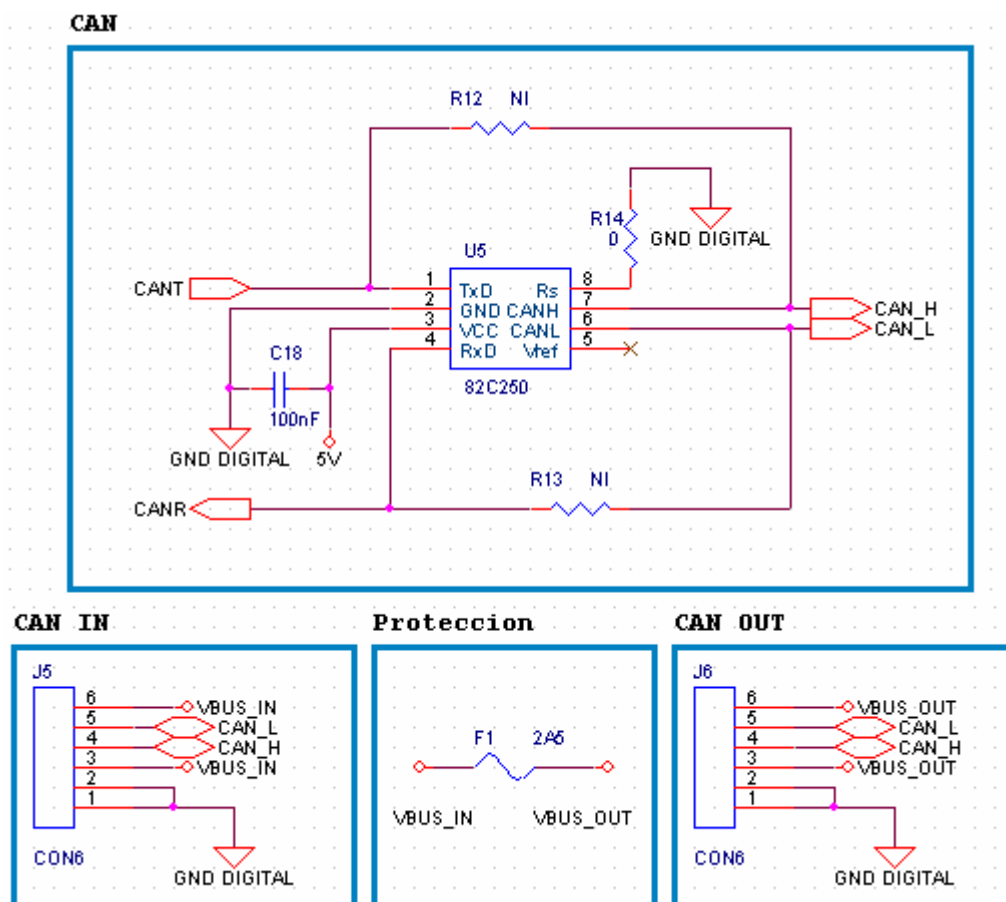


Figura 46: controlador i connectors de CAN



Un cop acabat tot el procés obtenim el resultat següent:

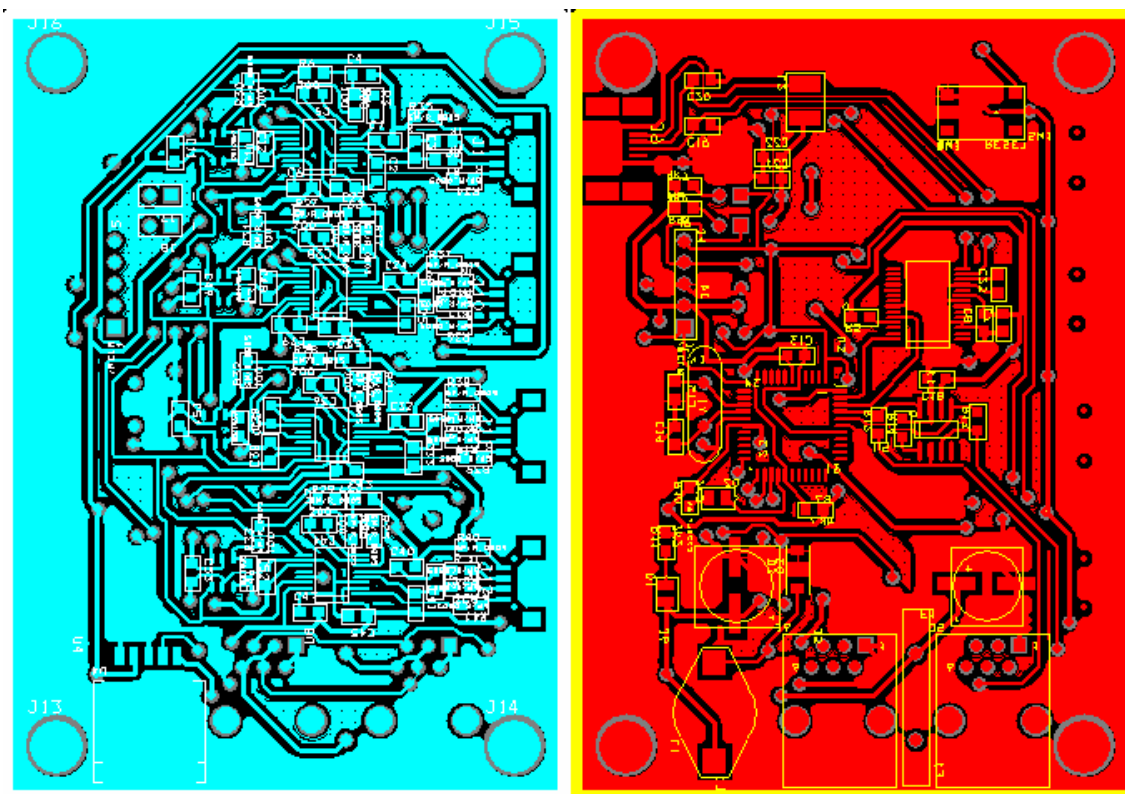


Figura 50: PCB dissenyada

La imatge en blau correspon a la cara de dalt (“top”) i la vermella a la cara de sota (“bottom”). Es podem observar en color blanc i groc respectivament la col·locació de tots els components electrònics per la superfície de les dues cares.

El programa Orcad Layout genera uns arxius que s’usen com a entrada de la màquina fresadora. Aquesta màquina el que fa es mecanitzar planxes de coure per tal de generar les pistes i pads que han estat definits.

Per tal de passar vies i unir pistes de les diferents cares de la placa, passem un filferro pel forat generat per la fresa i el soldem a banda i banda. La placa un cop fresada i amb les vies passades resulta com es mostra a la figura següent.

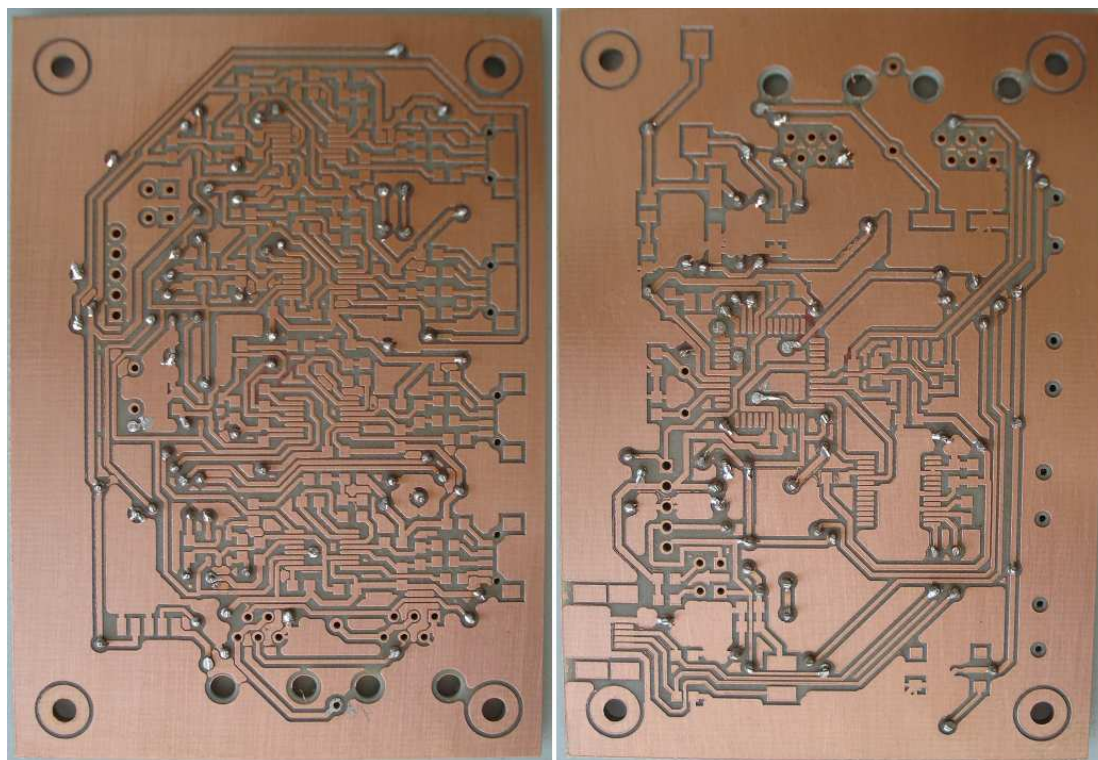


Figura 51: plaques fresades amb les vies passades

Un cop soldats tots els components a les dues cares de la PCB obtenim l'electrònica que hem dissenyat.

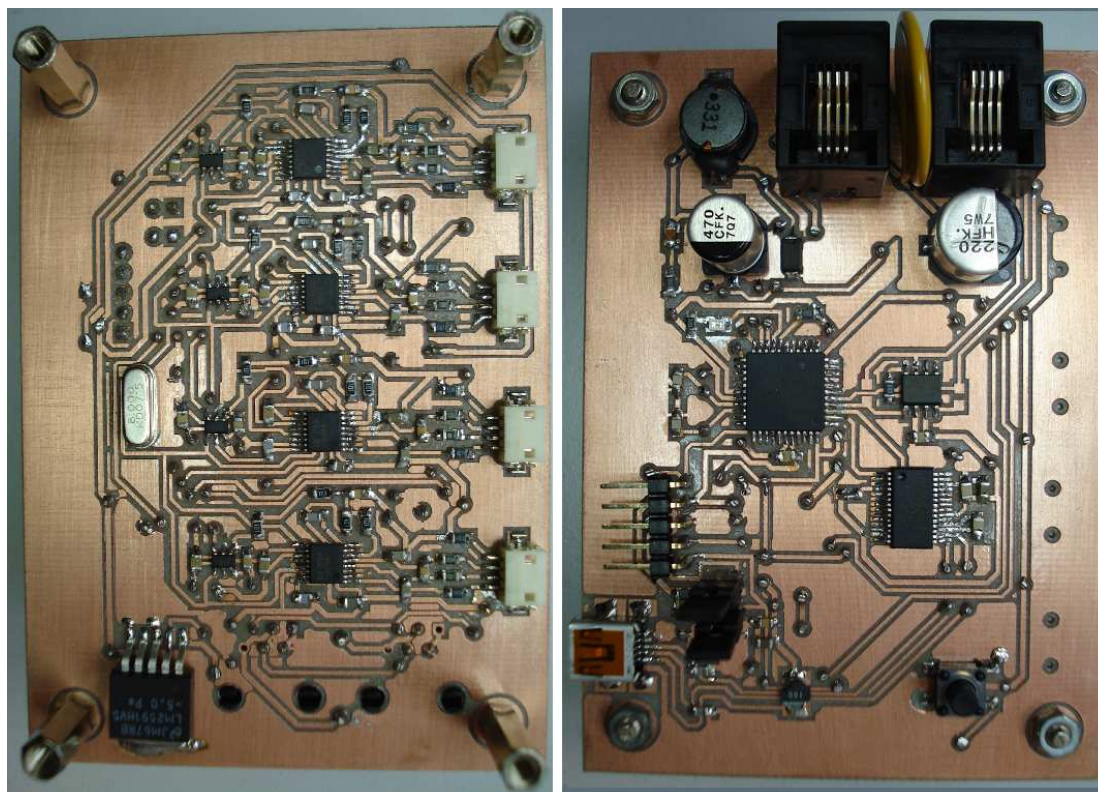


Figura 52: electrònica acabada



8 Firmware del Microcontrolador

A continuació es mostra en forma de diagrama de flux el codi que s'ha implementat per al dsPIC:

