

## Treball final de grau

**Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica**

**Títol: Actualització del robot Pioneer 2-DX amb MyRIO**

**Document:** 1. Memòria

**Alumne:** Lluís Teixidor Arguimbau

**Director/Tutor:** Bianca M. Innocenti Badano  
**Departament:** Enginyeria Elèctrica, Electrònica i Automàtica  
**Àrea:** ESA

**Convocatòria** (mes/any): Setembre/2015

## ÍNDEX

1	INTRODUCCIÓ .....	3
1.1	Antecedents .....	3
1.2	Objecte .....	3
1.3	Especificacions i abast .....	4
2	EL ROBOT PIONEER 2DX .....	5
2.1	Generalitats .....	5
2.2	El sistema de control .....	6
2.2.1	La placa del microcontrolador .....	6
2.2.2	La placa de control dels motors .....	7
2.2.3	Placa de guany dels ultrasons .....	8
2.2.4	Els motors .....	8
2.2.5	Els encoders del robot .....	9
3	NI MYRIO .....	11
4	MODELS MATEMÀTICS DEL ROBOT .....	13
4.1	Model de tracció .....	13
4.2	Representació de la posició en el pla .....	14
4.3	Model cinemàtic del robot .....	17
4.4	Velocitat al centre de referència del robot (centre de masses) .....	19
5	PROVES DE FUNCIONAMENT AMB EL MICROCONTROLADOR ORIGINAL .....	21
5.1	Motors .....	21
5.2	Sensors d'ultrasò .....	22
5.2.1	Conclusions de l'anàlisi .....	25
6	DISSENY DE LA FPGA .....	26
6.1.1	Motors .....	26
6.1.2	Encoders .....	28
6.1.3	Ultrasons .....	29
7	ARQUITECTURA MULTIAGENT .....	31
7.1	Programació del sistema de control .....	32
7.1.1	Ultrasound agent .....	33
7.1.2	Encoder agent .....	33
7.1.3	Goto agent .....	34
7.1.4	Avoid agent .....	35
7.1.5	Gothrough agent .....	37
7.1.6	Selecció i aplicació dels senyals .....	38
7.1.7	Interfície d'usuari .....	41

---

7.1.8	Visualització .....	42
8	PROVES DE FUNCIONAMENT .....	43
8.1	Paràmetres de comunicació.....	43
8.2	Resultats obtinguts .....	43
8.2.1	Mapa Go to .....	43
8.2.2	Mapa Go to i Avoid .....	46
8.2.3	Mapa Go to, Avoid i Gothrought .....	47
8.2.4	Conclusions de les proves.....	50
9	RESUM DEL PRESSUPOST .....	51
10	CONCLUSIONS .....	52
11	RELACIÓ DE DOCUMENTS.....	53
12	BIBLIOGRAFIA .....	54

# 1 INTRODUCCIÓ

## 1.1 Antecedents

Durant anys, el grup de recerca eXit de la UdG ha desenvolupat una arquitectura de control pel robot Pioneer 2DX, un robot mòbil de 3 rodes. Actualment, aquest robot s'ha deixat de fabricar. Tot i que hi ha nous models al mercat, el Pioneer 2DX ha quedat obsolet i per tant ja no es disposa de millores de firmware.

Existeixen diversos projectes realitzats anteriorment amb aquest robot. Primer, el projecte de Gemma Pou titulat "Implementació dels comportaments bàsics de navegació pel robot GRILL utilitzant el Sistema Operatiu de Temps Real QNX", el projecte de la Silvia Garcia titulat " Implementació d'una arquitectura multi-agent reactiva per al control del Grill",el projecte de la Encarnació González on es van aconseguir dissenyar un únic agent utilitzant la lògica difusa o el projecte de Gerard Alarcón titulat "Disseny de controladors òptims per al robot Pioneer". També com a precedent al 2008 la Dra. Bianca Innocenti va presentar la seva tesi doctoral anomenada "A Multi-agent Architecture with Distributed Coordination for an Autonomous Robot"

Tot i que s'han realitzat tots els estudis anteriors entre altres, aquest projecte no pretén continuar desenvolupant cap dels projectes anteriors, ja que com que ens proposem substituir la placa de control, s'haurà de dissenyar un nou sistema en un llenguatge de programació totalment diferent al que s'utilitzava en aquests projectes. Per altre banda, sí que s'han utilitzat alguns d'aquests com a models de consulta per dissenyar el sistema de control.

## 1.2 Objecte

El propòsit d'aquest projecte és actualitzar aquest robot. Podem trobar dos objectius principals. Primer la substitució de la placa de control del fabricant, per un sistema "embedded" de National Instruments (MyRIO). I segon, el disseny i implementació del sistema de control per aconseguir que el robot es pugui desplaçar d'un punt a un altre esquivant els objectes que es pugui trobar en el seu recorregut.

### **1.3 Especificacions i abast**

Per assolir els objectius, el sistema haurà d'accedir als sensors i als motors del robot per comprendre la seva naturalesa i adaptar els senyals necessaris a les necessitats del sistema. Pel assolir el segon objectiu, s'adaptaran els mòduls existents de posicionament i elusió d'obstacles per adaptar-los al nou sistema de programació. Com que els agents originals formen part d'una arquitectura multi-agent, han de comunicar-se amb un host remot i seguir el mateix llenguatge de comunicació.

## 2 EL ROBOT PIONEER 2DX

### 2.1 Generalitats

EL robot Pioneer 2DX és un robot mòbil de tres rodes. D'aquestes tres, dues són motrius i una és la conduïda.

El robot és d'alumini i el seu pes és de 16'55kg sense bateries. Les seves dimensions són 44cm de llargada, 33 cm d'amplada i 22 cm d'alçada.



Figura 1. Pioneer 2DX

Disposa de tres bateries de 12 V a la seva part posterior que li aporten una autonomia de unes 8-10 hores. A la part posterior, també hi ha situada la roda conduïda que disposa d'un grau de llibertat que permet que la roda giri lliurement sobre un eix vertical diferent de l'eix de rotació de la roda. Quasi a la part frontal, s'hi troben les dues rodes motrius alineades en el mateix eix de rotació. Són rodes independents que funcionen amb motors de corrent continua. La velocitat màxima a la que pot arribar el robot és de 1,6m/s o 300graus/s. Cada un dels motors de les rodes, porta incorporat un encoder òptic.

Finalment, a la part davantera hi trobem els ultrasons. El cinturó d'ultrasons, està format per 8 sondes. La disposició d'aquestes es compon d'una a cada costat (a 90° del centre) i entre elles trobem les 6 restants cada 20°. La freqüència d'adquisició del sonar es de 25Hz i la sensibilitat d'aquestes varia entre els 10cm i els 5m.

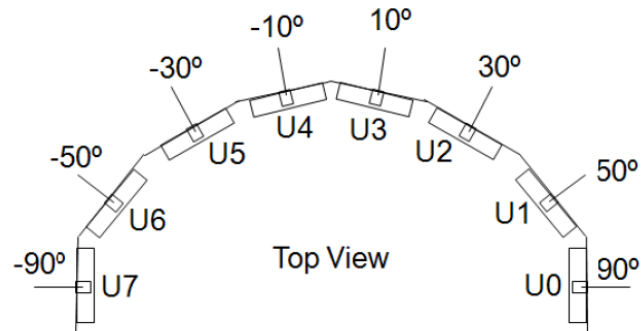


Figura 2. Disposició sensors

## 2.2 El sistema de control

Per controlar les diferents parts del robot, físicament disposa de tres plaques de control. La placa del microcontrolador, la placa de control dels motors i la placa d'aplicació del guany dels sensors d'ultrasò.

### 2.2.1 La placa del microcontrolador

El microcontrolador de la placa de control és un Siemens SAB 88C166-5M i s'encarrega de realitzar totes les funcions de control amb una velocitat màxima de 20MHz.

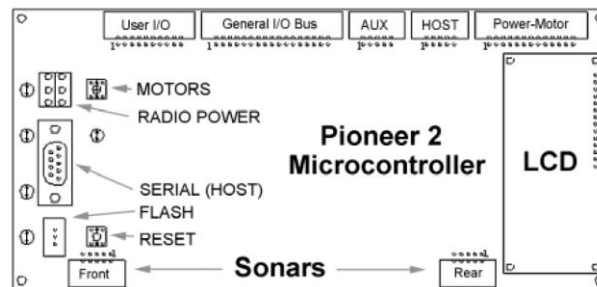


Figura 3. Placa de control original

Com es pot observar a la imatge anterior, aquesta placa disposa de diversos connectors. Els més importants per aquest projecte són el Power-Motor i el Front. El primer genera els senyals necessaris per controlar els motors i rep les dades que li envien els encoders. La segona, es comunica amb els dispositius d'ultrasò per realitzar el seu control i l'adquisició dels senyals per trobar les distàncies. També disposa del connector AUX que ens llegeix el nivell de càrrega de les bateries.

### 2.2.2 La placa de control dels motors

Aquesta placa s'alimenta amb una tensió que pot variar entre 5 i 10 volts. Conté dos connectors que comunica amb els encoders de cada motor i un altre que els alimenta. També es pot observar el connector de comunicació amb el microcontrolador.

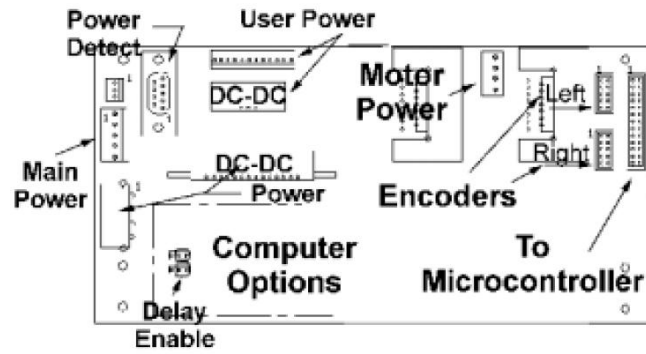


Figura 4. Placa de control dels motors

Aquests motors estan controlats per dos dispositius LMD18245. Aquests incorporen tots els blocs necessàries per controlar un motor de corrent continua o un motor pas a pas bipolar amb tecnologia CMOS. També incorpora blocs de protecció contra sobre intensitats i una protecció contra la temperatura que s'activa als 255°C.

Aquest dispositiu ens pot proporcionar una tensió de fins a 55V i una intensitat de 3A i fins a 6A de pic. Si el la tensió cau per sota dels 10 volts el circuit es desconnecta automàticament.

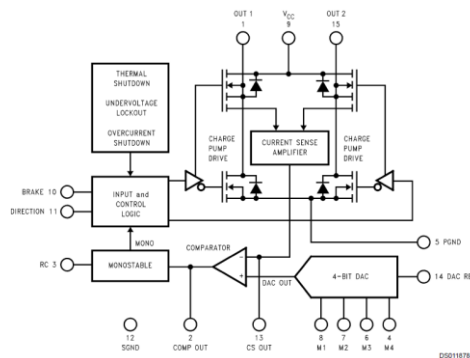


Figura 5. Diagrama de blocs LMD18245



### 2.2.3 Placa de guany dels ultrasons

El cinturó d'ultrasons al igual que els motors, disposa d'una circuit per ajustar el guany i així poder adaptar els senyals al medi. El potenciòmetre que es pot observar a la figura següent ajusten la sensibilitat dels sensors, és a dir, si reduïm el seu valor, reduïrem la capacitat de detectar petits obstacles. Aquesta posició serveix per ajustar els sensors a un entorn amb molts sorolls o amb un alt índex de reflexió. Si s'augmenta el valor, augmentem la percepció dels petits obstacles a una distància més gran. Serveix per treballar millor en entorns amb pocs obstacles o espais oberts.

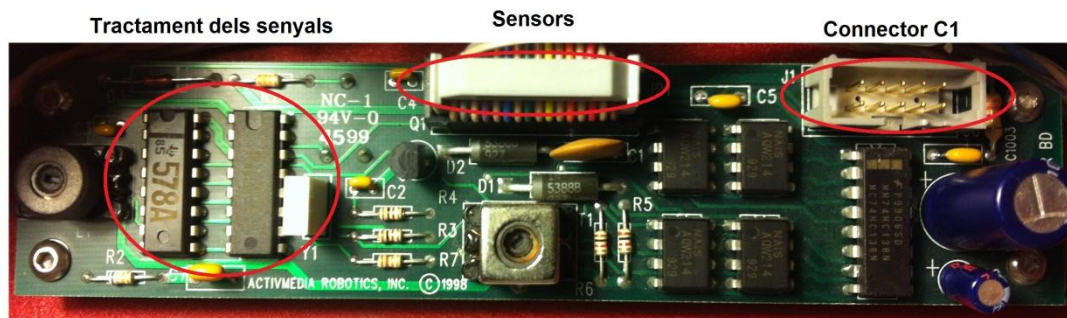


Figura 6. Placa de tractament d'ultrasons

Com podem veure a la figura 6, amb el connector C1 la placa rep el senyal necessari per seleccionar la sonda pertinent. Per fer-ho, els senyals han de passar per el selector i els 4 relés dels que disposa el bloc de selecció de la sonda de la figura 6.

Un cop s'ha activat el sensor, el senyal s'emeteix directament sense realitzar cap tractament, directament passa cap al connector Sensors (Figura 6) per ser emès.

Quan es rep la resposta dels sensors, aquesta passa pel bloc de tractament on el 578A i a continuació un TI851 apliquen el guany que s'ha establert.

Per acabar les dades passen pel selector i a través del C1 són rebudes per la placa de control.

### 2.2.4 Els motors

El sistema de tracció del robot està compost de dos motors DC Pittman GM9236E132 de 12 Vdc, situats en un mateix eix, però contraposats un de l'altre. Aquest model de robot s'anomena model diferencial. La figura 7 ens mostra els motors contraposats del Pioneer.



Figura 7. Motors amb el seu circuit de control

Resulta difícil trobar informació d'aquest motor, ja que es tracta d'un model obsolet. Avui de la sèrie GM9000 que és a la que pertany, fabriquen altres models amb característiques molt semblants. Això és degut a que són motors molt precisos i per aplicacions concretes que disposen de parts dissenyades expressament per un producte en concret.

La referència del motor està formada per dues parts, la GM9236 ens marca el model de motor i la part E132 l'encoder que porta incorporat. També disposa d'una referència suplementària, 19:7:1 que ens defineix la relació de transmissió interna del motor.

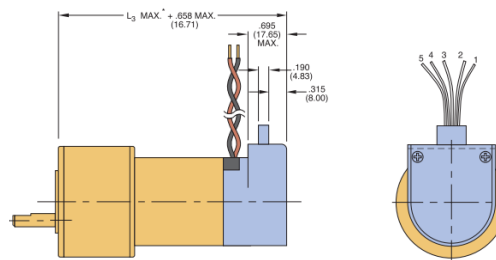


Figura 8. Motor i mòdul encoder i transmissió

Com es pot veure a la imatge, a la zona dreta trobem l'encoder del que disposa el robot i a l'esquerra el mòdul amb la relació de transmissió.

### 2.2.5 Els encoders del robot

Cada un dels motors disposa d'un encoder òptic de quadratura d'alta definició que ens permet llegir la posició dels motors en tot moment. Per adquirir els senyals de cada un d'aquests dispositius s'utilitzen dos pins amb els senyals dels dos canals A i B de l'encoder, un pin d'alimentació a 5 volts i la massa.

Pins encoder	Color cable	Funció
1	Negre	Massa
2	Groc	Canal A
3	Vermell	Vcc
4	Blau	Canal B

Taula 1. Pins encoder

Es tracta d'un sistema amb codificació X4, per tant, cada cicle del senyal del dispositiu resultarà en 4 senyals o canvis que el comptador del dispositiu ens sumarà. Aquest encoder en concret ens donarà 39400 senyals per revolució, o dit d'una altra manera, si tenim en compte que la roda del robot té un radi de 82.5mm, l'encoder ens donarà 76 ticks per mm de desplaçament.

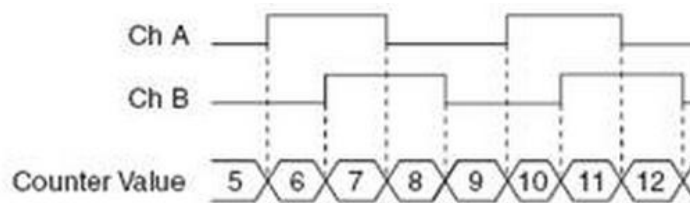


Figura 9. Model X4

### 3 NI MYRIO

El NI MyRIO-1900 és un dispositiu embedded dirigit al món de la docència. Per el seu funcionament, aquest dispositiu el podem descriure en 4 blocs. Un processador pel Real Time, un bloc FPGA reconfigurable, les entrades i sortides, i un sistema de programació gràfic (Labview), tot i que també es pot realitzar programació amb llenguatge C.



Figura 10. MyRIO

El microprocessador que incorpora aquest dispositiu és un Xilinx Z-7010 que inclou un processador de doble nucli ARM Cortex - A9 i una FPGA Artix -7, i que permet arribar fins a una velocitat de 667MHz. A més, em model Z-7010 permet la connexió Wi-fi o port sèrie via USB. També disposa de 256MB de memòria, una memòria DDR3 de 512MB, una freqüència de mostreig de 533MHz i una amplada de bus de dades de 16 bits.

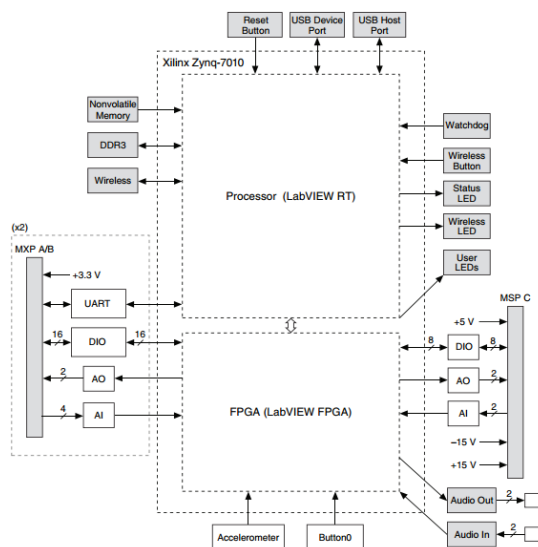


Figura 11. Blocs MyRIO

Les entrades i sortides estan dividides en tres blocs, A, B i C. Entre altres, el sistema disposa de deu entrades i deu sortides analògiques (amb una resolució de 12 bits), quaranta entrades i sortides digitals, dos botons, quatre leds, un acceleròmetre intern, una entrada USB, una entrada i una sortida d'àudio, etc.

## 4 MODELS MATEMÀTICS DEL ROBOT

### 4.1 Model de tracció

Dins del camp dels robots mòbils, existeix una gran varietat de configuracions, cada una d'elles disposa de les seves peculiaritats i limitacions. Podem trobar el robot tipus cotxe (o Ackerman) que disposa de quatre rodes, dues al davant que controlen la direcció i dues al darrere que s'encarreguen de la tracció, el robot amb configuració tricicle, amb dues rodes motrius al darrere i una que controla la direcció al davant, els robots omnidireccionals que normalment disposen de tres rodes amb les funcions de direcció i tracció, etc.

El model d'estudi que utilitzarem disposa d'una configuració tipus diferencial. Aquesta configuració, és una de les més utilitzades, ja que necessites pocs components i és una de les més senzilles de controlar.

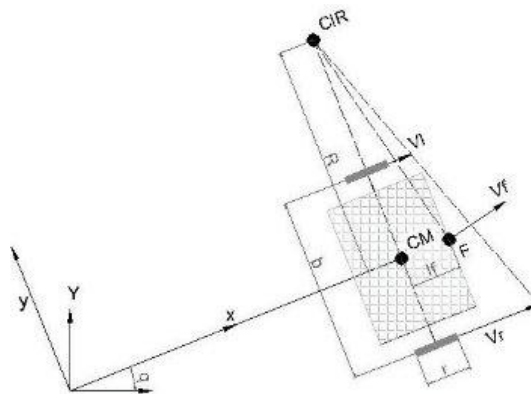


Figura 12. Configuració diferencial

Aquesta configuració estar formada per dos motors amb velocitats independents situats en un mateix eix. Aquests motors són els que realitzen tant la funció de tracció com la de direcció. La variació de direcció s'aconsegueix ajustant la diferència de velocitats entre les dues rodes i quan aquesta sigui 0, anirà recte. Amb aquesta configuració, el robot també ens permet realitzar la variació d'orientació sense que hi hagi moviment de translació.

Habitualment podem trobar aquesta configuració complementada amb rodes de moviment lliure que serveixen per facilitar el moviment del robot. Quants menys graus de llibertat tingui menys influirà en el control de direcció del robot.

## 4.2 Representació de la posició en el pla

Per tal de definir la posició en un robot mòbil, és necessària conèixer les coordenades del pla  $x$  i  $y$ , i també l'angle d'orientació del robot respecte un sistema de coordenades fix.

Per estudiar una mica el sistema de representació de la posició d'un robot partirem d'un sistema de coordenades en el que s'hi pot expressar la posició del robot com un vector de posició.

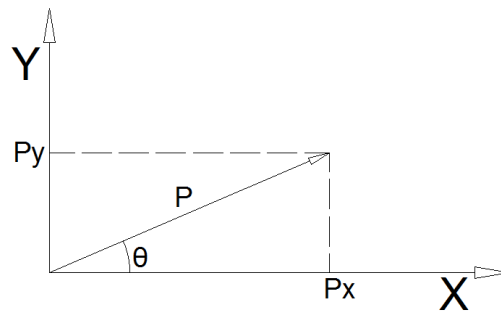


Figura 13. Sistema de coordenades fixa

Per tant, a partir d'aquest sistema, podem definir el vector posició com la següent expressió.

$${}^A P = \begin{pmatrix} {}^A P_x & {}^A P_y \end{pmatrix} \quad (\text{Eq. 1})$$

A continuació hem de situar un sistema de coordenades B que situat en el centre de referència (normalment centre de massa) del robot.

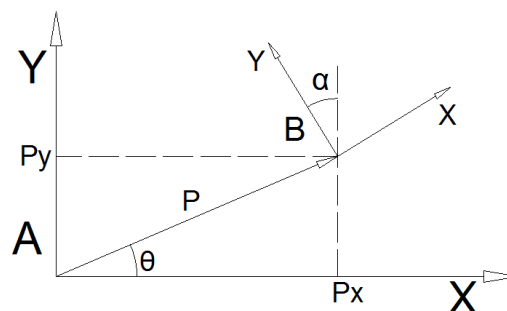


Figura 14. Sistema de coordenades fixa i mòbil

Com podem observar a la figura 3, l'eix Y del sistema B forma un angle amb X. Si expressem els dos vectors unitaris del sistema B al sistema A ens queden les expressions següents.

$${}^A X_B = \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \quad (\text{Eq. 2})$$

$${}^A Y_B = \begin{pmatrix} -\sin \alpha \\ \cos \alpha \end{pmatrix} \quad (\text{Eq. 3})$$

Aquests dos vectors formen la matriu de rotació del robot.

$${}^A R_B = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \quad (\text{Eq. 4})$$

Per conèixer la posició en el sistema A de l'origen de coordenades del sistema B, s'haurà de realitzar una translació dels eixos de B. Per tant, partirem de la suposició que no hi ha moviment de rotació dels eixos de B sobre A.

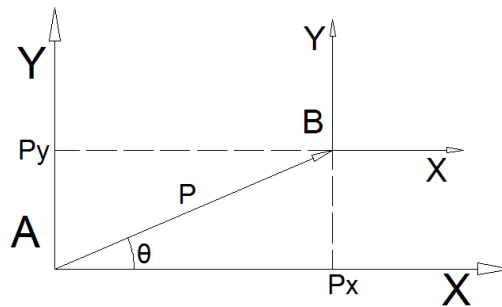


Figura 15. Sistema de coordenades sense rotació de B

El punt d'origen del sistema B el podem representar com el vector següent.

$${}^A P_{\text{origen}_B} = \begin{bmatrix} {}^A P_{X_{\text{origen}_B}} & {}^A P_{Y_{\text{origen}_B}} \end{bmatrix} \quad (\text{Eq. 5})$$

D'aquesta manera, podem conèixer la posició del robot com la suma del punt d'origen del robot (o origen del sistema de coordenades B) i el punt destí representat pel vector de posició del sistema B.

$${}^A P_X = {}^B P_X + {}^A P_{X_{\text{origen}_B}} \quad (\text{Eq. 6})$$

$${}^A P_Y = {}^B P_Y + {}^A P_{Y_{\text{origen}_B}} \quad (\text{Eq. 7})$$



Si el robot ja ha adquirit l'orientació necessària fins a un punt objectiu determinat, podem definir la posició d'aquest punt representat des del sistema de referència B amb les coordenades següents.

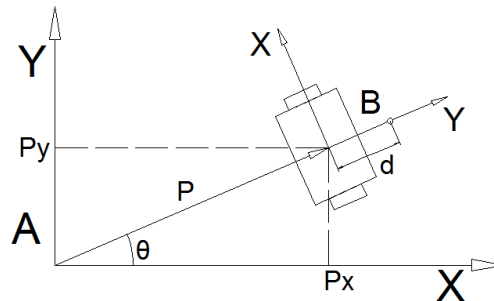


Figura 16. Punt objectiu

$${}^B P = \begin{pmatrix} 0 \\ d \end{pmatrix} \quad (\text{Eq. 8})$$

A continuació hem de realitzar les operacions necessàries per passar del sistema de referència B a l'A. Per realitzar-ho haurem de multiplicar per l'equació 2 i 3.

$${}^A P_{Xorigen\_A} = \begin{pmatrix} \cos \alpha \\ -\sin \alpha \end{pmatrix} \cdot \begin{pmatrix} 0 \\ d \end{pmatrix} = d \cdot \sin \alpha \quad (\text{Eq. 9})$$

$${}^A P_{Yorigen\_A} = \begin{pmatrix} \sin \alpha \\ \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} 0 \\ d \end{pmatrix} = d \cdot \cos \alpha \quad (\text{Eq. 10})$$

Si expressem les coordenades X i Y en el sistema A obtindrem les equacions per trobar qualsevol punt de destí amb referència al sistema de coordenades A.

$${}^A P_X = {}^B P_X + {}^A P_{Xorigen\_B} = {}^A P_{Xorigen\_B} - d \cdot \sin \alpha \quad (\text{Eq. 11})$$

$${}^A P_Y = {}^B P_Y + {}^A P_{Yorigen\_B} = {}^A P_{Yorigen\_B} - d \cdot \cos \alpha \quad (\text{Eq. 12})$$

### 4.3 Model cinemàtic del robot

Per realitzar el sistema mòbil del robot, s'ha partit d'algunes hipòtesis que ens simplifiquen el sistema.

Suposarem que es tracta d'un robot amb un model de tracció diferencial pur. Per tant, el moviment del robot es degut als dos motors amb moviment independent. També partirem del principi que el robot es mou per una superfície plana i no tindrem lliscament de rodes. Per desplaçar-se fins a un punt concret fins el següent, la trajectòria del robot seguirà un arc de circumferència.