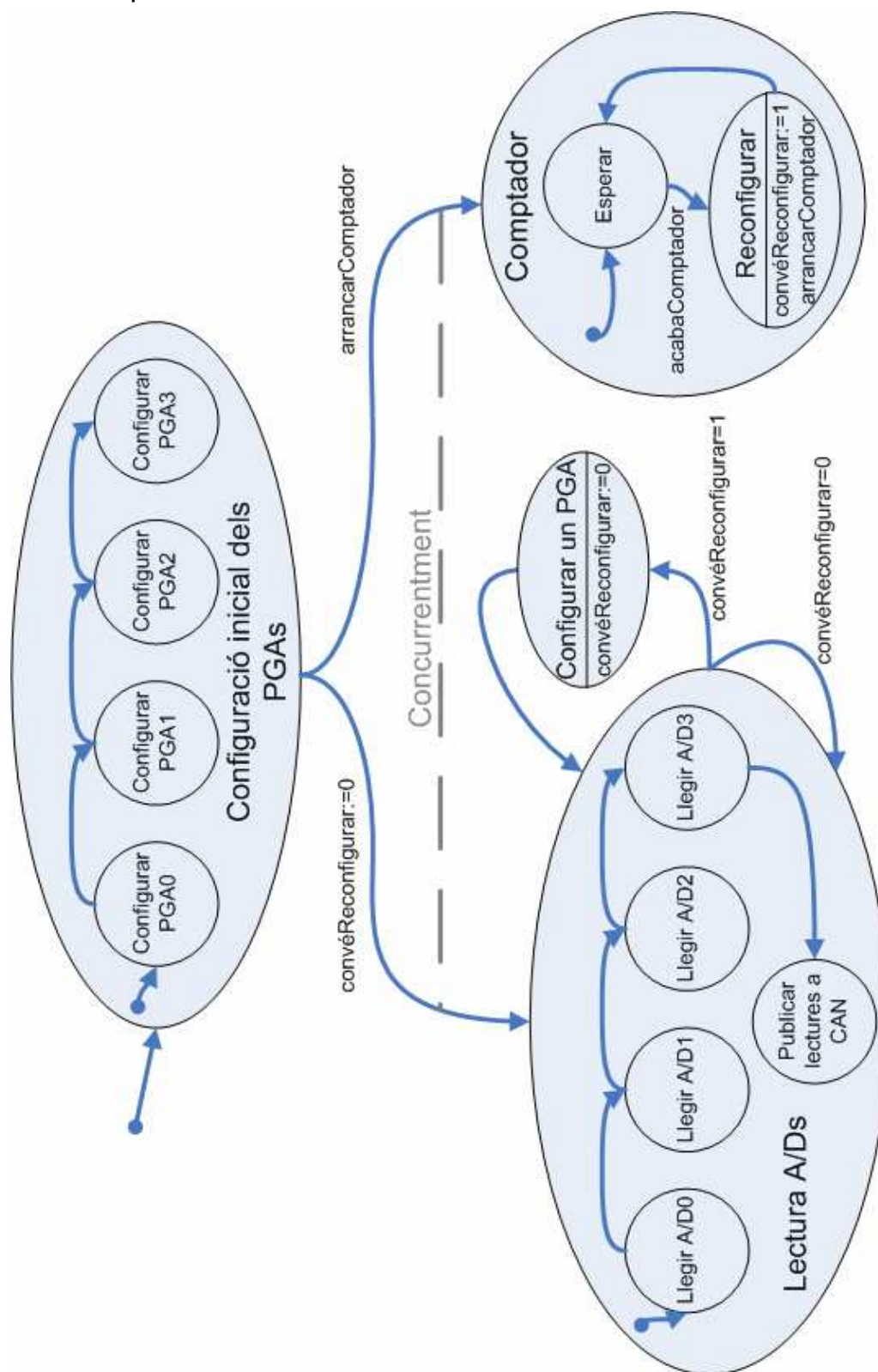


Figura 53: diagrama de flux del codi del μC



El dsPIC comença configurant els quatre PGAs, un després de l'altre. Per a fer-ho posa un zero lògic al pin connectat al test del PGA i després respon les peticions que li fa aquest. Un cop configurats els quatre PGAs s'arranca el comptador (Timer1) i es comencen a llegir de manera cíclica els quatre A/Ds. Les dades obtingudes es publiquen al bus CAN fent servir unes llibreries implementades per gent del departament.

A continuació es comprova el valor de la variable global *“conveConfigurar”*. Aquesta variable es posa a “1” quan el comptador acaba. Ens serveix per indicar que fa un cert període de temps que no es reconfigura cap PGA. Si això passa el dsPIC permet la configuració d'un dels amplificadors, el que fa més temps que no ho ha fet. Si es posa un comptador d'un minut (per exemple), cada PGA es reconfigurarà cada 4 minuts; i durant aquest temps seguirà amplificant i condicionant el senyal amb la configuració emmagatzemada als seus registres interns.

Per implementar el comptador de comptatge més llarg, que el que podem tenir fent servir els 16 bits del “Timer1”, fem servir una variable global. Cada cop que el “timer” arriba a zero s'incrementa el valor de la variable i quan aquesta assoleix un cert límit (per exemple el corresponent a un minut de comptatge) es posa a “1” la variable *“conveConfigurar”*.

Cal comentar que els codis de les dues electròniques no són exactament el mateix. La diferència recau en el identificador que posen a la trama CAN que publiquen: l'electrònica encarregada dels quatre sensors de la part esquerra del caminador fa servir el identificador 10, i l'encarregada de la part dreta fa servir el 20.

En quan al port sèrie, s'ha fet servir per tal de “debugar” el codi durant la seva implementació; però la versió definitiva d'aquest codi no escriu ni llegeix res mitjançant la UART. De totes maneres, recordar la utilitat del connector USB per a alimentar l'electrònica.



8.1 Codi comentat

A continuació es presenta el codi implementat en el llenguatge de programació C30.

```
#include <p30fxxxx.h>
#include <uart.h>
#include <i2c.h>
#include <timer.h>
#include "HL_CAN.h"

/*****
/* Configuration words
*****/
// Tenir en compte que:
// - el clock es de 8MHz
// - el prescaler el posem a 16x
// - el postscaler esta desactivat
// Això fa que FOSC sigui de 128MHz, i per tant, FCY=FOSC/4=32MHz

_FOSC(CSW_FSCM_OFF & XT_PLL16); //XT with 16xPLL oscillator, Failsafe
                                clock off

_FWDT(WDT_OFF); //Turn off the Watch-Dog Timer.
_FBORPOR(MCLR_EN & PWRT_OFF); //Enable MCLR reset pin and turn off
                                the power-up timers.
_FGS(CODE_PROT_OFF); //Disable Code Protection

/*****
/* Hardware
*****/
#define BAUD_RATE          115200
#define FXT                8000000//7372800
#define PLL                /*8*/16
#define FCY                (FXT * PLL) / 4
#define BRG                (FCY / (16L * BAUD_RATE)) - 1L
#define TIMER_PERIOD       0xFFFF

/*****
/* Global Variable declaration
*****/
int enviat;
unsigned int PGAConfigurada;
unsigned int conveConfigurar;
unsigned int proximaConfiguracio;
unsigned int contador;
```



```

/*****
/* Interrupts
*****/

// rutina d'interruptió del timer 1: cada 20 cops que salta indica que
convé configurar un PGA
void _ISR _T1Interrupt() {
    contador++;
    if(contador == 20){
        conveConfigurar = 1;
        contador = 0;
    }
    WriteTimer1(0x0000);           // Reset time counter
    IFS0bits.T1IF = 0;             // Clean interrupt flag
}

//rutina d'interruptió de I2C per a respondre a les peticions de
configuració dels PGAs
void _ISR _SI2CInterrupt(){
    switch ( I2CSTATbits.R_W ){ // mirem si és un read o un write
        case 0: //és una petició de read -> no fer res
            break;
        case 1: //és una petició de write
            switch ( I2CSTATbits.D_A ){ // mirem si és una trama
                                         de dades o d'adreces
                case 1: // és una trama de dades
                    break;
                case 0: // és una trama d'adreces
                    if (enviat ==0){
                        SlaveWriteI2C(0x49);
                        // Programmed Flag Value (LSBs)
                        enviat ++;
                    }
                    else if (enviat ==1){
                        SlaveWriteI2C(0x54);
                        // Programmed Flag Value (MSBs)
                        enviat ++;
                    }
                    else if (enviat ==2){
                        SlaveWriteI2C(0x00);
                        enviat ++;
                    }
                    else if (enviat ==3){
                        SlaveWriteI2C(0x00);
                        enviat ++;
                    }
                    else if (enviat ==4){
                        SlaveWriteI2C(0x00);
                        enviat ++;
                    }
                    else if (enviat ==5){
                        SlaveWriteI2C(0x00);
                        enviat ++;
                    }
                    else if (enviat ==6){
                        SlaveWriteI2C(0x00);
                        // Register 3 (LSBs)
                        enviat ++;
                    }
            }
        }
    }
}