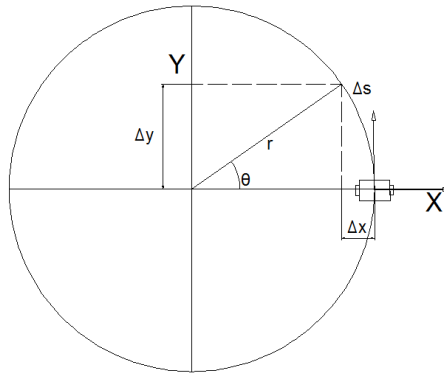


Figura 17. Següent punt de posició en un interval de control

Tal i com hem estudiat a l'apartat anterior, per determinar la posició del mòbil, partim d'un sistema de referència fix i un sistema de referència que es mou amb el mòbil. Si el mòbil es mou seguint un arc de circumferència, podem descriure la velocitat del robot amb dues equacions. Una per representar la velocitat lineal i l'altre per l'angular.

$$\text{Velocitat\_lineal} = v = \frac{\Delta s}{\Delta t} \quad (\text{Eq. 13})$$

$$\text{Velocitat\_angular} = w = \frac{\Delta \theta}{\Delta t} \quad (\text{Eq. 14})$$

En aquestes equacions, s és la distància recorreguda,  $\theta$  és el canvi d'orientació i t és l'interval de mostreig del control.

També podem saber la longitud recorreguda en un interval de temps. Aquesta distància la podem definir en funció de l'angle de l'arc i el radi de gir.

$$\Delta s = \Delta\theta \cdot r \quad (\text{Eq. 15})$$

A partir de la figura 17, s'han determinat les equacions del punt inicial del robot.

$$\Delta x = -(r - r \cdot \cos(\Delta\theta)) \quad (\text{Eq. 16})$$

$$\Delta y = r \cdot \sin(\Delta\theta) \quad (\text{Eq. 17})$$

Si les multipliquem per la matriu de rotació trobem les equacions del punt inicial des del sistema de referència fixa.

$$\Delta x = r \cdot (\cos(\Delta\theta) - 1) \cdot \cos(\theta) - r \cdot \sin(\Delta\theta) \cdot \sin(\theta) \quad (\text{Eq. 18})$$

$$\Delta y = r \cdot (\cos(\Delta\theta) - 1) \cdot \sin(\theta) + r \cdot \sin(\Delta\theta) \cdot \cos(\theta) \quad (\text{Eq. 19})$$

Si ara suposem que l'interval de temps és prou petit com perquè la variació de l'angle sigui 0, podrem simplificar les equacions.

$$\Delta x = r \cdot \Delta\theta \cdot \sin(\theta) \quad (\text{Eq. 20})$$

$$\Delta y = r \cdot \Delta\theta \cdot \cos(\theta) \quad (\text{Eq. 21})$$

A continuació, substituïm  $\Delta\theta$  per l'equació 12 que ens defineix la distància de l'arc recorregut.

$$\Delta x = \Delta s \cdot \sin(\theta) \quad (\text{Eq. 22})$$

$$\Delta y = \Delta s \cdot \cos(\theta) \quad (\text{Eq. 23})$$

Aquestes equacions ens defineixen la posició del següent punt en un interval de mostreig, per tant, si ara dividim les expressions pel temps de mostreig, obtindrem la velocitat en x i en y a cada un dels punts de coordenades.

$$\dot{x} = v \cdot \sin(\theta) \quad (\text{Eq. 24})$$

$$\dot{y} = v \cdot \cos(\theta) \quad (\text{Eq. 25})$$

#### 4.4 Velocitat al centre de referència del robot (centre de masses)

Per tal de poder aplicar les deduccions dels apartats anteriors, necessitem conèixer la relació de velocitat que hi ha entre les dues rodes per així, treure els valors de velocitat al centre de referència del robot (normalment centre de masses).

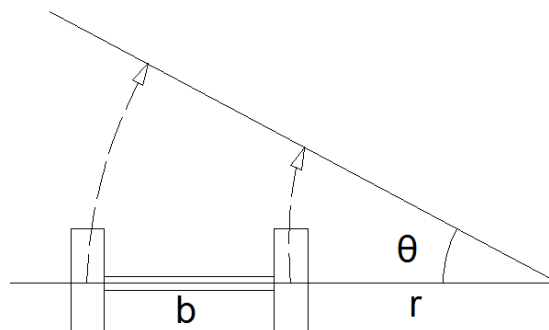


Figura 18. Velocitats

Si observem la il·lustració anterior, s'observa que les velocitats lineals de les rodes giren perpendicularment respecte a un eix. Per tant, la velocitat al centre de masses del robot també girarà perpendicularment a aquest eix (CIR). D'aquesta manera, la velocitat lineal del robot serà el producte de la velocitat angular del mòbil i el centre de masses respecte del centre instantani de rotació.

$$V_{CM} = \dot{w} \cdot \left( r + \frac{b}{2} \right) \quad (\text{Eq. 26})$$

Seguint el mateix plantejament, la velocitat lineal de cada una de les rodes serà la següent:

$$V_R = \dot{w} \cdot r \quad (\text{Eq. 27})$$

$$V_L = \dot{w} \cdot (r + b) \quad (\text{Eq. 28})$$

Si mirem la relació que hi ha entre les dues velocitats trobem l'equació que ens relaciona aquestes velocitats amb la del centre de masses del robot.

$$V_{CM} = \frac{V_R + V_L}{2} \quad (\text{Eq. 29})$$

També podem trobar la relació entre les velocitats de cada roda i la velocitat angular al centre de referència del robot, ja que la diferència de velocitats entre les rodes és la component lineal de la velocitat angular del robot i per tant, si dividim aquesta component per el radi de gir del robot ,en el nostre cas “b”, obtindrem la velocitat angular al centre de referència.

$$W_{CM} = \frac{V_L - V_R}{b} \quad (\text{Eq. 30})$$

Amb aquestes equacions, ja tenim totes les variables definides per poder determinar en qualsevol moment la velocitat i posició del robot. S'ha de tenir en compte que per determinar la velocitat lineal de cada roda depenem del radi d'aquesta.

## 5 PROVES DE FUNCIONAMENT AMB EL MICROCONTROLADOR ORIGINAL

Per tal d'obtenir alguns patrons de funcionament, s'ha procedit a realitzar alguns assajos amb la placa de control original del robot.

Aquesta placa disposa d'un sistema de test que ens permet verificar que tant els motors com els sensors estan en les condicions necessàries per posar en funcionament el robot. D'aquesta manera hem pogut analitzar els senyals necessaris per posar en funcionament el robot i resoldre algunes de les incògnites del funcionament dels dispositius que hem de controlar per fer funcionar el robot.

### 5.1 Motors

Per tal d'esbrinar els pins que siguin necessaris pel funcionament del control dels motors, s'ha procedit a realitzar el cicle de test amb la placa de control original mentre es visualitza amb un oscil·loscopi els senyals dels pins.

Pin	Descripció
1	Bit de sentit de gir esquerre
2	Senyal motor esquerre
3	Bit de sentit de gir dret
4	Senyal motor dret
17, 18 i 19	Massa
22 i 23	Encoder motor esquerre
24 i 25	Encoder motor dret

Taula 2. Pins dels motors

El sistema de control del sentit de gir dels motors es realitza amb els senyals de dos pins que es correspon amb la següent taula. Com que els dos motors estan situats en el mateix eix, però un en sentit contrari de l'altre, per aconseguir que el robot es mogui endavant o endarrere, un motor girarà en sentit contrari respecte l'altre.

Pin 1	Pin 3	Motor dret	Motor esquerre
0	0	Endarrere	Endavant
0	1	Endavant	Endavant
1	0	Endarrere	Endarrere
1	1	Endavant	Endarrere

Taula 3. Sentits de gir

## 5.2 Sensors d'ultrasò

A la placa dels ultrasons disposen d'un connector de 16 pins on hi arriben els senyals de cadascun dels sensors, en aquesta s'hi realitzen una sèrie de processos desconeguts i després, des de un altre connector de 10 pins, surten els senyals per comunicar amb la placa de control.



Figura 19. Placa de recepció dels sensors d'ultrasò.

Per tal d'esbrinar com es gestionen els pins de sortida del connector s'ha procedit a fer una adquisició de cada un dels senyals. Amb les dades obtingudes hem pogut emplenar la següent taula.

Sensor	Pins del connector de sortida									
	1	2	3	4	5	6	7	8	9	10
0	Massa	5 V	Varia	0 V	5 V	Varia	0 V	5 V	Varia	0 V
1	Massa	5 V	Varia	0 V	0 V	Varia	0 V	5 V	Varia	0 V
2	Massa	5 V	Varia	0 V	5 V	Varia	0 V	5 V	Varia	5 V
3	Massa	5 V	Varia	0 V	0 V	Varia	0 V	5 V	Varia	5 V
4	Massa	5 V	Varia	5 V	5 V	Varia	0 V	5 V	Varia	0 V
5	Massa	5 V	Varia	5 V	0 V	Varia	0 V	5 V	Varia	0 V
6	Massa	5 V	Varia	5 V	5 V	Varia	0 V	5 V	Varia	5 V
7	Massa	5 V	Varia	5 V	0 V	Varia	0 V	5 V	Varia	5 V

Taula 4. Pins sensor d'ultrasò

Com es pot observar, hem trobat pins que es mantenen constants durant tot el procés i d'altres que han variat. Els pins que s'han mantingut a 0 o a 5 V, són els senyals que ens determinaran la lògica de control per la selecció del sensor a utilitzar.

Sensor	Pins					
	4	10	5	4	10	5
0	0 V	0 V	5 V	0	0	1
1	0 V	0 V	0 V	0	0	0
2	0 V	5 V	5 V	0	1	1
3	0 V	5 V	0 V	0	1	0
4	5 V	0 V	5 V	1	0	1
5	5 V	0 V	0 V	1	0	0
6	5 V	5 V	5 V	1	1	1
7	5 V	5 V	0 V	1	1	0

Taula 5. Bits de control

Els pins 3,6 i 9, no es mantenen constants en cada un dels estats, per tant, es pot deduir que un d'aquests pins és l'emissor dels senyals pels sensors i un altre ha de ser el receptor.

A la figura 20, podem observar com el senyal del pin 3, és un senyal amb una freqüència de 25Hz (freqüència del senyal que ens indiquen les especificacions tècniques del Pioneer) i el senyal del pin 6 conserva la mateixa freqüència d'adquisició, però el temps en actiu del senyal és molt més petit.

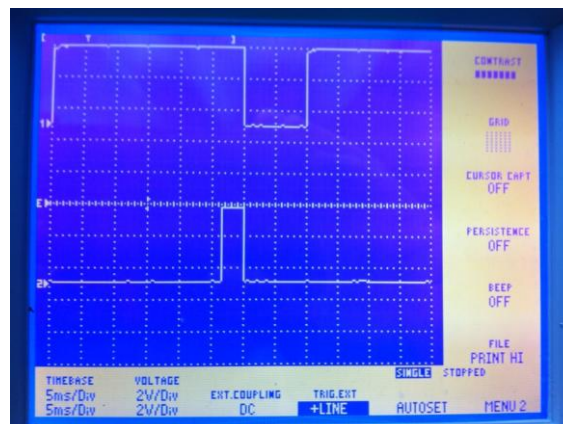


Figura 20. Pin 3 i 6

A la figura 21, podem observar el pin 3 comparat amb el pin 9. Aquests senyals són molt iguals i la freqüència també és de 25Hz, tot i que si sobreposem un senyal a l'altre podem observar que el temps en actiu del senyal 9 és 1000 us més petit que el del pin 3.





Figura 21. Pin 3 i 9

Com podem observar a les figures anteriors, el senyal del terminal 6 és diferent als altres. Si posem un obstacle al davant del sensor actiu, les potes 3 i 9 no varien, en canvi, l'amplitud del senyal del terminal 6 sí, com es pot observar a la figura 22. Quan més prop posem l'obstacle, més gran és el senyal del pin 6. Per tant, el pin 6 és la entrada dels senyals provinents dels sensors.

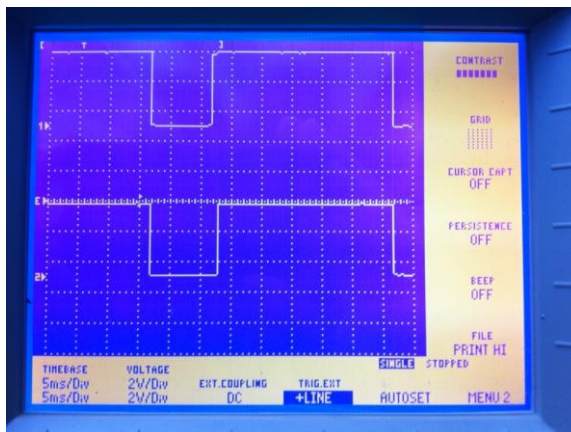


Figura 22. Pin 3 i 6 amb intervenció de distància

Per acabar, dels dos senyals restants, un ha de ser el senyal emissor. Hem observat que si desconnectem el cinturó d'ultrasò, els dos senyals continuen apareixent, per tant, els dos són generats per la placa de control. Si només fem servir el senyal del pin 3, el sensor emet senyals, però en canvi, si només connectem el senyal del pin 9, no emeten cap senyal.

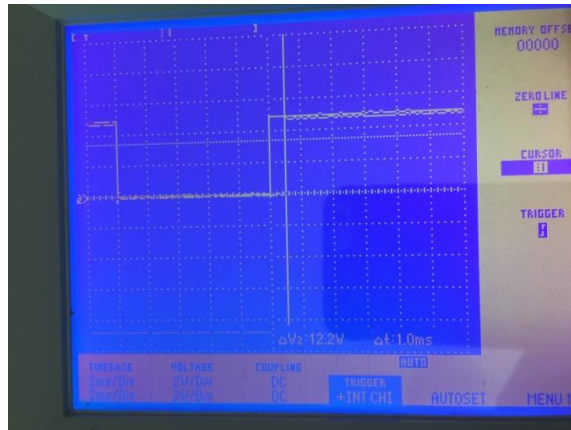


Figura 23. Diferència d'amplitud Pin3 i Pin9

Per altre banda, hem pogut observar que si només fem servir el senyal del pin 3, a l'hora d'acostar un obstacle, el senyal que rebem no es correspon. Com hem dit abans, l'amplitud del senyal del pin receptor dels ultrasons augmenta en el moment en que anem acostant un obstacle, en canvi, si no enviem també el senyal del pin 9, no segueix aquesta consigna, ja que realitza canvis d'amplitud sense sentit.