```




```
else if (enviat ==7){
    SlaveWriteI2C(0x05);
    // Register 3 (MSBs)
    enviat ++;
}
else if (enviat ==8){
    SlaveWriteI2C(0x01);
    // Register 4 (LSBs)
    enviat ++;
}
else if (enviat ==9){
    SlaveWriteI2C(0x57);
    // Register 4 (MSBs)
    enviat ++;
}
else if (enviat ==10){
    SlaveWriteI2C(0x07);
    // Register 5 (LSBs)
    enviat ++;
}
else if (enviat ==11){
    SlaveWriteI2C(0x0A);
    // Register 5 (MSBs)
    enviat ++;
}
else if (enviat ==12){
    SlaveWriteI2C(0x03);
    // Register 6 (LSBs)
    enviat ++;
}
else if (enviat ==13){
    SlaveWriteI2C(0x16);
    // Register 6 (MSBs)
    enviat ++;
}
else if (enviat ==14){
    SlaveWriteI2C(0xAB);
    // Checksum1 (LSBs)
    enviat ++;
}
else if (enviat ==15){
    SlaveWriteI2C(0x2F);
    // Checksum1 (MSBs)
    enviat ++;
}
else if (enviat ==16){
    SlaveWriteI2C(0x00);
    enviat ++;
}

//----- Second part of EEPROM

else if (enviat ==17){
    SlaveWriteI2C(0x00);
    // T0 (LSBs)
    enviat ++;
}
```



```
else if (enviat ==18){
    SlaveWriteI2C(0x08);
    // T0 (MSBs)
    enviat ++;
}
else if (enviat ==19){
    SlaveWriteI2C(0x00);
    // Z0 (LSBs)
    enviat ++;
}
else if (enviat ==20){
    SlaveWriteI2C(0x00);
    // Z0 (MSBs)
    enviat ++;
}
else if (enviat ==21){
    SlaveWriteI2C(0xFF);
    // G0 (LSBs)
    enviat ++;
}
else if (enviat ==22){
    SlaveWriteI2C(0xFF);
    // G0 (MSBs)
    enviat ++;
}
else if (enviat ==23){
    SlaveWriteI2C(0xFF);
    // Tend (LSBs)
    enviat ++;
}
else if (enviat ==24){
    SlaveWriteI2C(0x7F);
    // Tend (MSBs)
    enviat ++;
}
else if (enviat ==25){
    SlaveWriteI2C(0x00);
    // ZMend (LSBs)
    enviat ++;
}
else if (enviat ==26){
    SlaveWriteI2C(0x00);
    // ZMend (MSBs)
    enviat ++;
}
else if (enviat ==27){
    SlaveWriteI2C(0x01);
    // Checksum2 (LSBs)
    enviat ++;
}
else if (enviat ==28){
    SlaveWriteI2C(0x78);
    // Checksum2 (MSBs)
    enviat ++;
}
```



```
        else {
            enviat = 0; // preparem la pròxima
                        // configuració
            LATBbits.LATB1 = 1; // prohibim la
                        // configuració de la PGA 1
            LATBbits.LATB2 = 1; // prohibim la
                        // configuració de la PGA 2
            LATBbits.LATB3 = 1; // prohibim la
                        // configuració de la PGA 3
            LATBbits.LATB4 = 1; // prohibim la
                        // configuració de la PGA 4
            PGAConfigurada = 1; // la PGA ja
                        // està configurada

        }
        break;
    }
    break;
}
IFS0bits.SI2CIF = 0;
return;
}

/*****
/* Procedures */
*****/

// funció per a configurar la UART2
void ConfigUART(){
    U1MODE = 0; // Clear UART config - to avoid
                // problems with bootloader

    OpenUART2(UART_EN & // Enable UART
              UART_IDLE_CON & // Continue on idle mode
              UART_ALTRX_ALTTX & // Use alternate RX/TX pins
              UART_DIS_WAKE & // Disable wake-on start
              UART_DIS_LOOPBACK & // Disable loopback mode
              UART_DIS_ABAUD & // Disable autobaud mode
              UART_NO_PAR_8BIT & // 8bits / No parity
              UART_1STOPBIT, // 1 Stop bit
              UART_TX_PIN_NORMAL & // Tx break bit normal
              UART_TX_ENABLE, // Enable Transimition
              BRG); // Baudrate
}

// funció per a configurar els port d'entrada sortida del micro
void ConfigIO(){
    LATBbits.LATB0 = 1; //Led inicialment apagat
    TRISBbits.TRISB0 = 0; //Configurem pin RD0 com a sortida

    LATBbits.LATB1 = 1; //Prohibim configuració PGA1
    TRISBbits.TRISB1 = 0; //Configurem pin RD1 com a sortida

    LATBbits.LATB2 = 1; //Prohibim configuració PGA2
    TRISBbits.TRISB2 = 0; //Configurem pin RD2 com a sortida

    LATBbits.LATB3 = 1; //Prohibim configuració PGA3
    TRISBbits.TRISB3 = 0; //Configurem pin RD3 com a sortida

    LATBbits.LATB4 = 1; //Prohibim configuració PGA4
    TRISBbits.TRISB4 = 0; //Configurem pin RD4 com a sortida
}
```



```
// funció per configurar el Timer 1 que marcarà les configuracions
dels PGAs
void ConfigTimer()
{
    OpenTimer1(T1_ON &                // Start 16-bit Timer
               T1_IDLE_CON &          // Continue operation on idle
               T1_GATE_OFF &          // Timer gate accumulation
                                   disabled
               T1_PS_1_256 &          // Prescale 1:256
               T1_SYNC_EXT_OFF &      // Do not synchronize external
                                   clock input (N/A)
               T1_SOURCE_INT,         // Internal osc (FCY)
               TIMER_PERIOD);         // Desired Period

    ConfigIntTimer1(T1_INT_PRIOR_1 & // Interrupt Priority 4 (7
                                   Highest)
                   T1_INT_ON);       // Interrupt enabled
}

//programa principal
int main(void)
{
    unsigned char sortida[8]; // variable per posar els valors
                              // llegits dels vuit ADS1100

    ConfigUART(); // configurem la UART2
    ConfigIO(); // configurem els ports d'entrada/sortida del micro
    HL_Can_Init (32000000, 1000, 1); // inicialitzem CAN

    // Configuració I2C *****
    I2CBRG = 0x110;                // Baud rate de 100 Khz
    I2CCONbits.AL0M = 0;           // adreça de 7 bits
    I2CADD=0x50;                   // adreça I2C slave
    EnableIntSI2C;                 // habiltem interrupció slave I2C
    I2CCONbits.I2CEN = 1;         // habiltem I2C

    // Configurem les PGAs *****
    enviat = 0;
    PGAConfigurada = 0;
    LATBbits.LATB1 = 0;           // permetre configuració PGA1
    while(!PGAConfigurada); // esperem a que la PGA es configuri
    Delay5ms(1);

    PGAConfigurada = 0;
    LATBbits.LATB2 = 0;           // permetre configuració PGA2
    while(!PGAConfigurada); // esperem a que la PGA es configuri
    Delay5ms(1);

    PGAConfigurada = 0;
    LATBbits.LATB3 = 0;           // permetre configuració PGA3
    while(!PGAConfigurada); // esperem a que la PGA es configuri
    Delay5ms(1);

    PGAConfigurada = 0;
    LATBbits.LATB4 = 0;           // permetre configuració PGA4
    while(!PGAConfigurada); // esperem a que la PGA es configuri
    Delay5ms(1);

    conveConfigurar = 0; // ara no cal configurar PGAs
    proximaConfiguracio = 1;
    contador = 0;
}
```



```
ConfigTimer(); // timer per a les pròximes configuracions de PGA

unsigned int iter;

while(1){          // Infinite loop

    //lectura del A/Ds *****
    iter = 0;
    while (iter < 4){
        StartI2C();
        // Wait till Start sequence is completed
        while(I2CCONbits.SEN);
        if (iter == 0){
            MasterWriteI2C(0x91); // adreça A/D1
        } else if (iter == 1){
            MasterWriteI2C(0x93); // adreça A/D2
        } else if (iter == 2){
            MasterWriteI2C(0x95); // adreça A/D3
        } else { // iter == 3
            MasterWriteI2C(0x97); // adreça A/D4
        }

        while(I2CSTATbits.TRSTAT);

        // llegim el primer byte
        I2CCONbits.RCEN = 1;
        while(!DataRdyI2C());
        sortida[iter*2] = I2CRCV;
        // generar ACK
        I2CCONbits.ACKDT = 0;
        I2CCONbits.ACKEN = 1;
        while(I2CCONbits.ACKEN == 1); // esperar a que acabi
                                         el ACK/NACK

        // llegim el segon byte
        I2CCONbits.RCEN = 1;
        while(!DataRdyI2C());
        sortida[iter*2+1] = I2CRCV;
        // generar NACK
        I2CCONbits.ACKDT = 1;
        I2CCONbits.ACKEN = 1;
        while(I2CCONbits.ACKEN == 1); // esperar a que acabi
                                         el ACK/NACK

        StopI2C();
        while(I2CCONbits.PEN);

        iter++;
    }

    //posem les lectures dels A/D's a CAN
    HL_Can_TmpTxTrama.Longitud = 8;
    HL_Can_TmpTxTrama.FrameID = 10; // id=10 indica que
les dades són dels sensors del costat esquerra del caminador
    //HL_Can_TmpTxTrama.FrameID = 20; // id=20 indica que
les dades són dels sensors del costat dret del caminador
    HL_Can_TmpTxTrama.Datos[0] = sortida[0];
    HL_Can_TmpTxTrama.Datos[1] = sortida[1];
    HL_Can_TmpTxTrama.Datos[2] = sortida[2];
    HL_Can_TmpTxTrama.Datos[3] = sortida[3];
    HL_Can_TmpTxTrama.Datos[4] = sortida[4];
    HL_Can_TmpTxTrama.Datos[5] = sortida[5];
    HL_Can_TmpTxTrama.Datos[6] = sortida[6];
}
```



```

HL_Can_TmpTxTrama.Datos[7] = sortida[7];
HL_Can_Tramas_Tx_Put();
Delay5ms(100);

// possible reconfiguració d'una PGA
if (conveConfigurar == 1){
    // encenem el led verd
    LATBbits.LATB0 = ~ LATBbits.LATB0;
    GAConfigurada = 0;
    if (proximaConfiguracio == 1){
        // permetre configuració PGA1
        LATBbits.LATB1 = 0;
    }
    else if (proximaConfiguracio == 2){
        // permetre configuració PGA2
        LATBbits.LATB2 = 0;
    }
    else if (proximaConfiguracio == 3){
        // permetre configuració PGA3
        LATBbits.LATB3 = 0;
    }
    else if (proximaConfiguracio == 4){
        // permetre configuració PGA4
        LATBbits.LATB4 = 0;
    }
    // esperem a que la PGA es configuri
    while(!PGAConfigurada);
    proximaConfiguracio++;
    if(proximaConfiguracio == 5){
        proximaConfiguracio = 1;
    }
    conveConfigurar = 0;
    Delay5ms(1);

    // un cop reconfigurat el PGA llegim un A/D per
    // recuperar el control del bus I2C
    StartI2C();
    // Wait till Start sequence is completed
    while(I2CCONbits.SEN);
    MasterWriteI2C(0x91);
    while(I2CSTATbits.TRSTAT);
    // llegim el primer byte
    I2CCONbits.RCEN = 1;
    while(!DataRdyI2C());
    sortida[iter*2] = I2CRCV;
    // generar ACK
    I2CCONbits.ACKDT = 0;
    I2CCONbits.ACKEN = 1;
    while(I2CCONbits.ACKEN == 1); // esperar a que
    // acabi el ACK/NACK

    // llegim el segon byte
    I2CCONbits.RCEN = 1;
    while(!DataRdyI2C());
    sortida[iter*2+1] = I2CRCV;
    // generar NACK
    I2CCONbits.ACKDT = 1;
    I2CCONbits.ACKEN = 1;
    while(I2CCONbits.ACKEN == 1); // esperar a que
    // acabi el ACK/NACK

```



```
        StopI2C();  
        while(I2CCONbits.PEN);  
  
        // apaguem el led verd  
        LATBbits.LATB0 =~ LATBbits.LATB0;  
    }  
    return 0;  
}
```





9 Comunicacions (CAN)

9.1 Controller Area Network

Com ja s'ha mencionat, per tal de comunicar la dada digitalitzada al computador fem servir un bus de comunicacions CAN. CAN és un protocol de comunicacions desenvolupat per la firma alemanya Robert Bosch GmbH (1985), basat en una topologia bus per a transmissions de missatges en ambients distribuïts. A més ofereix una solució a la gestió de la comunicació entre múltiples CPUs (unitats centrals de procés).

El protocol de comunicacions CAN proporciona els següents beneficis:

- És un protocol de comunicacions normalitzat, simplificant i economitzant la tasca de comunicar sistemes de diferents fabricants per una xarxa comú o bus.
- El processador amfitrió (“host”) delega la càrrega de comunicacions a un perifèric intel·ligent. Així disposa de més temps per a executar les seves pròpies tasques.
- És una xarxa multiplexada i per tant redueix considerablement el cablejat i elimina les connexions punt a punt.

9.2 Principals característiques de CAN

CAN es basa en el model productor/consumidor. Aquest concepte o paradigma de comunicació de dades descriu una relació entre un productor i un o més consumidors.

CAN és un protocol orientat a missatges, es a dir, la informació que s'intercanvia es descompon en missatges als quals se'ls hi assigna un identificador i s'encapsulen en trames per a la transmissió. Cada missatge té un identificador únic dins de la xarxa, que els nodes usen per a decidir acceptar o no el missatge en concret.

Entre les principals característiques de CAN trobem:

- Prioritat de missatges
- Garantia de temps de latència
- Flexibilitat en la configuració
- Recepció per multidifusió (“multicast”) amb sincronització de temps
- Sistema robust en quan a consistència de dades



- Sistema multimestre
- Detecció i senyalització d'errors
- Retransmissió automàtica de trames errònies
- Distinció entre errors temporals i fallades permanents del nodes de la xarxa, i desconexió autònoma dels nodes defectuosos

CAN fou desenvolupat inicialment per a aplicacions en els automòbils i, per tant, la plataforma del protocol és resultat de les necessitats existents en l'àrea de l'automoció. L'Organització Internacional per a l'Estandarització (ISO) defineix dos tipus de xarxes CAN: una xarxa d'alta velocitat (fins 1 Mbps), sota l'estàndard ISO 11898-2, destinada a controlar el motor i a interconnectar les unitats de control electrònic (ECU); i una xarxa de baixa velocitat tolerant a fallades (menor o igual a 125 kbps), sota l'estàndard ISO 11519-2/ISO 11898-3, dedicat a la comunicació dels dispositius electrònics interns d'un automòbil com són el control de portes, llums o seients.

9.3 Protocol de comunicacions CAN

CAN és un protocol de comunicacions sèrie que suporta control distribuït en temps real amb un alt nivell de seguretat i multiplexació.

L'establiment d'una xarxa CAN per a interconnectar els dispositius electrònics interns a un vehicle té la finalitat de substituir o eliminar el cablejat. Les ECUs, sensors, sistemes antilliscants, etc. es connecten mitjançant una xarxa CAN a velocitats de transferència de fins a 1 Mbps.

D'acord amb el model de referència OSI (Open Systems Interconnections), l'arquitectura de protocols CAN inclou tres capes: física, d'enllaç de dades i d'aplicació; a més d'una capa d'especial especial per a la gestió i control del node, anomenada capa supervisor.

- Capa física: defineix els aspectes del medi físic per a la transmissió de dades entre nodes d'una xarxa CAN. Els més importants són nivells de senyal, representació, sincronització i temps en els que els bits es transfereixen al bus. L'especificació del protocol CAN no defineix una capa física, encara que els estàndards ISO 11899 estableixen les característiques que han de complir les aplicacions per a la transferència d'alta i baixa velocitat.



- Capa d'enllaç de dades: defineix les tasques independents del mètode d'accés al medi. A més, degut a que el protocol CAN dona suport per al processat en temps real a tots els sistemes que l'integren, es requereix un sistema de transmissió a freqüències altes i retards mínims. En les xarxes multimestre la tècnica d'accés al medi és molt important ja que tot node actiu té els drets per a controlar la xarxa i acaparar els recursos. Per tant la capa d'enllaç de dades defineix el mètode d'accés al medi així com els tipus de trames per a l'enviament de missatges.

Quan un node necessita enviar informació per una xarxa CAN pot passar que altres nodes intentin transmetre simultàniament. CAN resol aquest fet assignant prioritats mitjançant l'identificador de cada missatge. L'assignació es realitza durant el disseny del sistema en forma de nombres binaris i no es pot modificar dinàmicament. L'identificador amb el menor nombre binari és el té major prioritat.