D'aquesta manera podem deduir que el pin 3 ens genera el senyal que emeten els sensors i que el pin 6 és el que ens proporciona la resposta dels sensors amb l'amplitud a processar i que ens determinarà la distància a l'obstacle. Per altre banda, hem arribat a la conclusió que el senyal del pin 9 es fa servir com a referència a la placa de guany dels ultrasons i que ens sincronitza els senyals fent coincidir els intervals de temps.

### 5.2.1 Conclusions de l'anàlisi

Amb aquest estudi, s'ha determinat el model de connexió a utilitzar, al igual que les característiques del senyal enviat als sensors. Aquest senyal té una amplitud de 0,03s i la freqüència de treball del sensor és 25Hz (període de 0,04s) com indica també a les especificacions del manual d'usuari del robot.

## 6 DISSENY DE LA FPGA.

Dins l'arxiu "myRIO-1900 Customized FPGA" situat al nivell de programació més baix del robot, s'ha dissenyat la assignació de les entrades i sortides necessàries i les operacions per adaptar els senyals a les unitats desitjades.

A continuació es pot observar la relació de connexió entre el connector de la placa de control dels motors i el MyRio:

Connector dels motors		MyRio
1	Senyal motor dret	ConnectorC/DIO3
2	Bit de sentit de gir dret	ConnectorC/DIO6
3	Senyal motor esquerre	ConnectorC/DIO7
4	Bit de sentit de gir esquerre	ConnectorC/DIO5
17, 18 i 19	Massa	DGND
22	Encoder motor esquerre A	ConnectorB/DIO11
23	Encoder motor esquerre B	ConnectorB/DIO12
24	Encoder motor dret A	ConnectorA/DIO11
25	Encoder motor dret B	ConnectorA/DIO12

Taula 6. Connector dels motors amb MyRio

La següent taula mostra la relació de connexió amb el cinturó d'ultrasons:

Connector del cinturó d'ultrasons		MyRio
1	Massa	DGND
2	5 volts	5 V
3	Emissor del senyal del sensor	ConnectorC/DIO0
4	Selecció del sensor	ConnectorB/DIO14
5	Selecció del sensor	ConnectorC/DIO4
6	Adquisició	ConnectorC/DIO1
7	Massa	DGND
8	5 volts	5 V
9	Senyal de referència	ConnectorB/DIO0
10	Selecció del sensor	ConnectorC/DIO2

Taula 7. Connector dels ultrasons amb MyRio

### 6.1.1 Motors

Per controlar els motors s'han utilitzat dos blocs. El primer, assigna la sortida que farem servir per enviar el senyal a la placa de control dels dos motors. Per controlar-los, es farà

servir un PWM al que l'hi variarem l'amplada per tal de regular la velocitat. S'ha escollit una freqüència de 1000Hz.

Aquest loop, també disposa de dues entrades digitals, una per cada motor que serveixen per canviar de sentit de gir. S'ha relacionat la posició del senyal de sentit de gir amb el signe del control del senyal de la velocitat, com es mostra a la figura 27.

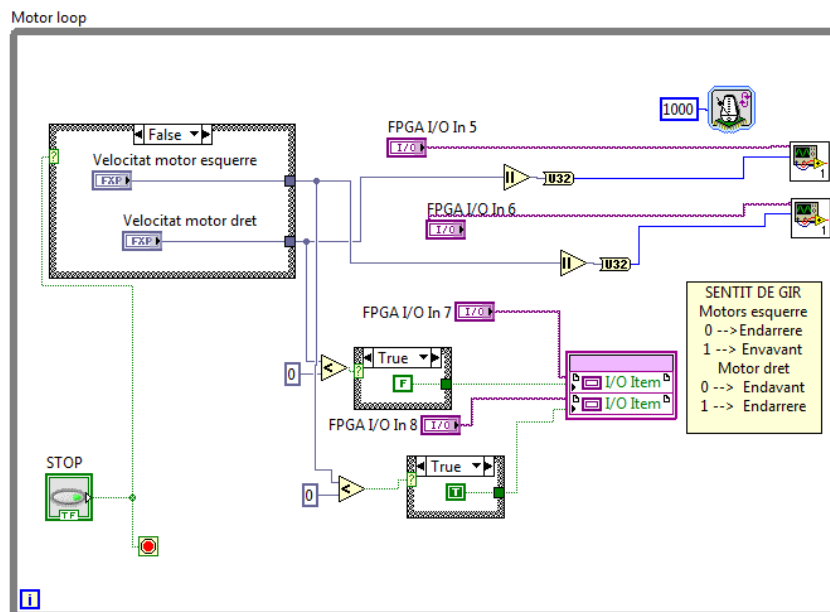


Figura 24. Generació dels senyals motors

El segon loop, conté un bloc PID predeterminat del Labview, hem vist que aquest controlador compleix amb les especificacions bàsiques per ser utilitzat per regular la velocitat del motor. Hem ajustat aquest empíricament fins que la resposta ha sigut la desitjada, sense sobrepic i prou ràpida. S'ha limitat l'entrada a la velocitat màxima a la que pot girar el motor, uns 170ticks/count. A aquest bloc, per tal de no provocar canvis bruscos de velocitat, també s'ha incorporat una limitació per la variació del valor de consigna que pot entrar al PID, veure figura??. S'ha limitat la velocitat a 110 ticks/count, que són uns 1,45 m/s una mica menys de la velocitat màxima a la que es pot desplaçar el robot, d'aquesta manera hi ha un marge per assegurar que en cap cas es sobrepassa aquesta velocitat. També s'ha limitat l'acceleració del robot a 0,5 ticks per cycle, és a dir, cada 100 microsegons, o dit, d'una altre manera una acceleració de 6,57 cm/s. A la figura 25 es poden observar les limitacions de velocitat i acceleració i com a continuació els valors entren al PID. Es pot veure el bloc a la figura 25.

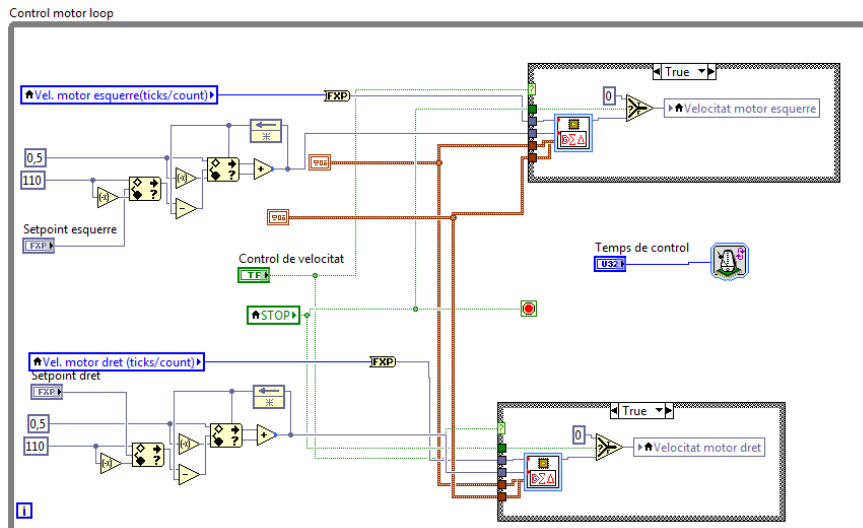


Figura 25. Control dels motors

### 6.1.2 Encoders

En aquest apartat, s'ha dissenyat el tractament de l'encoder de quadratura x4. S'ha fet servir un "Timed loop" ja que necessitem precisió i rapidesa amb el temps. Com que les entrades dels encoders com a màxim accepten fins a 100KHz, necessitem una freqüència igual o major a aquesta si volem que el comptador no es deixi cap senyal. Per altre banda, hem especificat un temps de mostreig, és a dir, l'encoder anirà sumant el número de polsos que vagi rebent fins que la diferència de temps entre els Tick Count sigui la que s'ha establert.

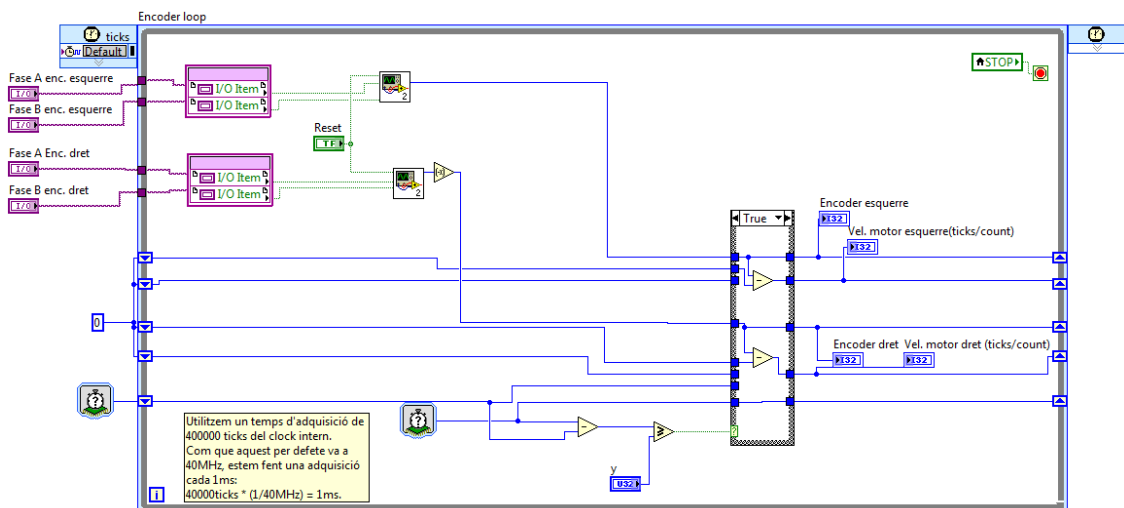


Figura 26. Encoder

Per complementar l'encoder s'han realitzat les operacions necessàries per realitzar la conversió d'unitats. La sortida dels encoders ens indica ticks/count. Per realitzar proves

amb el sistema FPGA, hem volgut treure els rad/s i els cm/s com a referència visual i així poder tenir una millor percepció de les velocitats.

$$\text{Velocitat}_{\text{rad/s}} = \text{Velocitat}_{\text{ticks/s}} \cdot \frac{2\pi}{n^{\circ} \text{ ticks}} \quad (\text{Eq. 31})$$

Un cop es té el valor en radiants per segon, només s'ha de multiplicar per el radi de la roda per obtenir la velocitat lineal de cada roda en centímetres per segon.

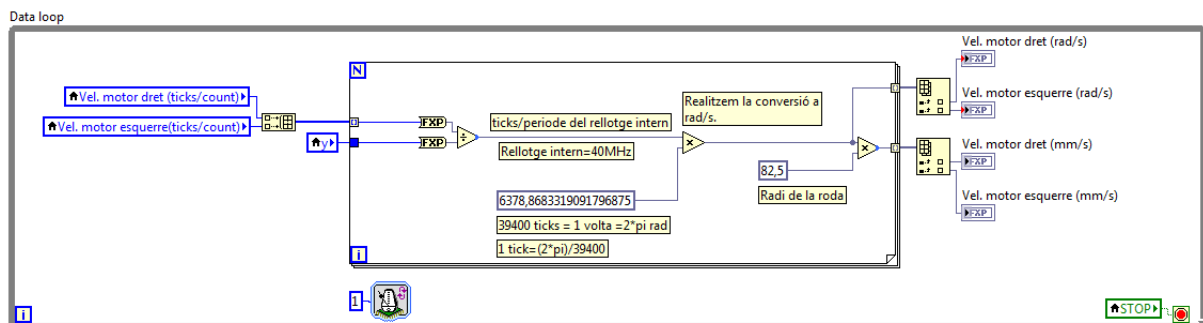


Figura 27. Conversions cm/s i rad/s

### 6.1.3 Ultrasons

S'ha tractat el sensor amb dos blocs independents. Per una banda l'emissió dels senyals necessaris i per l'altre l'adquisició i tractament per convertir a distància.

Dins el loop d'emissió dels senyals, trobem una estructura que primer selecciona la sonda a utilitzar per després generar els senyals necessaris. La freqüència d'aquests senyals és de 25 Hz i l'amplitud de 30 milsegons, com indiquen les especificacions del manual d'usuari del robot. Per realitzar aquest loop. També disposem de tres senyals digitals que determinen quin dels sensors està actiu en cada moment.

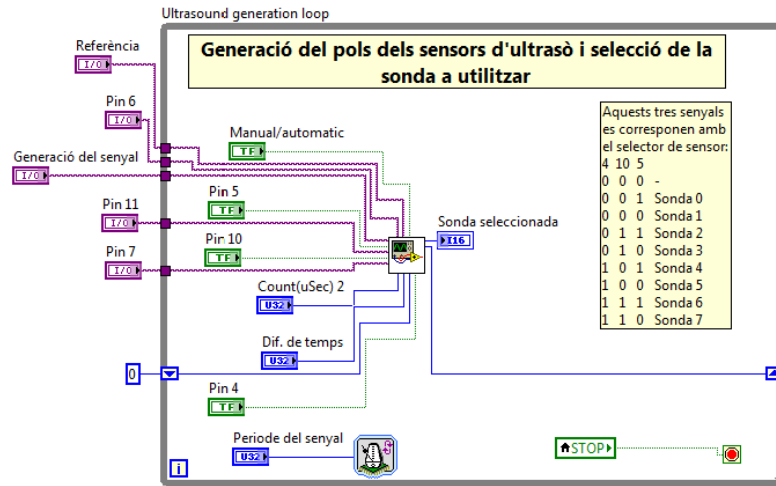


Figura 28. Generació ultrasons

Per altre banda, el loop d'adquisició dels senyals ens mesura la diferència de temps entre l'amplitud del senyal de sortida amb el senyal d'entrada. Aquest és multiplicat per la velocitat del so a 20°C, uns 343m/s, i obtenim la distància d'anada i tornada entre el robot i l'obstacle. A continuació, es divideix per dos per obtenir la distància a l'obstacle. Si el senyal d'emissió s'ha perdut o no ha rebotat en cap obstacle per donar una lectura, s'ha ajustat un temps màxim d'espera i si es supera, ens indicarà el valor màxim de distància.

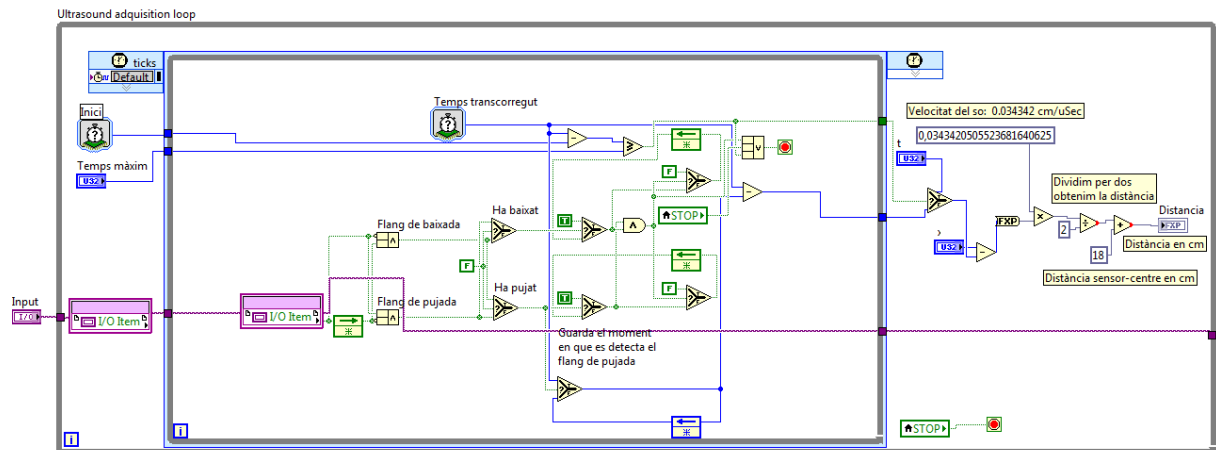


Figura 29. Adquisició ultrasons

## 7 ARQUITECTURA MULTIAGENT

La arquitectura de control original del robot anomenada ARMADiCO (Autonomous Robot Multi-Agent Architecture with Distributed Coordination) és un sistema de control que combina la generació d'una resposta a la percepció dels canvis que es produeixen a l'entorn amb el disseny dels moviments que realitza el robot en funció d'aquesta resposta. Els agents que componen aquest control el formen cinc grups que podem observar a la següent imatge extreta de la tesis de la Dra. Bianca Innocenti.

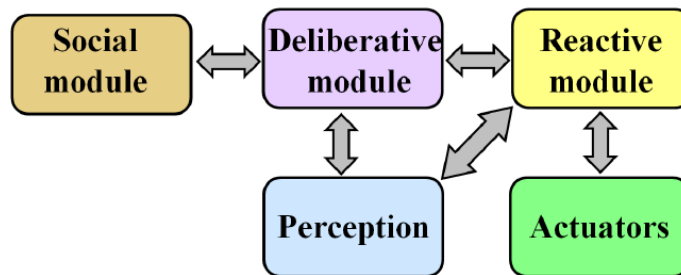


Figura 30. Tipus d'agents

Com podem veure a la figura, aquests grups són els agents socials, els deliberatius, els reactius, els de percepció i els agents d'acció.

Els agents socials són els que s'encarreguen de transmetre les dades introduïdes per l'usuari i dona l'ordre als altres agents de que s'executin. A continuació trobem els agents deliberatius són els que fan possible que el robot realitzi tasques com el càlcul de trajectòries, determina en quin punt del mapa es troba el robot o decideix quan el robot s'ha de dirigir a la plataforma de càrrega. Els agents de percepció s'encarreguen d'adquirir les dades que provenen dels diversos sensors. Els agents reactius que generen la resposta als canvis de l'entorn i els agents d'acció que fan possible que el robot es mogui.

Per tal de dissenyar el nou programa ens hem basat en els principals agents que fan possible la percepció i la resposta a l'entorn. Per tant, hem reduït el numero d'agents i alguns els hem modificat per adaptar-los al funcionament del MyRio. S'ha centrat l'atenció sobre els agents reactius, els agents de percepció i els d'acció. També ha d'existir un agent social que faci possible la comunicació amb l'agent humà.

Amb la interfície d'usuari s'enviaran les coordenades i donarà permís al MyRio per activar els agents que conté. A continuació, l'agent dels ultrasons i el de l'encoder llegiran els



senyals dels sensors que provenen de la FPGA per tractar-les i comunicar-se amb els agents reactius Goto, Avoid i Gothrough. Cada un d'aquests agents generarà una resposta que depenent del pes que l'agent de selecció i aplicació els hi atribueixi i s'enviarà els valors de velocitat de l'agent seleccionat cap a la FPGA per generar els senyals de control dels motors

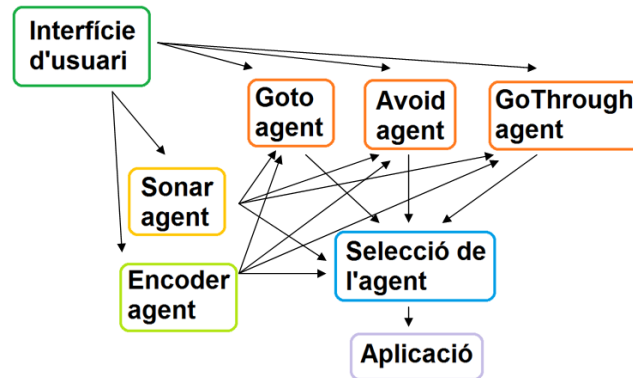


Figura 31. Comunicació dels agents

## 7.1 Programació del sistema de control

Dins l'arxiu "Programa\_RT", s'ha dissenyat el programa de control que conté els agents.

Abans d'executar els agents, el programa ha de inicialitzar el model FPGA que hem programat anteriorment i ha d'activar els sensors. La comunicació entre la FPGA i el programa Real Time la realitza el dispositiu sense que nosaltres haguem de realitzar cap programació específica per la comunicació. També s'ha realitzat un cicle que no ens executa els agents fins que no s'han rebut de la interfície d'usuari les coordenades de posició inicials i finals.

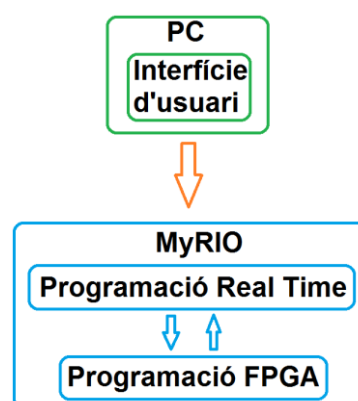


Figura 32. Nivells de programació

### 7.1.1 Ultrasound agent

Al activar el programa, aquest agent crea una taula on cada posició contindrà un angle d'incidència corresponent al de cada sensor i la distància. Al començament, aquesta distància serà infinit, ja que així no provocarà resultats erronis al comparar les distàncies per primer cop.

A continuació, es llegirà el numero que s'ha assignat a cada sensor a la FPGA i sobreescrirà el valor de distància que conté el registre en qüestió. cada senyal li assigna l'angle d'orientació depenent del sensor que s'estigui utilitzant en cada moment.

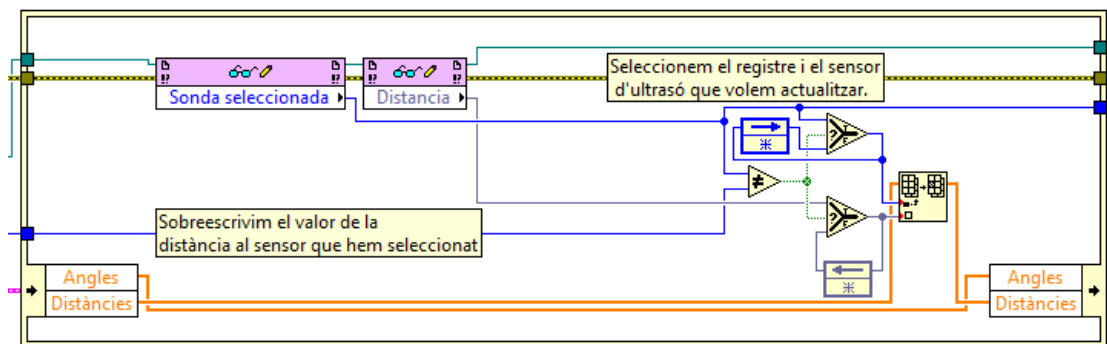


Figura 33. Escriure distàncies

L'adquisició de les dades d'aquest procés és el més lent, ja que la freqüència de funcionament dels sensors es de 25Hz, per tant, la resposta del sistema està limitada per aquest valor.

### 7.1.2 Encoder agent

Aquest agent, serveix per situar el robot en el sistema de coordenades i ens indica la posició del robot a cada cicle de mostreig.

Per executar aquest agent, li arriben les dades de la posició inicial del robot que li enviem des de la interfície d'usuari. A continuació, adquireix la velocitat de cada una de les rodes i realitza una sèrie d'operacions aplicant les fórmules del moviment cinemàtic estudiades en apartats anteriors trobarà la velocitat lineal i angular al centre del robot. Aquesta es multiplica per el temps de mostreig i s'obté la distància avançada per cada cicle. Tot seguit, apliquem les fórmules trigonomètriques estudiades per representar la posició del robot en els eixos de coordenades X i Y.

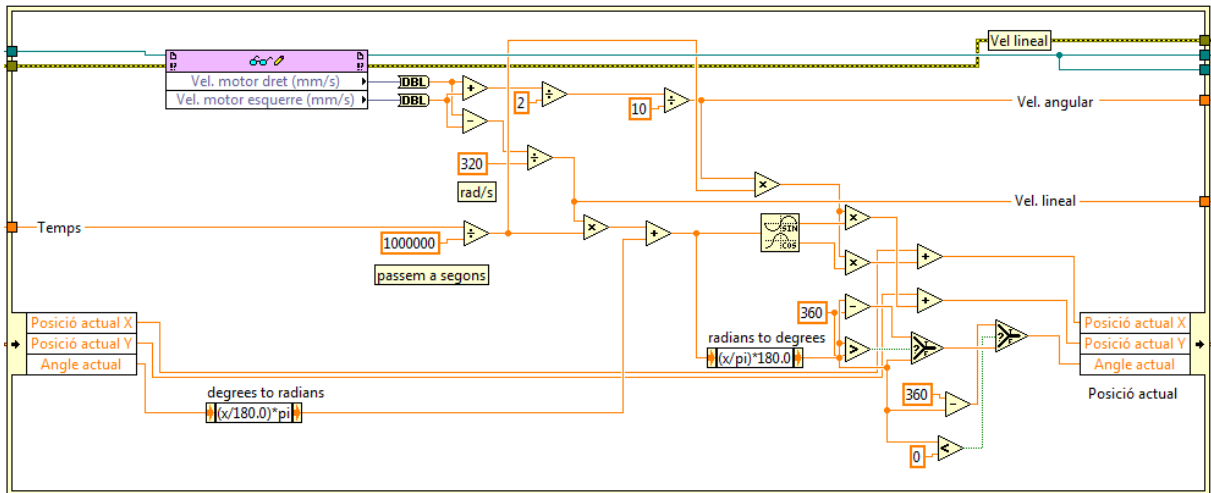


Figura 34. Posició actual

### 7.1.3 Goto agent

Quan li arriba la posició actual i la posició final, busca la distància restant per arribar al punt destí. D'aquesta manera pot treure una distància en X i en Y i un angle de gir. Quan té aquesta informació, l'aplica al seus controls de posició lineal i angular. Posteriorment els resultats d'aquest control determinaran les ordres per donar els valors al control de velocitat de cada roda.

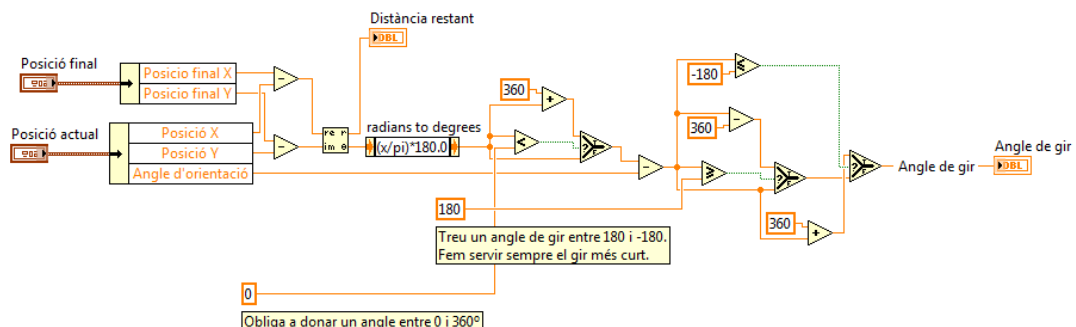


Figura 35. Distància restant

Aquest agent ens divideix la distància per arribar al destí en tres zones on ens ofereix una velocitat màxima diferent. Com més prop sigui el punt final, menor serà la velocitat màxima del robot. També s'ha establert que si l'angle de gir inicial és més gran a 45°, el robot realitzarà un gir sobre ell mateix per reduir aquest angle i així suavitzar la curvatura del seu recorregut.