El mètode d'accés al medi utilitzat és el d'Accés Múltiple per Detecció de Portadora amb Detecció de Col·lisions i Arbitratge de Prioritat de Missatge (CSMA/CD+AMP, "Carrier Sense Multiple Access with Collision Detection and Arbitration Message Priority"). D'acord amb aquest mètode, els nodes de la xarxa que necessiten transmetre informació han d'esperar a que el bus estigui lliure (detecció de portadora). Quan es compleix aquesta condició els nodes transmeten un bit d'inici (accés múltiple). Cada node llegeix el bus bit a bit durant la transmissió de la trama i comparen el valor transmès amb el valor rebut. Si els valors són idèntics el node continua amb la transmissió, si es detecta una diferència en els valors es du terme el mecanisme d'arbitratge.

CAN estableix dos formats de trames de dades ("data frame") que difereixen en la longitud del camp de l'identificador. Les trames estàndards ("standard frame") amb un identificador de 11 bits definides en l'especificació CAN 2.0A, i les trames esteses ("extended frame") amb un identificador de 29 bits definides en l'especificació CAN 2.0B.

Per a la transmissió i control de missatges CAN es defineixen quatre tipus de trames: de dades, remota ("remote frame"), d'error ("error frame") i de sobrecàrrega ("overload frame"). Les trames



remotes també s'estableixen en els dos formats, estàndard i estès, i tant les trames de dades com les remotes es separen en trames precedents mitjançant espais entre trames ("interframe space").

En quan a la detecció i maneig dels errors, un controlador CAN té la capacitat de detectar i manejar els errors que ocorrin a la xarxa. Tot error detectat per un node es notifica immediatament a la resta de nodes.

- Trama de dades: pot contenir entre 0 i 8 bytes d'informació útil ("broadcast").
 - SOF(Inici de trama): 1 bit dominant, inici de la transmissió.
 - Identificador: 11 bits + RTR (1 bit dominant).
 - Control: IDE (1 bit, estàndard o extès) + r0 (1 bit, sempre recessiu) + DLC (4 bits, mida del camp de dades).
 - Dades: de 0 a 8 bytes.
 - CRC: Codi detecció errors (15 bits, s'inclou 1 recessiu per a indicar la fi del CRC).
 - ACK: El transmissor posa 2 bits recessius.
 - EOF: Fi de la trama, 7 bits recessius consecutius.

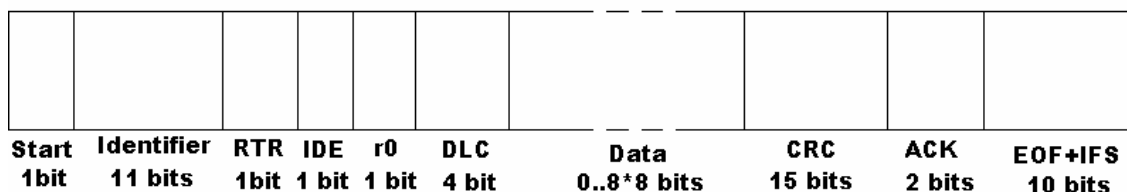


Figura 54: trama de dades CAN

- Remota: la d'interrogació remota, és fa servir per qualsevol node que vol sol·licitar la transmissió d'una trama de dades associada a un identificador, té format anàleg a la de dades però amb bit RTR recessiu, camp de dades buit, identificador el del missatge sol·licitat i el camp de longitud correspon a la longitud d'aquest missatge.
- La d'error s'usa per a indicar a la resta de nodes que s'ha detectat un error, invalidant la trama errònia. La genera



qualsevol node que detecta un error. La trama està formada per dos camps:

- Indicador d'error: és una superposició de flags d'error aportats per diferents nodes, i és diferent depenent de l'error:
 - passiu: 6 bits dominants que poden ser seguits dels generats per altres nodes fins a un màxim de 12 bits.
 - recessiu: 6 bits recessius consecutius a menys que siguin sobreescrits amb bits dominants per altres nodes.
- Delimitador d'error: 8 bits recessius consecutius.
- La recuperació d'un error és de 29 vegades el de temps de bit.

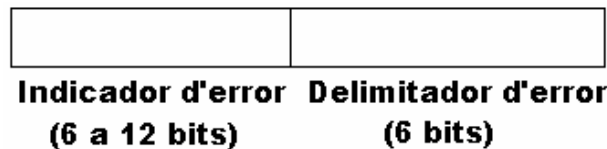


Figura 55: trama d'error CAN

- La de sobrecàrrega permet que un node forci els altre nodes a allargar el temps entre transmissions consecutives. Té el mateix format que la trama d'error, però es transmet durant l'espai "inter-trama" per a diferenciar-les.

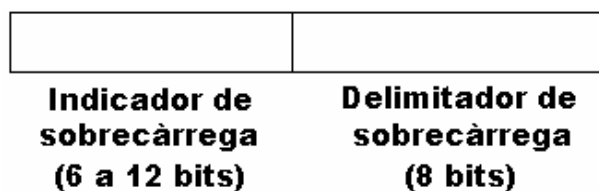


Figura 56: trama de sobrecàrrega CAN

- Capa de supervisor: la substitució del cablejat convencional per un sistema de bus sèrie presenta el problema de que un node defectuós pot bloquejar el funcionament del sistema complet. Cada node actiu transmet una bandera d'error quan detecta algun tipus d'error i pot ocasionar que un node defectuós acapari el medi físic. Per a eliminar aquest risc el protocol CAN defineix un mecanisme autònom per a detectar i desconnectar un node



defectuós del bus. Aquest mecanisme es coneix com aïllament de fallades.

- Capa d'aplicació: existeixen diferents estàndards que defineixen la capa d'aplicació. Alguns són específics i estan relacionats amb els seus camps d'aplicació. Entre les capes d'aplicació més usades s'han de mencionar CAL, CANopen, DeviceNet, SDS ("Smart Distributed System"), OSEK i CANKingdom.



10 Interfície gràfica

El darrer dels objectius d'aquest PFC és el de dissenyar i implementar una interfície gràfica senzilla i intuïtiva per tal d'il·lustrar el valor de les vuit forces captades i que qualsevol persona, sigui del àmbit professional que sigui, ho pugui entendre. No cal mencionar que aquesta interfície, si volem que tingui sentit, ha de funcionar en temps real.

Per a assolir aquest objectiu ens decantem per la realitat virtual (VR). Aquesta disciplina ens permet mostrar un mon virtual prou similar al “real” per a que qualsevol persona pugui interpretar el que hi apareix, fent un senzill símil.

A més, tenim l'avantatge que el professor Sergi de Haro ja disposa d'un model del caminador en VR. Usant aquest model es capaç de generar vídeos fent servir Matlab, on apareix un complet entorn pel qual el caminador es va desplaçant, tot indicant les forces que s'hi haurien d'aplicar per a aconseguir aquest comportament, així com la velocitat o les calories consumides.

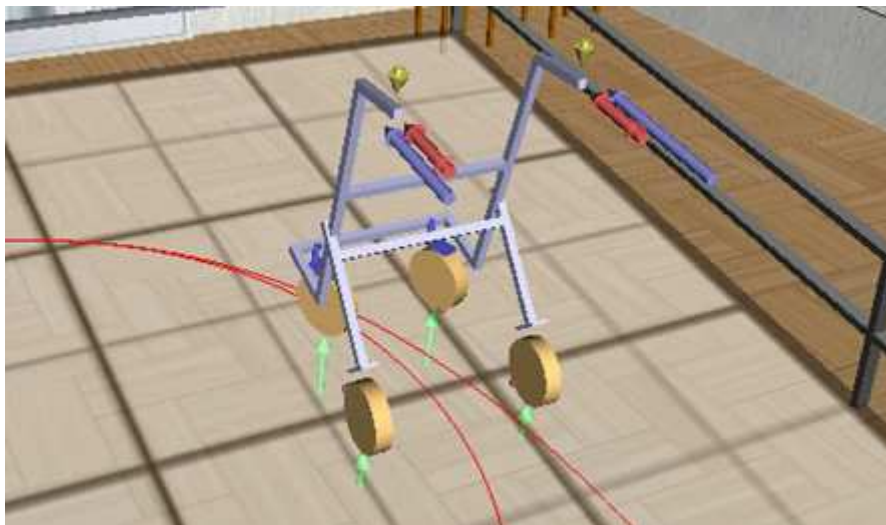


Figura 57: frame d'un vídeo generat amb l'entorn i codi de Sergi de Haro

Partint d'aquest model i codi Matlab, el meu treball ha consistit en fer que pogués ser executat en temps real (en comptes de generar un vídeo), ajustant el codi a l'entorn Simulink i captant com a entrades els valors de les forces publicats al bus CAN.

Altres modificacions fetes al model de partida han estat fetes per tal de modificar els colors i gruixos de les fletxes, així com la seva posició (situar les



tres fletxes de cada una de les manetes com si formessin un origen de coordenades) o direcció de creixement (ens interessava que la fletxa s'allargués per la punta i no per l'origen).

Per a la connexió del computador amb el bus CAN s'ha fet servir un dispositiu anomenat USBcan de la companyia Kvaser. Aquest dispositiu es connecta al bus i permet que el computador rebi les trames CAN mitjançant el port USB. Per tal de captar les trames des de CAN s'han fet servir unes llibreries ja existents implementades per Toni Benedico, del departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial.

Usant diferents blocs de Simulink es realitzen les tasques següents de manera seqüencial:

- S'estableix la connexió amb el bus CAN
- Es capturen les trames del bus CAN que tenen els identificadors 10 i 20 (corresponen a cada una de les electròniques)
- De cada trama s'extreuen 8 bytes que recombinem en forma de 4 doubles en complement a dos (format de sortida dels ADS1100)
- S'obté el valor del voltatge de sortida dels PGAs a partir de cada un dels doubles (procés invers al realitzat pels ADS1100)
- S'obté el valor de sortida de les galgues eliminant l'"offset" i el guany introduïts pels PGA309
- S'obté el valor de les forces aplicades a partir dels voltatges de sortida de les galgues, fent servir la informació del seu calibratge
- S'actualitzen els marcadors d'agulla amb els nous valors de força, així com la interfície de realitat virtual

Els blocs de Simulink ens resulten com es veu a la Figura 58. En ella s'observen els blocs següents:

- Cada indicador d'agulla presenta la força d'una de les 8 forces.
- El bloc "canInterface" (taronja) és l'encarregat d'obrir i tancar la connexió amb el bus CAN, cridant a un funcions escrites en C.
- El bloc "sampleTimeSync" (cian) sincronitza la simulació amb el rellotge intern del computador, assolint així el temps real.
- El bloc "VRModel" (verd) actualitza l'entorn de la realitat virtual, fent crida a una funció escrita en codi Matlab.

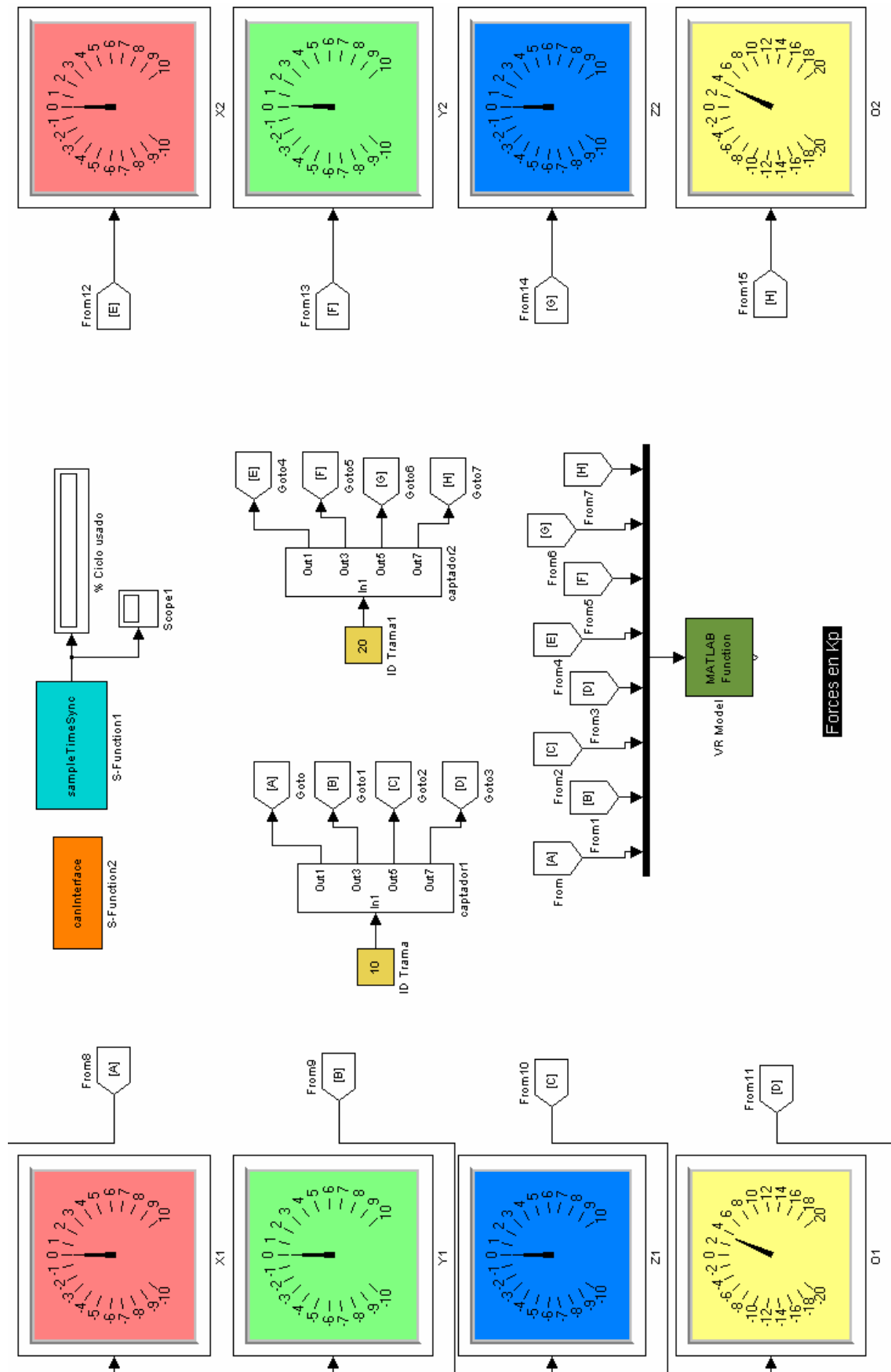


Figura 58: blocs Simulink per a la interfície



Els blocs “captador1” i “captador2” són subsistemes que contenen els blocs següents:

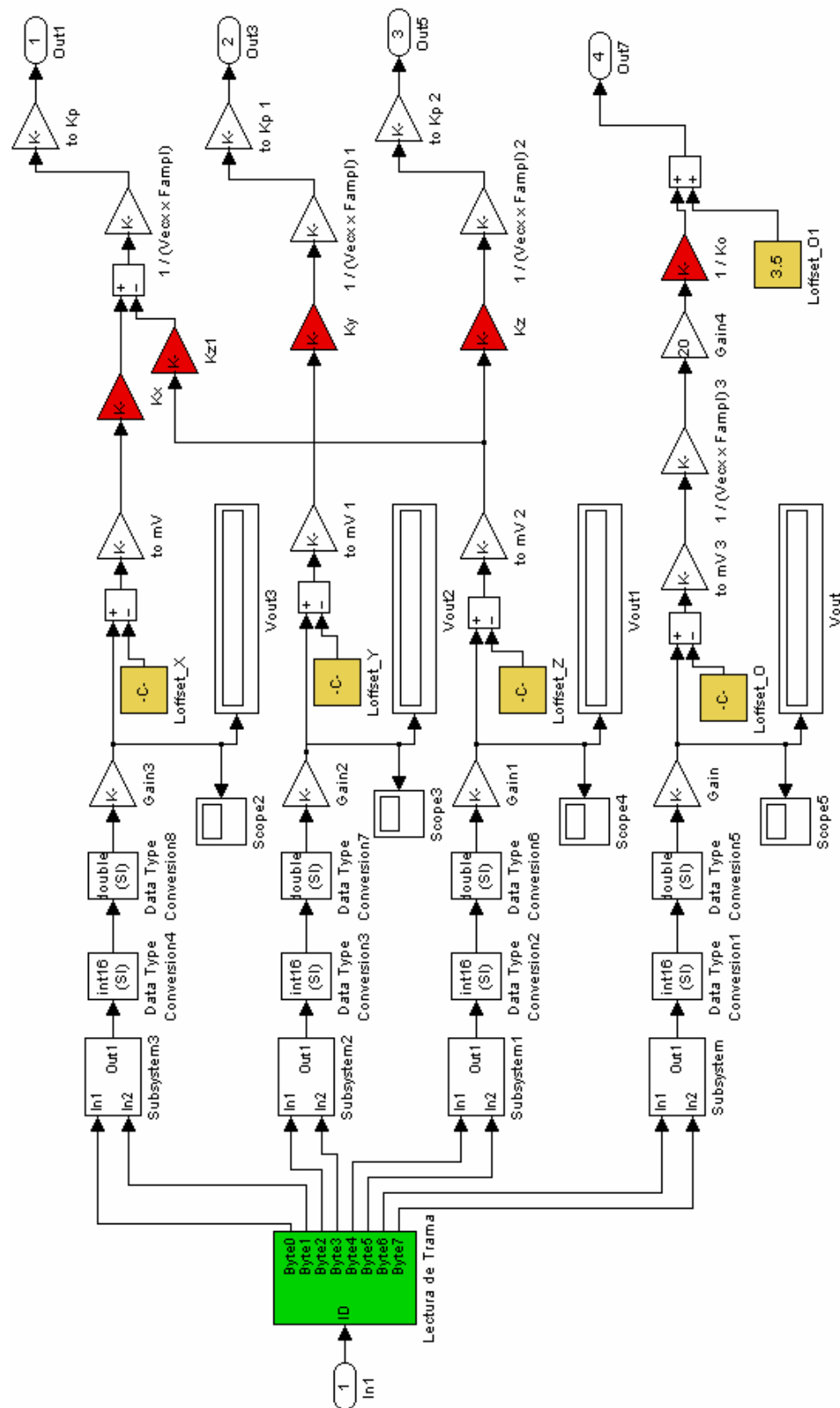


Figura 59: contingut dels subsistemes “captador1” i “captador2”



En aquests subsistemes és on es calculen els valors de les forces en kiloponds a partir dels bytes continguts a la trama CAN. Aquesta es llegida pel bloc “Lectura de Trama” (verd) que fa crida a una funció escrita en C. Els blocs “K” (vermells) contenen les constants calculades en el procés de calibratge dels sensors. Els blocs “Loffset” (daurats) contenen el valor de l’“offset” que s’ha de restar a cada senyal.

L’entorn de VR de Matlab es pot ajustar durant l’execució, permetent modificar el punt de vista de la càmera o activar el model de filferros.

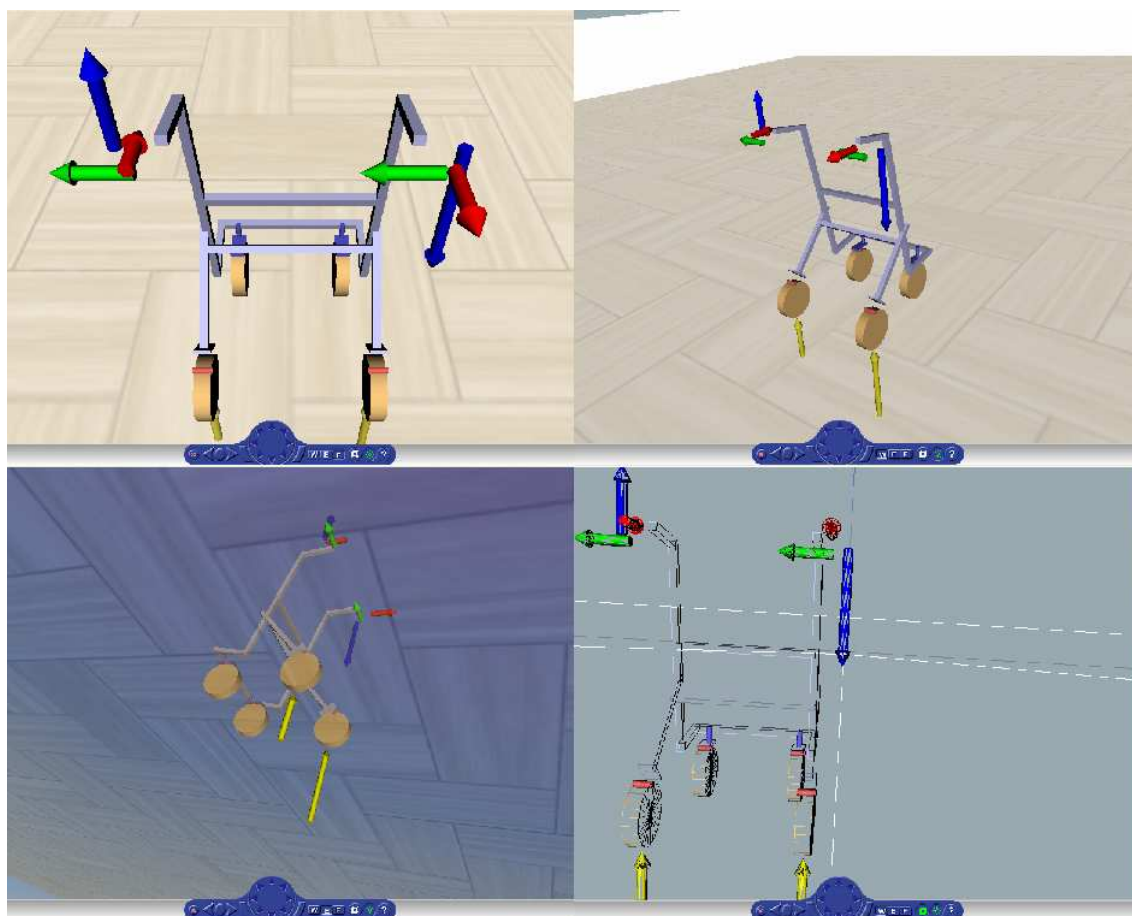


Figura 60: diferents vistes del caminador VR

Superposant la finestra amb l’entorn de realitat virtual a sobre dels blocs Simulink podem veure com varien les elongacions de les fletxes al mateix temps que veiem com es modifiquen els marcadors d’agulla indicant els valors de les forces.