Aquest agent disposa d'un control de posició per tal de relacionar la posició actual i final del robot amb la velocitat que haurà d'aplicar. Per fer-ho s'ha dissenyat un PID amb la següent expressió.

$$u(k) = u(k-1) + e(k) \cdot \left[ K_p + K_i \cdot \Delta T + \frac{K_D}{\Delta T} \right] + e(k-1) \cdot \left[ K_p - 2 \cdot \frac{K_D}{\Delta T} \right] + e(k-2) \cdot \frac{K_D}{\Delta T} \quad (\text{Eq. 32})$$

On  $u(k)$  és la sortida,  $u(k-1)$  el valor anterior,  $e(k)$  és l'error,  $e(k-1)$  l'error anterior,  $e(k-2)$ , l'error de fa dues execucions,  $K_p$  la constant de proporcionalitat,  $K_i$  la constant d'integració,  $K_D$  la de derivació i  $\Delta T$  el temps de cada execució. S'han ajustat els valors d'aquest control segons el mètode de prova i error, aquests són 0.5, 0.2 i 0.2 respectivament.

Disposem de dos controls per cada agent, un per la velocitat lineal, on el valor actual serà la distància en que es troba el robot en cada moment i la consigna la distància al punt de destí, i un control per la velocitat angular on el valor actual és l'angle del robot i el valor de consigna l'angle que forma la posició final amb l'actual.

En funció de la sortida, s'ha limitat la velocitat màxima i mínima a la velocitat màxima de cada un dels trams en el que es troba el robot. Si la distància restant és més gran de 3 metres, la velocitat màxima serà 50 cm/s, si és inferior però major a 50 cm la velocitat màxima serà 20 cm/s i si és inferior serà 15 cm/s. Si la distància està entre els valors anteriors, el valor de la velocitat serà proporcional a aquesta distància. Amb el control de la posició angular s'ha fet el mateix, però s'ha limitat l'angle màxim a  $20^\circ$  i la velocitat angular si la distància és inferior a 50 cm serà de 1 rad/s i si és més gran 2 rad/s. Si l'angle del control de posició té un valor entre 0 i  $20^\circ$  la velocitat de sortida serà proporcional a aquest valor.

També s'ha introduït una consigna anomenada error màxim que fa possible que les velocitats dels motors es posin a 0 si la distància al destí és inferior a 5 cm, si no disposem d'aquest error, el robot sempre es mourà encara que sigui molt poc.

#### 7.1.4 Avoid agent

En aquest apartat, l'agent, com en el cas anterior, separa el control en la posició angular del robot i la distància als obstacles.

Per realitzar el control, el robot selecciona els dos obstacles més propers a ell i defineix els possibles casos que ens podem trobar. Els dos obstacles dins la zona d'interès, l'obstacle més proper dins la zona d'interès, l'obstacle més proper a davant del robot dins la zona d'interès o cap dels obstacles dins la zona d'interès.

A continuació, en funció del cas en el que es trobi el robot, aquest tractarà els senyals per saber cap a quin costat és més convenient realitzar el moviment de rotació i així poder esquivar els obstacles.

Aquest agent també ens distingeix entre tres zones de funcionament diferents per limitar les velocitats màximes. Així, el robot adequarà la velocitat màxima a la que pot circular en funció de la zona d'acció a la que l'obstacle estigui situat. Com que el robot pot portar diferents velocitats a l'hora d'apropar-se a l'obstacle, s'ha buscat la manera de que aquest tingui prou temps per respondre a la distància restant, d'aquesta manera s'ha acordat que si la distància que el robot pot recorre en 2 segons és més gran que la distancia la zona de stop de 40 cm, aquesta distància passarà a ser la distància recorreguda en 2 segons, i el mateix passarà amb la zona de pas i la zona d'acció, però com que en aquests trams la velocitat màxima pot ser més alta, el robot també necessitarà més temps per respondre i frenar, per tant, s'ha decidit utilitzar per la zona de pas 3 segons i per la d'acció 4. Tot i que amb les velocitats que s'estableixen en aquest agent és improbable que es produeixi aquesta situació, ja que les velocitats no són prou altes, és possible que amb la combinació d'agents es pugui produir.

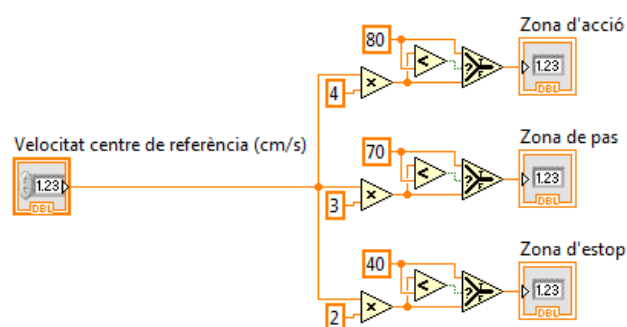


Figura 36. Zones d'acció

Anomenarem les zones com zona d'acció, zona de pas i zona de stop. Si l'obstacle està situat a la zona d'acció, el robot es desplaçarà en línia recta per anar a buscar la zona de pas. Un cop dins, el robot es troba a la zona on ha de variar la seva orientació per evitar l'obstacle, en aquest cas, com que es troba dins la zona d'interès, el canvi d'orientació serà

de 90°. Si no ha realitzat un gir suficientment gran o per alguna altre raó entrem dins la zona de stop, redueix la velocitat lineal i realitza un canvi d'orientació de 180° per sortir d'aquesta zona i així allunyar-se de l'obstacle.

Per acabar, aquest agent disposa d'un control de posició PID com el de l'agent anterior, però en aquest cas, només per controlar la posició angular, les constants són les mateixes, però en aquest cas el valor actual del control és l'angle d'orientació i el valor de consigna serà l'angle de rotació per esquivar l'obstacle. Per la posició lineal, la velocitat s'estableix segons la zona a la que es troba l'obstacle. S'ha limitat la sortida del controlador a 30°, d'aquesta manera amb aquest valor s'obté una velocitat angular màxima de 2 rad/s. Si la orientació és inferior, la velocitat serà proporcional a aquest valor.

#### 7.1.5 Gothrough agent

Aquest agent s'utilitza en les situacions en que el robot es troba entre dos obstacles relativament propers i s'ha de desplaçar entre ells. Per exemple un passadís o una porta.

Per realitzar aquesta acció, l'agent busca els senyals dels dos sensors laterals del robot i amb les distàncies als obstacles busca en cada moment el punt mig entre els dos que ens definirà una línia de pas. A continuació reflexa la posició final sobre la línia de pas i la divideix per dos, amb aquest nou punt com a posició objectiu, busca l'angle d'orientació i la distància restant. Si la distància al punt final és inferior a 50 cm, l'angle d'orientació de referència passarà a ser la del punt final.

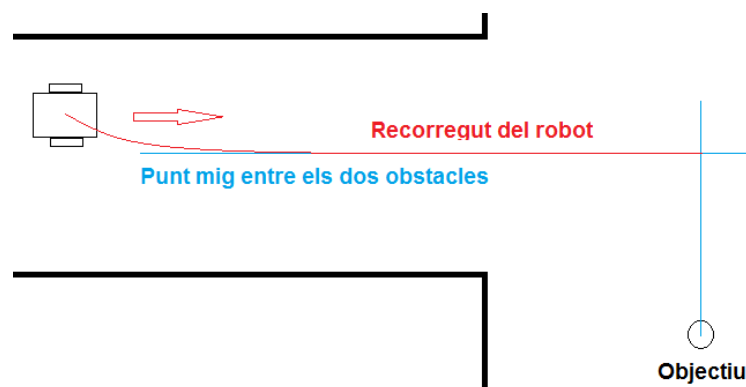


Figura 37. Funció Gothrough

Els controladors d'aquest agent són iguals que els de l'agent Go to, amb la diferència de que com a punt de consigna fa servir la distància al punt calculat. S'ha limitat la sortida del PID

de posició a 40 cm i la angular a 45° d'aquesta manera quan els controladors tinguin aquest valor la velocitat lineal serà de 40 cm/s i la angular de 1 rad/s. Si la distància és més petita, la velocitat que s'obtéindrà serà proporcional a aquest valor.

### 7.1.6 Selecció i aplicació dels senyals

Aquest apartat de la programació assigna un pes a cada agent en funció de les variables que el condicionen per així seleccionar l'agent Goto, Avoid o Gothrough i poder arribar a la posició destí establerta.

Per realitzar el càlcul dels pesos dels agents farem servir l'equació que trobem a continuació:

$$u = \begin{cases} p_{\max} & x > u_L \\ \alpha \cdot x + \beta & u_L \leq x \leq u_H \\ p_{\min} & u_H \end{cases} \quad (\text{Eq. 33})$$

$$\alpha = \frac{p_{\max} - p_{\min}}{u_L - u_H} \quad (\text{Eq. 34})$$

$$\beta = \frac{u_L \cdot p_{\min} + u_H \cdot p_{\max}}{u_L - u_H} \quad (\text{Eq. 35})$$

On  $u_L$  és el valor mínim de la variable feta servir per cada agent,  $u_H$  el valor màxim,  $p_{\max}$  el pes màxim i  $p_{\min}$  el pes mínim.

L'agent Goto farà servir la distància restant per arribar a la posició destí com a variable de control. El valor de distància màxima serà 20 i la mínima 50 centímetres. Aquest agent ha de tenir un pes mínim major que els altres agents si volem que tingui prioritat sobre els altres, per això el pes mínim serà 0,6 i el pes màxim serà 1, com es pot veure a la figura següent.

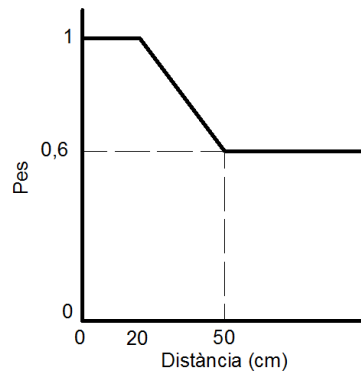


Figura 38. Pes Goto

L'agent Gothrough es basarà amb la distància mitja entre els obstacles dels dos costats del robot. El valor mínim de distància és 40 i el màxim 120 centímetres. El pes màxim i mínim serà 1 i 0. A continuació trobem la representació de l'equació d'aquest agent.

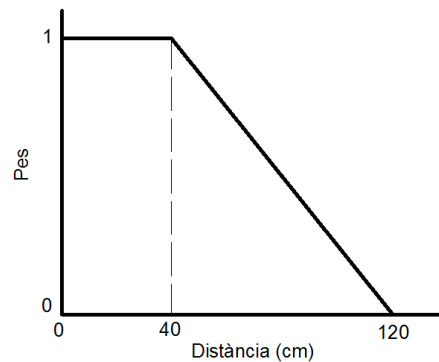


Figura 39. Pes Gothrough

L'agent Avoid farà servir dues equacions. La primera utilitza la distància a l'obstacle més proper al robot i la segona l'angle d'incidència. A continuació realitzarà una comparació dels dos valors i seleccionarà el valor més petit. El pes que donarem a cada agent anirà de 0 a 1. Les distàncies màxima serà el valor de la zona de pas i el mínim el de la de stop. L'angle màxim serà 110° i el mínim 40°.

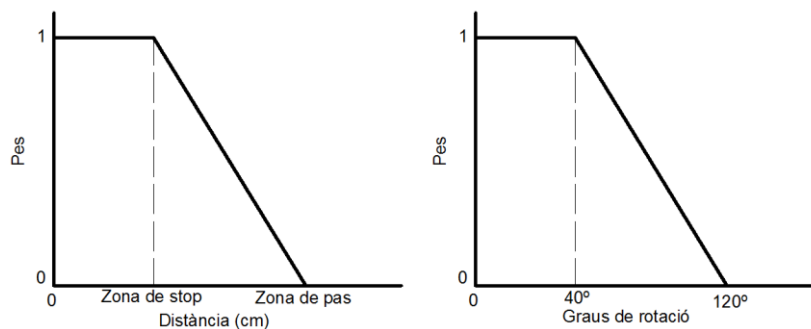


Figura 40. Pes Avoid



A continuació es pot veure la part del programa que assigna aquests pesos. Com es pot observar, l'agent Goto i Gothrough disposen del seu bloc per calcular el pes, mentre que l'Avoid en té dos, un per l'angle i un per la distància. A continuació, s'introdueixen aquests valors en un array que selecciona el valor més gran i treu un valor de 0 a 2 que servirà per seleccionar les velocitats de l'agent amb el major pes.