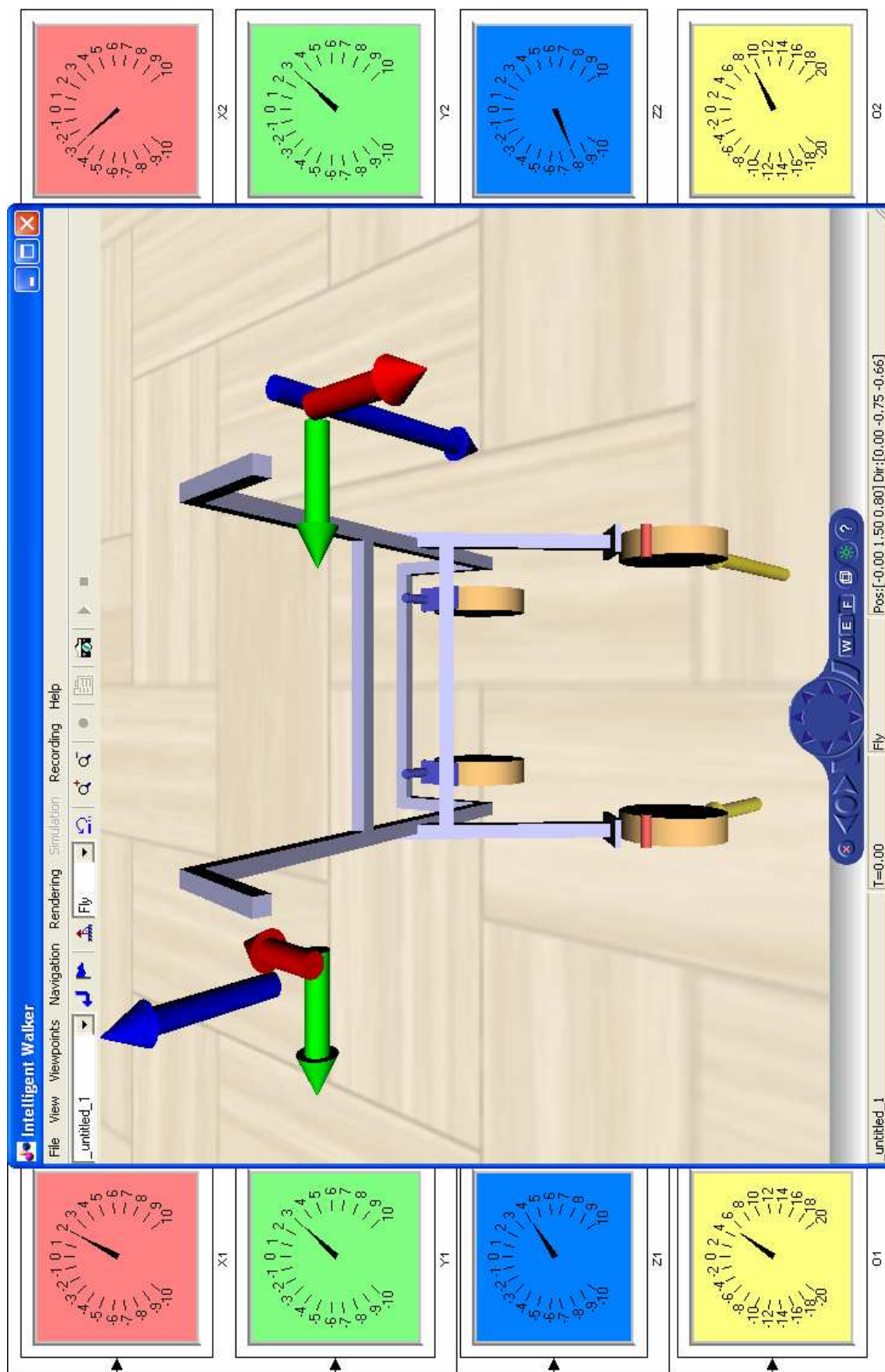


Figura 61: entorn gràfic final



Abans d'executar el Simulink, cal carregar l'entorn de realitat virtual i fer una seria d'inicialitzacions. Això s'aconsegueix amb el codi Matlab següent, adaptat de l'implementat per Sergi de Haro:

```
clear; clc;

% temps de mostreig
global Ts;
Ts=0.1;

% paràmetres i tria d'opcions

global grafF1; global grafF1U; global grafF2; global grafF3;
global grafNrear; global grafNfront;
global grafPunts; global grafO; global grafU; global usuari;
global petjades; global peus;
global comentar; global numFL; global numFR; global numV; global numW;
global logos;
global trajO; global trajT; global fletxes; global esferes;
global pista; global decoracio;
global grabar; global girarVista; global fps;

grafF1='s'; grafF1U='s'; grafF2='s'; grafF3='s'; grafNrear='s';
grafNfront='n';
grafPunts='s'; grafO='s'; grafU='s'; usuari='n'; petjades='n';
peus='n';
comentar='n'; numFL='n'; numFR='n'; numV='s'; numW='n'; logos='n';
trajO='n'; trajT='n'; fletxes='s'; esferes='s'; pista='n';
decoracio='i';
grabar='n'; girarVista='n'; fps=15;

% lectura del paràmetres del model
llistaParametres=load('parameters.tsv');
global dt;
dt=llistaParametres(1);
global radiRoda;
radiRoda=llistaParametres(14);
global Zthi;
Zthi=llistaParametres(23);
global Zths;
Zths=llistaParametres(24);
global Nthi;
Nthi=llistaParametres(31);
global Nths;
Nths=llistaParametres(32);
global nul;
nul=llistaParametres(37);
global nur;
nur=llistaParametres(38);
global Tmax;
Tmax=llistaParametres(39);
global Trod;
Trod=llistaParametres(40);

% lectura del món virtual VW
vrsetpref('DefaultFigureAntialiasing','on');
vrsetpref('DefaultFigurePosition',[100 100 800 600]);
global world;
```



```
world=vrworld('walker6.wrl'); open(world); view(world);

% assignació a variables dels nodes del VW
global FlS;
FlS=vrnode(world, 'Fl_switch');
global FlUS;
FlUS=vrnode(world, 'FlU_switch');
global F2S;
F2S=vrnode(world, 'F2_switch');
global F3S;
F3S=vrnode(world, 'F3_switch');
global NrearS;
NrearS=vrnode(world, 'Nrear_switch');
global NiS;
NiS=vrnode(world, 'Ni_switch');
global NdS;
NdS=vrnode(world, 'Nd_switch');
global puntsS;
puntsS=vrnode(world, 'punts_switch');
global puntOS;
puntOS=vrnode(world, 'O_switch');
global userS;
userS=vrnode(world, 'user_switch');
global footprintS;
footprintS=vrnode(world, 'footprint_switch');
global peusS;
peusS=vrnode(world, 'shoes_switch');
global puntUS;
puntUS=vrnode(world, 'U_switch');

global puntU;
puntU=vrnode(world, 'punt_U');
global footprintL;
footprintL=vrnode(world, 'petjada_l');
global footprintR;
footprintR=vrnode(world, 'petjada_r');
global puntOt;
puntOt=vrnode(world, 'puntOt');
global puntOr;
puntOr=vrnode(world, 'puntOr');
global rodaL;
rodaL=vrnode(world, 'roda_l');
global rodaR;
rodaR=vrnode(world, 'roda_r');
global rodaLC;
rodaLC=vrnode(world, 'l_color');
global rodaRC;
rodaRC=vrnode(world, 'r_color');
global castorI;
castorI=vrnode(world, 'castor_i');
global castorD;
castorD=vrnode(world, 'castor_d');
global rodaI;
rodaI=vrnode(world, 'roda_i');
global rodaD;
rodaD=vrnode(world, 'roda_d');

global FlI;
FlI=vrnode(world, 'FlI_vector');
global Flr;
```



```
Flr=vrnode(world, 'Flr_vector');
global Fl1S;
Fl1S=vrnode(world, 'Fl1_switch');
global FlrS;
FlrS=vrnode(world, 'Flr_switch');
global Fl1T;
Fl1T=vrnode(world, 'Fl1_target');
global FlrT;
FlrT=vrnode(world, 'Flr_target');
global Fl1U;
Fl1U=vrnode(world, 'Fl1U_vector');
global FlrU;
FlrU=vrnode(world, 'FlrU_vector');
global Fl1SU;
Fl1SU=vrnode(world, 'Fl1U_switch');
global FlrSU;
FlrSU=vrnode(world, 'FlrU_switch');

global Fl1SUP;
Fl1SUP=vrnode(world, 'Fl1U_pos');
global Fl1SUn;
Fl1SUn=vrnode(world, 'Fl1U_neg');
global FlrSUP;
FlrSUP=vrnode(world, 'FlrU_pos');
global FlrSUn;
FlrSUn=vrnode(world, 'FlrU_neg');

global F2lSp;
F2lSp=vrnode(world, 'F2l_pos');
global F2lSn;
F2lSn=vrnode(world, 'F2l_neg');
global F2rSp;
F2rSp=vrnode(world, 'F2r_pos');
global F2rSn;
F2rSn=vrnode(world, 'F2r_neg');

global F3lSp;
F3lSp=vrnode(world, 'F3l_pos');
global F3lSn;
F3lSn=vrnode(world, 'F3l_neg');
global F3rSp;
F3rSp=vrnode(world, 'F3r_pos');
global F3rSn;
F3rSn=vrnode(world, 'F3r_neg');

global F2l;
F2l=vrnode(world, 'F2l_vector');
global F2r;
F2r=vrnode(world, 'F2r_vector');
global F2lS;
F2lS=vrnode(world, 'F2l_switch');
global F2rS;
F2rS=vrnode(world, 'F2r_switch');
global F3l;
F3l=vrnode(world, 'F3l_vector');
global F3r;
F3r=vrnode(world, 'F3r_vector');
global F3lS;
F3lS=vrnode(world, 'F3l_switch');
```



```

global F3rS;
F3rS=vrnode(world, 'F3r_switch');
global Nl;
Nl=vrnode(world, 'Nl_vector');
global Nr;
Nr=vrnode(world, 'Nr_vector');
global Ni;
Ni=vrnode(world, 'Ni_vector');
global Nd;
Nd=vrnode(world, 'Nd_vector');
global NlS;
NlS=vrnode(world, 'Nl_switch');
global NrS;
NrS=vrnode(world, 'Nr_switch');
global NlvC;
NlvC=vrnode(world, 'Nl_vcolor');
global NrvC;
NrvC=vrnode(world, 'Nr_vcolor');
global NivC;
NivC=vrnode(world, 'Ni_vcolor');
global NdvC;
NdvC=vrnode(world, 'Nd_vcolor');
global NlcC;
NlcC=vrnode(world, 'Nl_ccolor');
global NrcC;
NrcC=vrnode(world, 'Nr_ccolor');
global NicC;
NicC=vrnode(world, 'Ni_ccolor');
global NdcC;
NdcC=vrnode(world, 'Nd_ccolor');

vista=vrnode(world, 'vista_front');
vrdrawnow; pause(2);

global tAnt; global iAnt; global zitaRAnt; global dZitaRAnt; global
zitaLAnt; global dZitaLAnt; global zitaIAnt;
global dZitaIAnt; global zitaDAnt; global dZitaDAnt;
global xV0; global yV0; global zV0; global radiV; global alfaV0;
global dalfaV;
global incr;

% condicions inicials
tAnt=0; iAnt=0; zitaRAnt=0; dZitaRAnt=0; zitaLAnt=0; dZitaLAnt=0;
zitaIAnt=0; dZitaIAnt=0; zitaDAnt=0; dZitaDAnt=0;
xV0=1; yV0=0.3; zV0=-2.5; radiV=sqrt(xV0^2+zV0^2);
alfaV0=atan(abs(zV0)/abs(xV0)); dalfaV=2*pi/40;
incr=round(1/(dt*fps));

global t;
t=0;

```

Totes les variables han estat definides de forma global per tal que puguin ser consultades i/o modificades des de la funció que s'encarrega d'actualitzar el mon virtual, invocada pel model Simulink. El seu codi és el que es mostra a continuació:



```
function[] = vrSim(entrada)

% carreguem totes les variables globals
global grafF1; global grafF1U; global grafF2; global grafF3;
global grafNrear; global grafNfront;
global grafPunts; global grafO; global grafU; global usuari;
global petjades; global peus;
global comentar; global numFL; global numFR; global numV;
global numW; global logos;
global trajO; global trajT; global fletxes; global esferes;
global pista; global decoracio;
global grabar; global girarVista; global fps;

global F1S; global F1US; global F2S; global F3S; global NrearS;
global NiS; global NdS; global puntsS;
global puntOS; global userS; global footprintS; global peusS;
global puntUS; global puntU; global footprintL;
global footprintR; global puntOt; global puntOr; global rodaL;
global rodaR; global rodaLC; global rodaRC;
global castorI; global castorD; global rodaI; global rodaD;
global F1l; global F1r; global F1lS; global F1rS;
global F1lT; global F1rT; global F1lU; global F1rU; global F1lSU;
global F1rSU; global F2l; global F2r;
global F2lS; global F2rS; global F3l; global F3r; global F3lS;
global F3rS; global Nl; global Nr; global Ni;
global Nd; global NlS; global NrS; global NlvC; global NrvC;
global NivC; global NdvC; global NlcC; global NrcC;
global NicC; global NdcC;

global F1lSU; global F1lSUn; global F1rSU; global F1rSUn;
global F2lSp; F2lSn; global F2rSp; global F2rSn;
global F3lSp; global F3lSn; global F3rSp; global F3rSn;

% forces longitudinals de l'usuari (direccio 1)
unit=25;
mod1LU=entrada(1);% esquerra
mod1RU=entrada(5);%fuerza DERECHA
F1lU.scale = [1 abs(mod1LU/(unit*0.2)) 1];
F1lU.translation = [0 0.5*mod1LU/unit-0.025 0];
F1rU.scale = [1 abs(mod1RU/(unit*0.2)) 1];
F1rU.translation = [0 0.5*mod1RU/unit-0.025 0];
if mod1LU<0
    F1lSU.whichChoice=1;
F1lSUn.translation = [0 mod1LU/unit-0.05 0];
elseif mod1LU==0
    F1lSU.whichChoice=-1;
else
    F1lSU.whichChoice=0;
F1lSU.translation = [0 mod1LU/unit 0];
end
if mod1RU<0
    F1rSU.whichChoice=1;
F1rSUn.translation = [0 mod1RU/unit-0.05 0];
elseif mod1RU==0
    F1rSU.whichChoice=-1;
else
    F1rSU.whichChoice=0;
F1rSU.translation = [0 mod1RU/unit 0];
end
if grafF1U=='s'
```



```

        F1US.whichChoice=0;
    else
        F1US.whichChoice=-1;
    end

    % forces transversals de l'usuari (direcció 2)
    unit=25;
    mod2L=entrada(2); % esquerra
    mod2R=entrada(6); % dreta
    F2l.scale = [1 abs(mod2L/(unit*0.1)) 1];
    F2l.translation = [0 0.5*mod2L/unit-0.025 0];
    F2r.scale = [1 abs(mod2R/(unit*0.1)) 1];
    F2r.translation = [0 0.5*mod2R/unit-0.025 0];
    if mod2L<0
        F2lS.whichChoice=1;
        F2lSp.translation = [0 mod2L/unit-0.05 0];
    elseif mod2L==0
        F2lS.whichChoice=-1;
    else
        F2lS.whichChoice=0;
        F2lSn.translation = [0 mod2L/unit 0];
    end
    if mod2R<0
        F2rS.whichChoice=1;
        F2rSp.translation = [0 mod2R/unit-0.05 0];
    elseif mod2R==0
        F2rS.whichChoice=-1;
    else
        F2rS.whichChoice=0;
        F2rSn.translation = [0 mod2R/unit 0];
    end
    if grafF2=='s'
        F2S.whichChoice=0;
    else
        F2S.whichChoice=-1;
    end

    % forces verticals de l'usuari (direcció 3)
    unit=25;
    mod3L=-entrada(3); % esquerra
    mod3R=-entrada(7); % dreta
    F3l.scale = [1 abs(mod3L/(unit*0.1)) 1];
    F3l.translation = [0 0.5*mod3L/unit-0.025 0];
    F3r.scale = [1 abs(mod3R/(unit*0.1)) 1];
    F3r.translation = [0 0.5*mod3R/unit-0.025 0];
    if mod3L<0
        F3lS.whichChoice=1;
        F3lSp.translation = [0 mod3L/unit-0.05 0];
    elseif mod3L==0
        F3lS.whichChoice=-1;
    else
        F3lS.whichChoice=0;
        F3lSn.translation = [0 mod3L/unit 0];
    end
    if mod3R<0
        F3rS.whichChoice=1;
        F3rSp.translation = [0 mod3R/unit-0.05 0];
    elseif mod3R==0
        F3rS.whichChoice=-1;
    else
        F3rS.whichChoice=0;