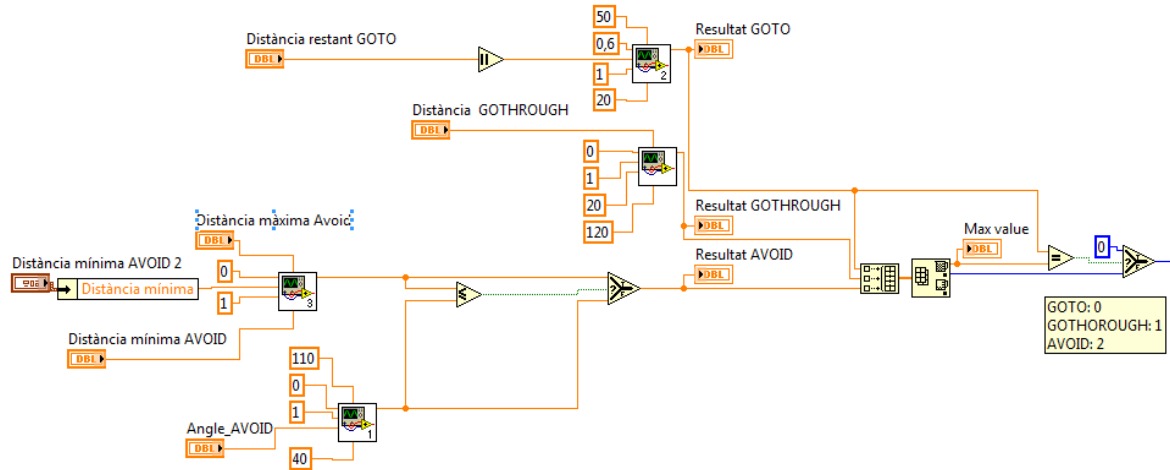


Figura 41. Selecció de l'agent

A continuació es limita la variació de velocitat que hi pot haver d'una execució a una altra per tal de evitar els canvis sobtats de velocitat al passar d'un controlador a un altre. Assignem la constant de 0,005. Significa que a cada execució podrà variar la velocitat un màxim de 5 cm/s, ja que el cicle s'executa cada mil·l·segon. Seguidament, realitzem les operacions inverses per passar de cm/s a ticks/count

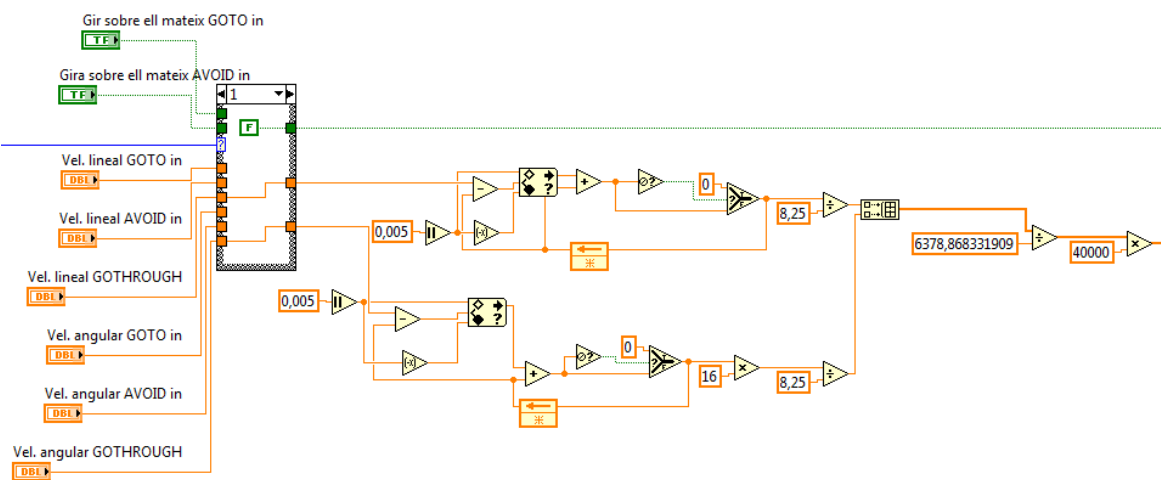


Figura 42. Límits de variació i conversió.

Per acanar, dividim la velocitat en dues, per tal de enviar els senyals a cada motor. Per fer-ho s'ha hagut de invertir els sentits de gir dels motors segons la velocitat angular del robot i també s'han contemplat els casos en que el robot gira sobre ell mateix. Es sumen els valors que provenen de la velocitat lineal i angular de cada motor i s'envien a la FPGA.

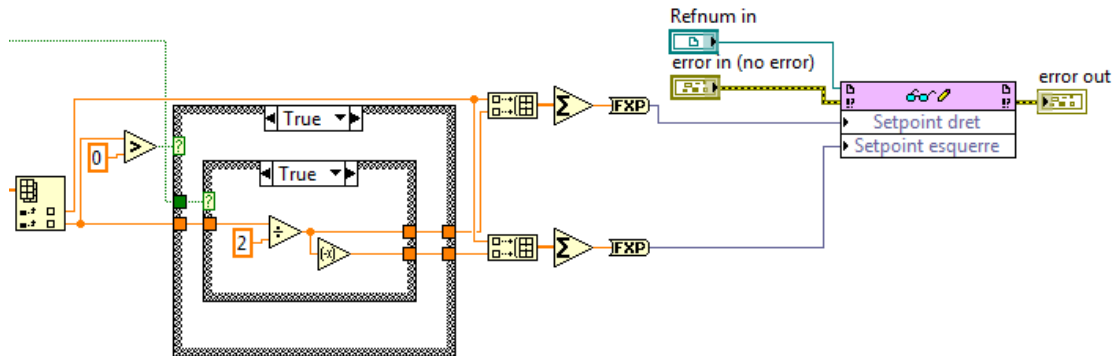


Figura 43. Sentit de gir i velocitats a la FPGA

### 7.1.7 Interfície d'usuari

El programa s'executa des de l'ordinador de l'usuari i envia i sobreescriu les variables de posició inicial i final situades al MyRIO. Un cop introduïdes aquestes variables s'activarà el botó d'acció que donarà l'ordre al dispositiu remot de que pot continuar amb el cicle del programa. Aquestes variables s'anomenen "Shared Variables" i són variables que comparteix el PC amb el dispositiu.

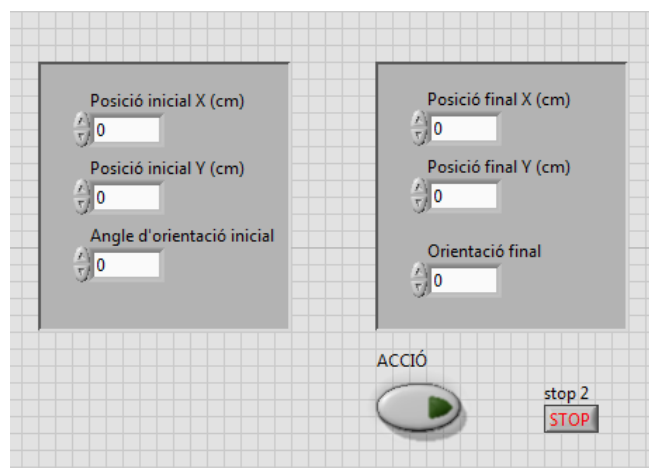


Figura 44. Interfície

### 7.1.8 Visualització

Per acabar, s'han representat els obstacles i el recorregut del robot sobre el sistema de coordenades per poder visualitzar els resultats obtinguts a les proves. Abans de fer-ho s'han adaptat els senyals dels ultrasons perquè es situïn sobre el sistema de referència en funció de la posició del robot. D'aquesta manera es podrà veure com apareixen i desapareixen els obstacles de l'entorn.

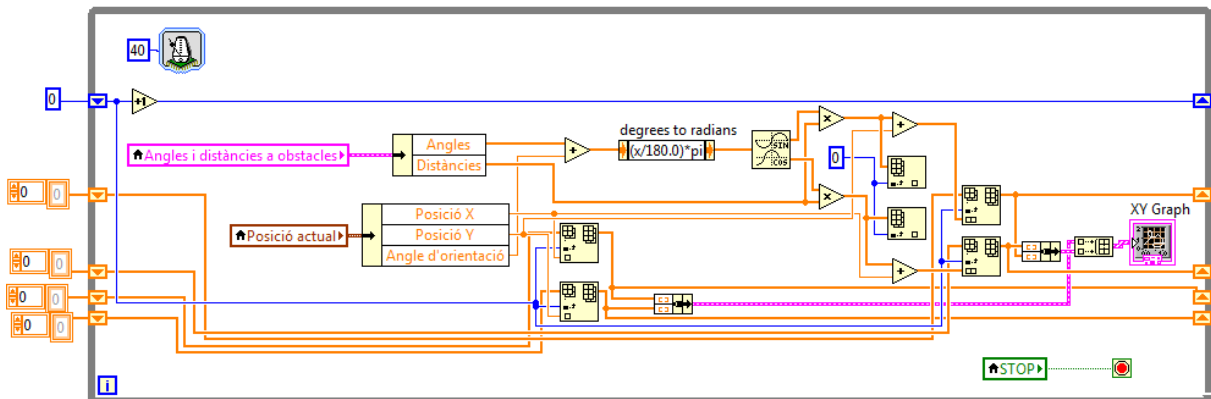


Figura 45. Visualització

## 8 PROVES DE FUNCIONAMENT

### 8.1 Paràmetres de comunicació

Abans de posar en marxa el robot, s'ha hagut d'establir la connexió entre l'ordinador i el dispositiu. Normalment, el dispositiu t'ofereix la connexió via USB de manera predeterminada i es comunica via la direcció IP que té assignada. En el nostre cas, la direcció IP del dispositiu és la 172.22.11.2, per tant, s'ha canviat aquesta connexió per una connexió Wi-fi mitjançant un router.

Per realitzar aquesta connexió, primer s'ha connectat el dispositiu via USB i accedit a les seva configuració, li hem afegit la xarxa Wi-fi seleccionada. Un cop guardats els canvis, podem veure la IP per la connexió "wireless" del dispositiu. En el nostre cas la 192.168.1.102.

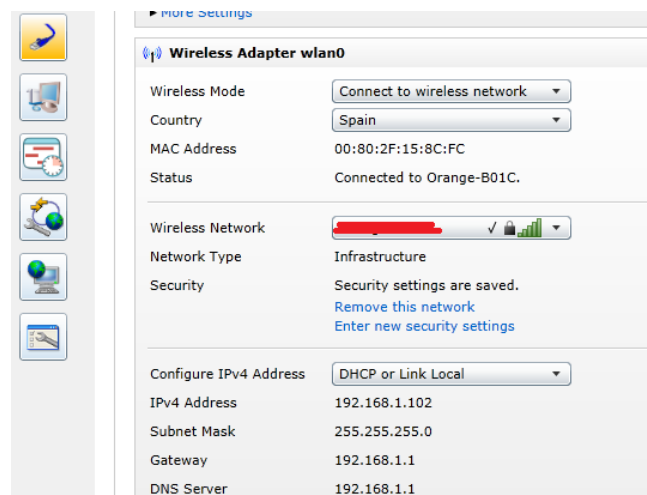


Figura 46. Xarxa i IP

### 8.2 Resultats obtinguts

Amb l'objectiu de comprovar el funcionament de la programació realitzada s'han realitzat una sèrie de proves per tal de analitzar el funcionament dels diferents agents.

#### 8.2.1 Mapa Go to

Les primeres proves que hem realitzat, les hem executat en una habitació lliure d'obstacles per tal de veure com es comporta el robot quan l'agent Goto és el que té el control.

En el primer cas, el robot ha d'anar d'un punt amb coordenades [0,0] i una orientació de  $90^\circ$  fins a un punt amb coordenades [-50, 200] i un angle final de  $0^\circ$ .

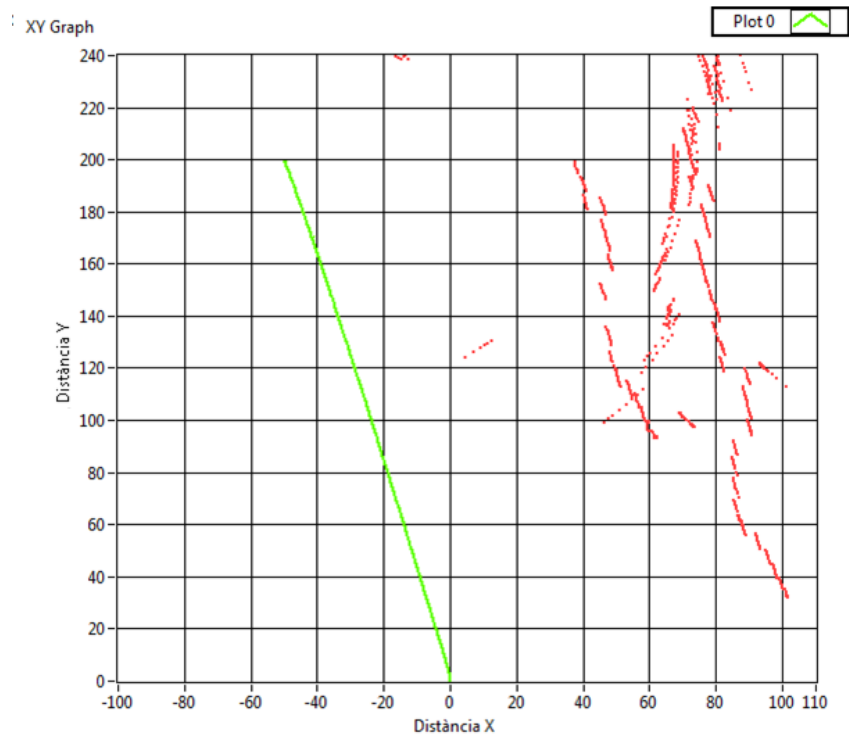


Figura 47. Prova 1

Com podem observar el robot realitza un petit ajust d'orientació per arribar a l'angle d'orientació desitjat i així arribar al punt final. Un cop en aquest punt, realitza una rotació més lenta per canviar l'angle de gir a  $0^\circ$ . A la figura següent es pot observar l'angle de gir.

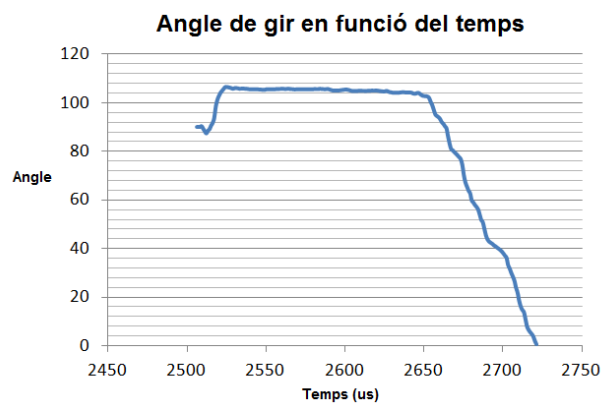


Figura 48. Angle de gir, prova 1.