```



```
F3rSn.translation = [0 mod3R/unit 0];
end
if grafF3=='s'
    F3S.whichChoice=0;
else
    F3S.whichChoice=-1;
end

% forces normals
unit=25;
modNL=entrada(4); % esquerra darrera
modNR=entrada(8); % dreta davant
modNI=0;
modND=0;
Nl.scale = [1 abs(modNL/(unit*0.1)) 1];
Nl.translation = [0 -0.5*modNL/unit-0.025 0];
Nr.scale = [1 abs(modNR/(unit*0.1)) 1];
Nr.translation = [0 -0.5*modNR/unit-0.025 0];
Ni.scale = [1 abs(modNI/(unit*0.1)) 1];
Ni.translation = [0 -0.5*modNI/unit-0.025 0];
Nd.scale = [1 abs(modND/(unit*0.1)) 1];
Nd.translation = [0 -0.5*modND/unit-0.025 0];
if modNL<0
    NlS.whichChoice=1;
elseif modNL==0
    NlS.whichChoice=-1;
else
    NlS.whichChoice=0;
end
if modNR<0
    NrS.whichChoice=1;
elseif modNR==0
    NrS.whichChoice=-1;
else
    NrS.whichChoice=0;
end

if grafNrear=='s'
    NrearS.whichChoice=0;
else
    NrearS.whichChoice=-1;
end
if grafNfront=='s'
    NiS.whichChoice=0;
    NdS.whichChoice=0;
else
    NiS.whichChoice=-1;
    NdS.whichChoice=-1;
end
% actualització del model
vrdrawnow;
end
```





11 Avaluació de Resultats

11.1 Metes assolides

Amb el sistema desenvolupat hem aconseguit:

- Captar les forces que realitza una persona a sobre d'un caminador durant el seu ús, així com les forces normals al terra que realitzen les rodes posteriors. La resolució d'aquestes forces és molt elevada (16 bits).
- Publicar les forces captades per les electròniques en un bus CAN per que puguin ser llegides des de un computador.
- Mostrar de manera gràfica, intuïtiva i en temps real el valor d'aquestes forces, amb l'objectiu de que pugui ser entès de manera clara i senzilla per qualsevol persona.

Considerem aleshores assolits els tres objectius marcats al inici d'aquest projecte final de carrera.

11.2 Millores/Ampliacions

Amb les dades que obtenim dels vuit sensors es pot fer l'estudi de com es comporten els usuaris quan volen fer que el caminador, per exemple, giri a la dreta o es pari. Amb aquest informació es podria arribar a inferir la voluntat de l'usuari de manera que es podria ajudar a dur-la a terme accionant uns motors integrats a les rodes posteriors.

També es podria donar el cas que l'usuari del caminador hagués de realitzar una certa força horitzontal cap endavant prescrita per un metge. Usant el sistema dissenyat podem fer actuar els motors de manera que l'usuari sempre tingui la mateixa resistència a l'avenç (el caminador frena més a les baixades i ajuda a les pujades).

En general el sistema dissenyat ens pot servir per a controlar diferents actuadors que depenguin de les forces que s'exerceixen a sobre del caminador.

En quan a la interfície gràfica, es podria completar fent que el caminador es moguéss de manera realista per l'entorn virtual, fent servir la cinemàtica inversa del caminador.



Igualment es podria completar la interfície ensenyant en temps real els valor de les dades captades per altres tipus de sensors, com per exemple la posició dels peus respecte al caminador, captades per un sensor de seguiment magnètic PolhemusTM, o la posició i velocitat del caminador fent servir “encoders” a les rodes.



12 Valoració econòmica

Donat que el projecte és d'investigació, i no pretén fabricar un producte final per a sortir a l mercat, no té gaire sentit parlar del seu cost total. El caminador de sèrie té un preu de 120 €, però el fet de que s'hagin dissenyat i implementat uns transductors ad hoc per a aquest caminador fa que el seu preu s'elevi molt.

De totes maneres, si que faré un estudi del cost del treball que he desenvolupat jo realitzant aquest PFC, així com el cost de tot el material i dels components electrònics que s'han necessitat.

Per a la realització de les electròniques s'ha necessitat tot el material que s'especifica a continuació. De cada un d'ells es mostra la quantitat emprada, així com el seu cost.

material	preu unitari	quantitat	preu total
planxa coure	6	1	6
condensador electrolític	1	4	4
condensador ceràmic	0,2	102	20,4
diode led	0,2	4	0,8
diode schottky	0,4	2	0,8
connector galga	0,5	8	4
connector USB	1	2	2
connector programació	0,3	2	0,6
bobina	2,4	4	9,6
resistència	0,05	78	3,9
interruptor	0,5	2	1
PGA309	7	8	56
ADS1100	2,3	8	18,4
dsPIC30F4011	11	2	22
regulador conmutat	5	2	10
transceiver USB	4,85	2	9,7
transceiver CAN	2,5	2	5
cristall	1,2	2	2,4
fusible	1,2	2	2,4
connector RJ11	0,5	4	2
			181 €

Cal tenir en compte que no tots els components d'una mateixa classe tenen el mateix preu. Per exemple, no totes les 39 resistències usades en cada una de les electròniques tenen el mateix preu. Els preus que apareixen a la taula anterior resulten de la mitjana dels que tenen tots els elements d'un mateix tipus.



Pel que fa al treball realitzat al meu PFC, podem diferenciar-lo en distintes tasques. Aquestes tenen un diferent valor dins d'un projecte d'enginyeria i, per tant, són remunerades de distinta manera.

Tasca	cost/hora	hores dedicades	cost total
Anàlisi i Disseny "Software"	35	240	8.400
Implementació "Software"	25	80	2.000
Anàlisi i Disseny "Hardware"	35	320	11.200
Implementació "Hardware"	25	160	4.000
			25.600 €



13 Planificació seguida

En l'informe previ del meu projecte final de carrera es dividia el treball a realitzar en les taques següents:

#	Acrònim	Nom	Descripció
1	CAL	Calibratge dels sensors	Mesurar el voltatge de sortida de les galgues per a diferents forces i veure d'aquesta manera quina ha de ser la configuració del condicionador de senyal.
2	PGA	Entorn condicionador de senyal	Aprendre a configurar el condicionador de senyal PGA309 per a que amplifiqui el senyal de les galgues amb el guany i el "offset" desitjats.
3	PIC	Entorn de programació del dsPIC	Familiarització amb l'entorn de programació del dsPIC (MPLAB) i desenvolupar programes capaços d'interactuar amb les diferents interfícies i perifèrics del xip.
4	PCB	Entorn de disseny de PCB's	Familiarització amb l'entorn de disseny d'un circuit electrònic (Orcad Capture) i de plaques de circuit imprès (Orcad Layout) per tal de poder generar les electròniques.
5	COM	Comunicacions I2C i CAN	Aprendre a fer servir els busos de comunicacions I2C i CAN i implementar els codis que permeten configurar el PGA309 i llegir el senyal de sortida del A/D (I2C) i publicar-lo (CAN).
6	RRV	Representació realista	Implementar una representació en realitat virtual de les forces que es realitzen a sobre del caminador per tal que pugui ser entès per qualsevol persona
7	MEM	Redacció de memòria	Redacció de la memòria del Projecte Final de Carrera

Igualment es definia la planificació en forma de diagrama de Gantt:

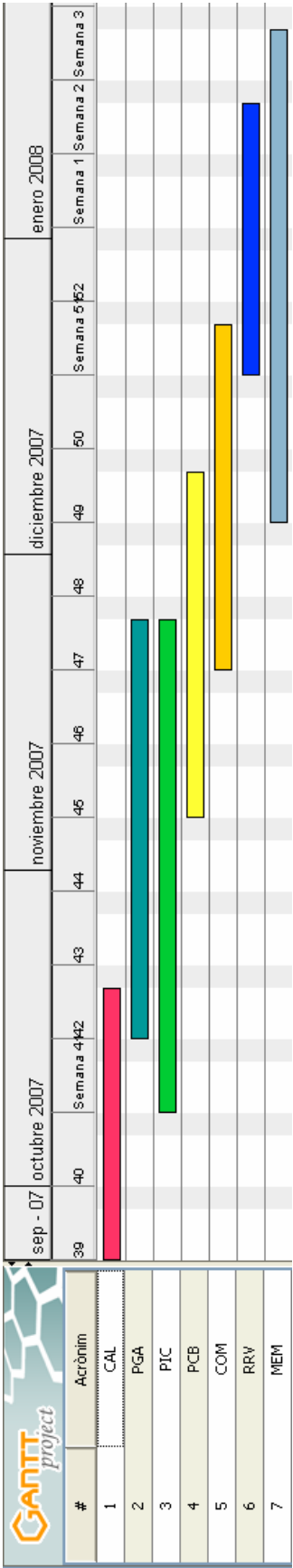


Figura 62: planificació prevista en l'informe previ



La planificació s'ha anat complint més o menys tal i com es va predir fins a les vacances de Nadal. El fet de marxar a Mallorca a veure la família em va impossibilitar fer cap treball que impliqués la meua presència al departament o l'ús de qualsevol component "hardware". Aleshores vaig optar per escriure quasi tota la memòria durant aquest període (tasca 7).

Un cop passades les vacances, em faltava per fer tota la tasca 6 (interfície de realitat virtual en temps real) així com la implementació d'una segona electrònica igual que la primera (que ja es trobava completament operativa abans de marxar), que forma part de tasca 6. La darrera setmana i mitja abans de la lectura es dedicarà a la preparació de la presentació i les demos.





14 Agraïments

Voldria donar les gracies a totes aquelles persones que m'han ajudat d'alguna manera a fer que jo pogués arribar a escriure en la pàgina 109 de la memòria del meu PFC.

En primer lloc voldria donar les gràcies a Toni Benedico per haver-me ajudat tant, i sobretot per la seva paciència; al Laure, company de batalla, per amenitzar les hores al despatx i amb qui he compartit tants moments; i al Jordi, amb qui vaig arrancar a fer feina a l'Omega i em va instruir en l'art del Word.

També vull agrair al departament de Resistència de Materials el magnífic treball que han realitzat, i a Sergi de Haro per el haver-me ajudat amb la realitat virtual de Matlab i haver-me cedit el seu treball.

Per descomptat vull agrair al meu tutor i cap Antoni B. Martinez tot el suport i guiatge, així com la confiança que ha dipositat en mi.

Finalment, a totes aquelles persones que han suportat durant aquest any totes les meves il·lusions i desil·lusions, als meus pares, germans i amics que m'han fet el camí més divertit: moltes gràcies.





15 Bibliografia

[1] “Motion Control of Intelligent Walker based on Renew of Estimation Parameters for User State”. Yasuhisa Hirata, Asami Muraki i Kazuhiro Kosuge. Departament de Bioenginyeria i Robòtica. Universitat de Tohoku. Aoba-yama 01, Sendai, Japó.

[2] “A Robotically-Augmented Walker for Older Adults”. Jared Glover, David Holstius, Michael Manojlovich et al. School of Computer Science. Carnegie Mellon University. Pittsburgh. EEUU.

[3] “Pearl: A Mobile Robotic Assistant for the Elderly”. Martha E. Pollack, Laura Brown, et al. Universitat de Michigan. EEUU. Sandra Engberg, Judith T. Matthews, et al. Universitat de Pittsburgh. EEUU. Sebastian Thrun, Michael Montemerlo, et al. Carnegie Mellon University. EEUU.

[4] “User Intent in a Shared Control Framework for Pedestrian Mobility Aids”. Glenn Wasson. Departament d’enginyeria informàtica. Universitat de Virgínia. EEUU. Majd Alwan. Centre de recerca en automatització mèdica. Universitat de Virgínia. EEUU. Kevin Granata. Departament de ciència de l’enginyeria i mecànica. Virgínia Tech. EEUU. Pradip Sheth, Alexandre Ledoux i Cunjun Huang. Departament d’enginyeria mecànica i aeroespacial. Universitat de Virgínia. EEUU.

[5] “An Assistive Robotic Agent for Pedestrian Mobility”. Glenn Wasson, Sean Graves, Robin Felder. Centre de recerca en automatització mèdica. Universitat de Virgínia Centre Mèdic. EEUU. Jim Gunderson. Departament d’enginyeria informàtica. Universitat de Virgínia. EEUU.

[6] “Stability Margin Monitoring in Steering-Controlled Intelligent Walkers for the Elderly”. Majd Alwan. Centre de recerca en automatització mèdica. Universitat de Virgínia. EEUU. Prabhu Jude Rajendran. Departament d’enginyeria elèctrica i informàtica. Universitat de Virgínia. EEUU. Alexandre Ledoux, Cunjun Huang, Pradip Sheth. Departament d’enginyeria mecànica i aeroespacial. Universitat de Virgínia. EEUU. Glenn Wasson. Departament d’enginyeria informàtica. Universitat de Virgínia. EEUU.



PGA309 datasheet

PGA309 user's manual

ADS1100 datasheet

<http://www.comunidadelectronicos.com/articulos/i2c.htm>

<http://www.microchip.com>

dsPIC30F family reference manual

dsPIC30F4011 datasheet

www.utilcell.es

www.wikipedia.org



Annex 1: Pont de Wheatstone

Un pont de Wheatstone és un instrument elèctric de mesura inventat per Samuel Hunter Christie l'any 1832, millorat i popularitzat per Sir Charles Wheatstone en 1843. S'utilitza per a mesurar resistències desconegudes mitjançant l'equilibri dels braços del pont. Aquests estan constituïts per quatre resistències que formen un circuit tancat, essent una d'elles la resistència mesurada. La figura següent il·lustra la disposició elèctrica del pont:

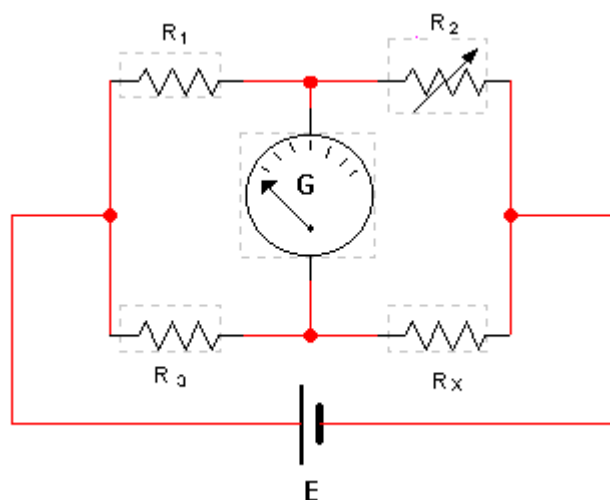


Figura 63: pont de Wheatstone

A la figura R_x és la resistència de la que volem conèixer el valor, R_1 , R_2 i R_3 són resistències de valors coneguts i a més R_2 es ajustable. Si la relació de les dues resistències del braç conegut (R_2/R_1) és igual a la relació de les dues del braç desconegut (R_x/R_3), el voltatge entre els dos punts del mig serà nul i per tant no circularà corrent entre aquests punts.

Per a efectuar la mesura el que es fa es variar la resistència R_2 fins aconseguir l'equilibri. La detecció del corrent nul es pot fer amb gran precisió fent servir un galvanòmetre. La direcció del corrent, en cas de desequilibri, indica si R_2 és massa gran o massa petita. En cas d'equilibri sempre es compleix que

$$R_x = \frac{R_2 \times R_3}{R_1}$$

Si els valors de R_1 , R_2 i R_3 es coneixen amb molta precisió el valor de R_x podrà ser determinat amb igual precisió. Petits canvis en el valor de R_x rompran l'equilibri i podran ser detectats usant el galvanòmetre.



De forma alternativa, si els valors de R_1 , R_2 i R_3 són coneguts i R_2 no és ajustable, el corrent que passa pel galvanòmetre pot ser utilitzada per a determinar el valor de R_x .

Variants del pont de Wheatstone poden ser usades per a mesurar impedàncies, capacitàncies i inductàncies.

La disposició en pont també es usada en instrumentació electrònica. D'aquesta manera, es substitueixen una o més resistències per sensors, que al variar la seva resistència provoquen una variació proporcional en la sortida del pont. Usant un amplificador a la sortida es pot mesurar en gran precisió aquesta variació.



Annex 2: Amplificadors Operacionals

Un amplificador operacional (A.O., normalment anomenat op-amp) és un circuit electrònic (es pot presentar en forma de circuit integrat) que té dues entrades i una sortida. La sortida és la diferència de les dues entrades multiplicada per un factor (G) que és el guany:

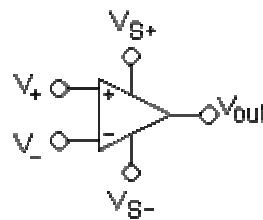
$$V_{out} = G \cdot (V_+ - V_-)$$

El primer amplificador operacional monolític data dels anys 1960, era el Fairchild μ A702 (1964), dissenyat per Bob Widlar. El va seguir el Fairchild μ A709 (1965), també de Widlar, i que va constituir un gran èxit comercial. Més endavant va ser substituït pel popular Fairchild μ A701 (1968), de David Fullagar, i fabricat per nombroses empreses, basat en tecnologia bipolar.

Originàriament els A.O. s'usaven per a les operacions matemàtiques (suma, resta, multiplicació, divisió, integració, derivació, etc.) en calculadores analògiques. D'aquí el seu nom.

El A.O. ideal té un guany infinit, una impedància d'entrada infinita, un ample de banda també infinit, una impedància de sortida nul·la, un temps de resposta nul i cap soroll. Com la impedància d'entrada és infinita també es diu que els corrents d'entrada són zero.

El símbol d'un A.O. es mostra a la figura següent:



Els terminals són:

- V_+ : entrada no inversora
- V_- : entrada inversora
- V_{out} : sortida
- V_{S+} : alimentació positiva
- V_{S-} : alimentació negativa

Els terminals d'alimentació poden rebre diferents noms, per exemple en els A.O. basats en FET V_{DD} i V_{SS} respectivament, i per als basats en BJT són V_{CC} i V_{EE} .

Normalment s'ometen els pins d'alimentació en els diagrames elèctrics per claredat.



Comportament en corrent continua

Llaç obert

Si no existeix realimentació la sortida del A.O. serà la resta de les seves dues entrades multiplicada per un factor. Aquest factor sol ser de l'ordre de 100.000 (que es considerarà infinit en càlculs amb el component ideal). Per tant si la diferència entre les dues tensions és de 1mV, la sortida hauria de ser de 100V. Degut a la limitació que suposa el fet de no poder entregar més tensió que la que hi ha l'alimentació, el A.O. estarà saturat si es dona aquest cas. Si la tensió més alta és l'aplicada a la patilla + la sortida serà la que correspon a l'alimentació V_{s+} , i si la tensió més elevada és la de la patilla - la sortida serà l'alimentació V_{s-} .

Llaç tancat

Es coneix com a llaç la realimentació en un circuit. Aquí suposarà realimentació negativa. Per a conèixer el funcionament d'aquesta configuració es parteix de les tensions de les dues entrades exactament iguals, es suposa que al tensió en la patilla + puja i, per tant, la tensió en la sortida també s'eleva. Com existeix la realimentació entre la sortida i la patilla -, la tensió en aquesta patilla també s'eleva, i per tant la diferència entre les dues entrades es redueix, disminuint també la sortida. Aquest procés aviat s'estabilitza i es té que la sortida és la necessària per a mantenir les dues entrades, idealment, amb el mateix valor.

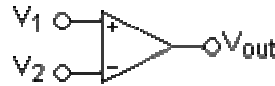
Sempre que hi ha realimentació negativa s'apliquen aquestes dues aproximacions per a analitzar el circuit:

- $V_+ = V_-$
- $I_+ = I_- = 0$



Configuracions

Comparador

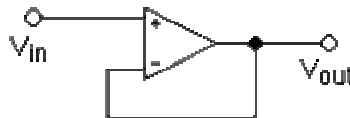


- És una aplicació sense realimentació. Compara entre les dues entrades i treu la sortida en funció de quina entrada sigui major. Es pot usar per a adaptar nivells lògics.

$$V_{out} = \begin{cases} V_{s+} & V_1 > V_2 \\ V_{s-} & V_1 < V_2 \end{cases}$$

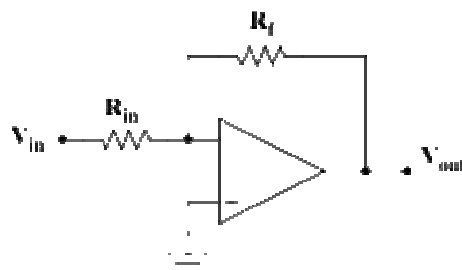
Seguidor

Circuit que proporciona a la sortida la mateixa tensió que a l'entrada



- S'usa com a buffer, per a eliminar efectes de càrrega o per a adaptar impedàncies (connectar un dispositiu amb gran impedància a un altre amb baixa impedància i bisa versa)
- La tensió entre les dues patilles d'entrada és la mateixa
- $Z_{in} = \infty$.

Inversor



L'anàlisi del circuit és el següent:

- $V_+ = V_- = 0$



- Definint corrents: $\frac{V_{in} - 0}{R_{in}} = -\frac{V_{out} - 0}{R_f}$ i d'aquí es treu que

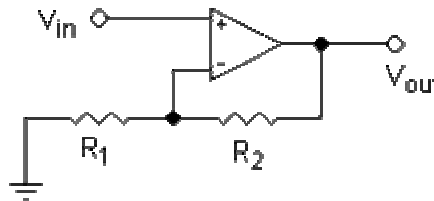
$$V_{out} = -V_{in} \frac{R_f}{R_{in}}$$

- Per a la resta dels circuits l'anàlisi és similar
- $Z_{in} = R_{in}$

Es pot controlar la impedància d'entrada mitjançant l'elecció de R_1

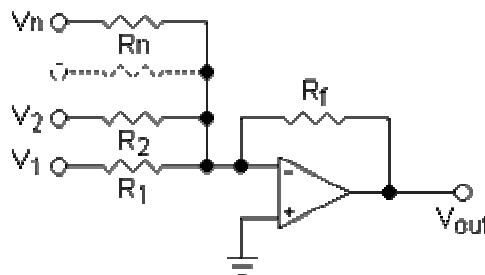
- $V_{out} = -\frac{R_2}{R_1} * V_{in}$

No inversor



- $V_{out} = V_{in} \left(1 + \frac{R_2}{R_1} \right)$
- $Z_{in} = \text{infinit}$

Sumador inversor



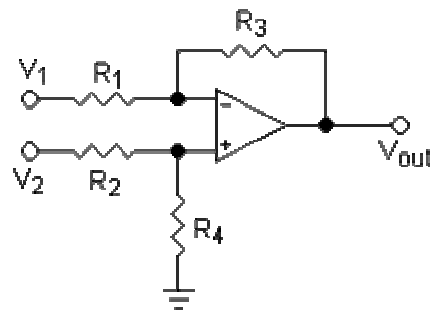
- La sortida està invertida
- Per a resistències independents R_1, R_2, \dots, R_n

$$V_{out} = -R_f \left(\frac{V_1}{R_1} + \frac{V_2}{R_2} + \dots + \frac{V_n}{R_n} \right)$$

- L'expressió es simplifica bastant si s'usen resistències del mateix valor
- Impedàncies d'entrada: $Z_n = R_n$



Restador

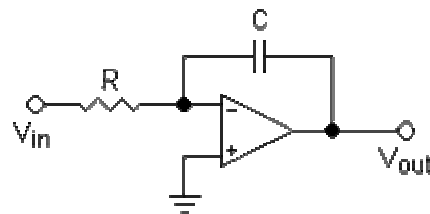


- Per a resistències R_1 , R_2 , R_3 , R_4 :

$$V_{out} = V_2 \left(\frac{(R_3 + R_1)R_4}{(R_4 + R_2)R_1} \right) - V_1 \left(\frac{R_3}{R_1} \right)$$

- L'expressió es pot simplificar amb resistències iguals
- La impedància diferencial entre dues entrades és $Z_{in} = R_1 + R_2$

Integrador ideal



- Integra i inverteix la senyal (V_{in} i V_{out} són funcions dependents del temps)

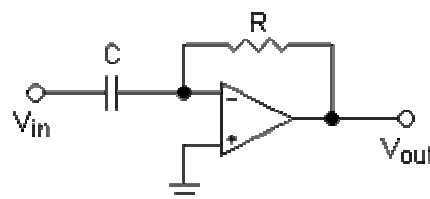
- $$V_{out} = \int_0^t -\frac{V_{in}}{RC} dt + V_{inicial}$$

- $V_{inicial}$ és la tensió de sortida a l'origen de temps ($t=0$)

- Aquest circuit també s'usa com a filtre

NOTA: a la pràctica es realitzen modificacions al circuit ja que no és estable.

Derivador ideal



- Deriva i inverteix la senyal respecte al temps



- $V_{out} = -RC \frac{dV_{in}}{dt}$

- Aquest circuit també s'usa com a filtre

NOTA: a la pràctica no s'utilitza per que no és estable