En el següent cas, realitzem la mateixa operació però haurà de realitzar una rotació major per arribar a la posició destí. Les coordenades inicials són [30,15] amb una orientació inicial de  $90^\circ$ , i la posició final serà [80,80] i un angle d'orientació de  $0^\circ$ .

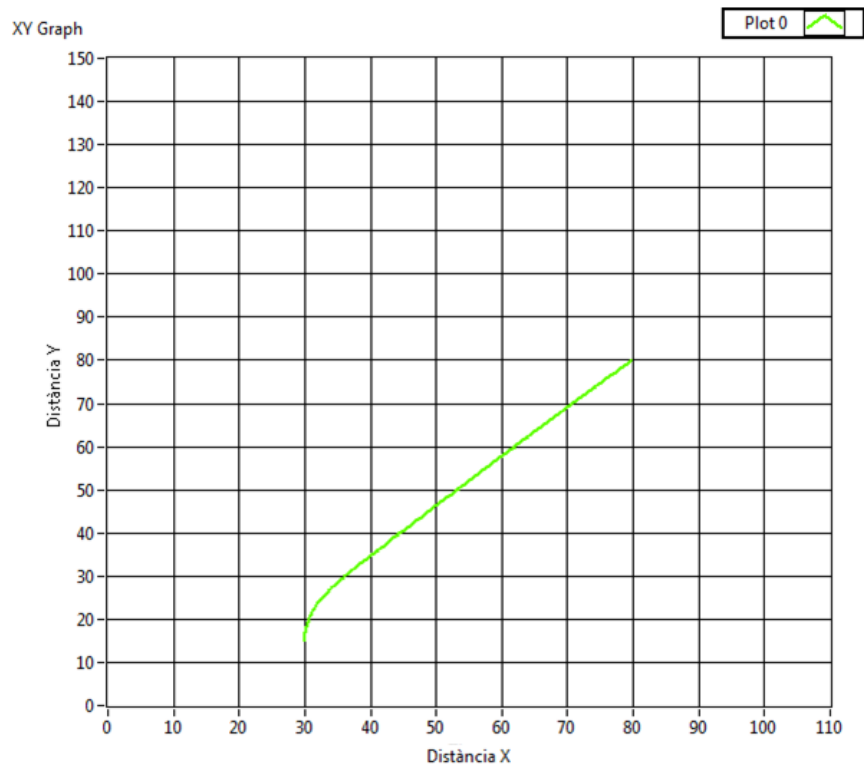


Figura 49. Prova 2

Podem observar com a la figura anterior, la curvatura del camí seguit per el robot és major que a la prova 1, ja que l'angle de gir és més gran. Com que aquest és major a 45° el primer moviment que farà serà girar sobre el seu propi eix. Quan sigui inferior, es començarà a desplaçar cap a l'objectiu. A la figura 50 es pot observar l'angle de gir. La posició final del robot varia una mica dels 0°, ja que s'ha donat una petita tolerància per tal de que el robot no estigui sempre en moviment, encara que es mogui molt poc.

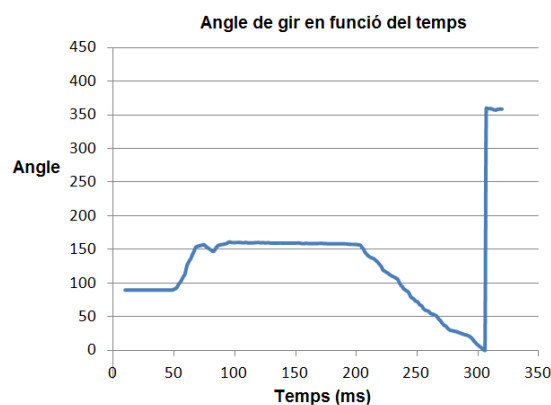


Figura 50. Angle de gir, prova 2.

A continuació s'ha realitzat una prova en una habitació sense obstacles una mica més gran, de tal manera que el robot ha pogut agafar més velocitat. Podem observar com el robot

s'encara cap al punt objectiu i quan hi arriba realitza una sèrie de moviment per ajustar-se al punt i orientació final desitjada amb un cert error que hem permès.

Les coordenades inicials són [30, 30] amb una orientació de  $90^\circ$  i la posició final és [280, 200] amb una orientació de  $-90^\circ$ .

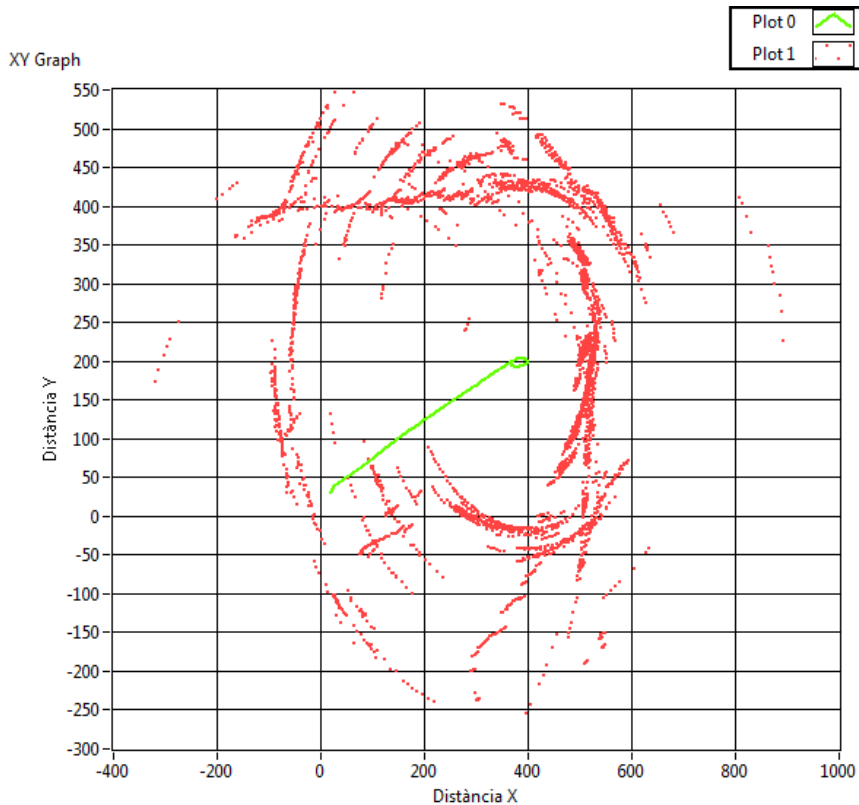


Figura 51. Prova 3

### 8.2.2 Mapa Go to i Avoid

En el següent cas d'estudi veurem com es complementen els agents de control Avoid i Goto per acabar arribant a la posició de destí esquivant un obstacle. L'escenari en el que s'ha realitzat l'assaig és una habitació que disposa d'una columna al costat d'una de les parets. Les coordenades de posició inicials són [0,0] amb una orientació de  $90^\circ$  i la posició final correspon a les coordenades [-20,200] i un angle final de  $30^\circ$ .

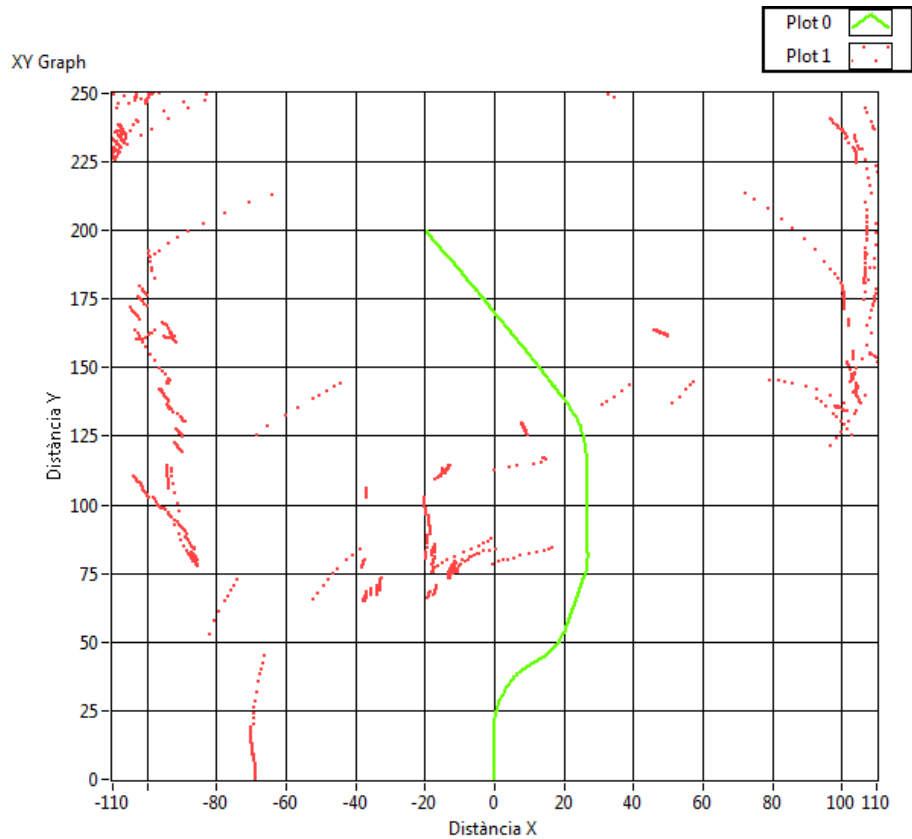


Figura 52. Prova 3

Com es pot observar, des de el primer moment el robot és controlat per l'agent Goto i després passa a ser controlat per l'agent Avoid que el fa avançar fins arribar a una distància de l'obstacle en que entra en acció i girar per evitar l'obstacle. Un cop ha passat l'obstacle torna a entrar en joc l'agent Goto que el reconduïx per arribar a la posició final. Tot i que no podem veure l'objecte de manera clara, amb la imatge es pot interpretar la zona on es troba.

### 8.2.3 Mapa Go to, Avoid i Gothrought

A continuació es pot veure un seguit de proves que s'han fet per veure el funcionament dels tres agents i com es van alternant per arribar al punt objectiu.

En el primer cas d'estudi el robot parteix de les coordenades  $[0, 0]$  amb una orientació de  $90^\circ$  i ha d'arribar al punt objectiu  $[320, 230]$  amb un angle de  $0^\circ$ .



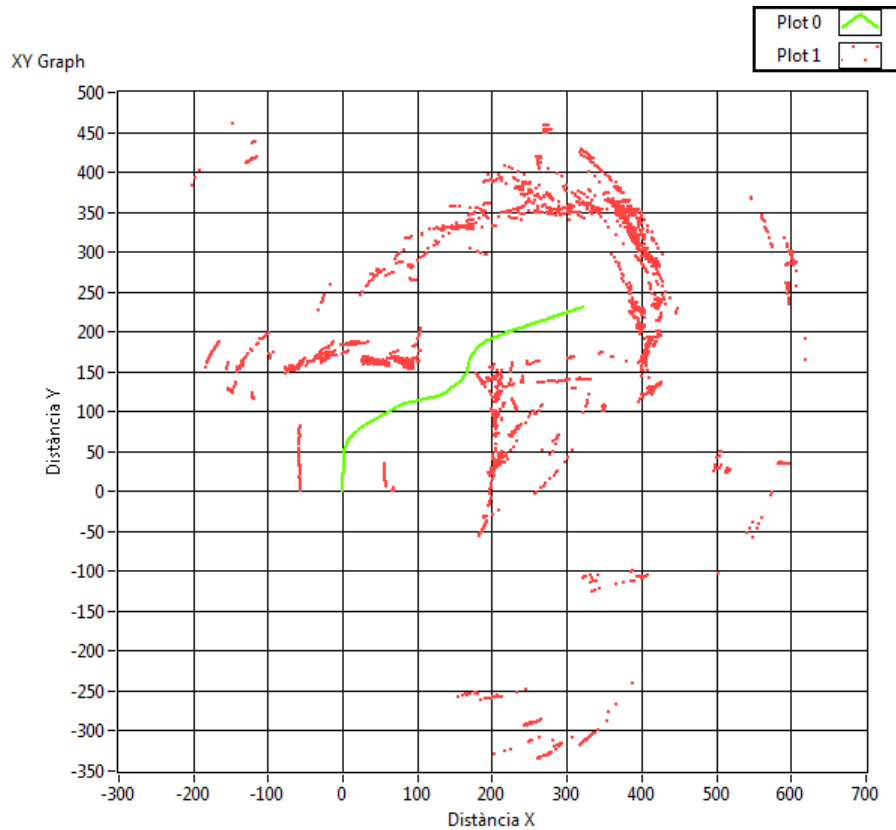


Figura 53. Prova 4

Com es pot observar, el robot parteix de dins un petit passadís on segurament aplica l'agent Gothrough, al sortir del passadís, es dirigeix directament cap al punt objectiu fent servir l'agent Go to, arriba un punt on ha de passar d'una habitació a una altra passant per una porta. En aquest cas, segurament primer apliqui l'agent Avoid per esquivar el punt de la porta més proper i un cop detecta obstacles a cada canto del robot aplica l'agent Gothrough un altre cop, però en aquest cas, només en un instant. A continuació, torna a aplicar l'agent Go to per arribar fins al punt de destí.

A continuació el robot parteix del punt  $[0, 0]$  amb una orientació de  $90^\circ$  per arribar al punt  $[100, 250]$  amb una orientació de  $0^\circ$ .

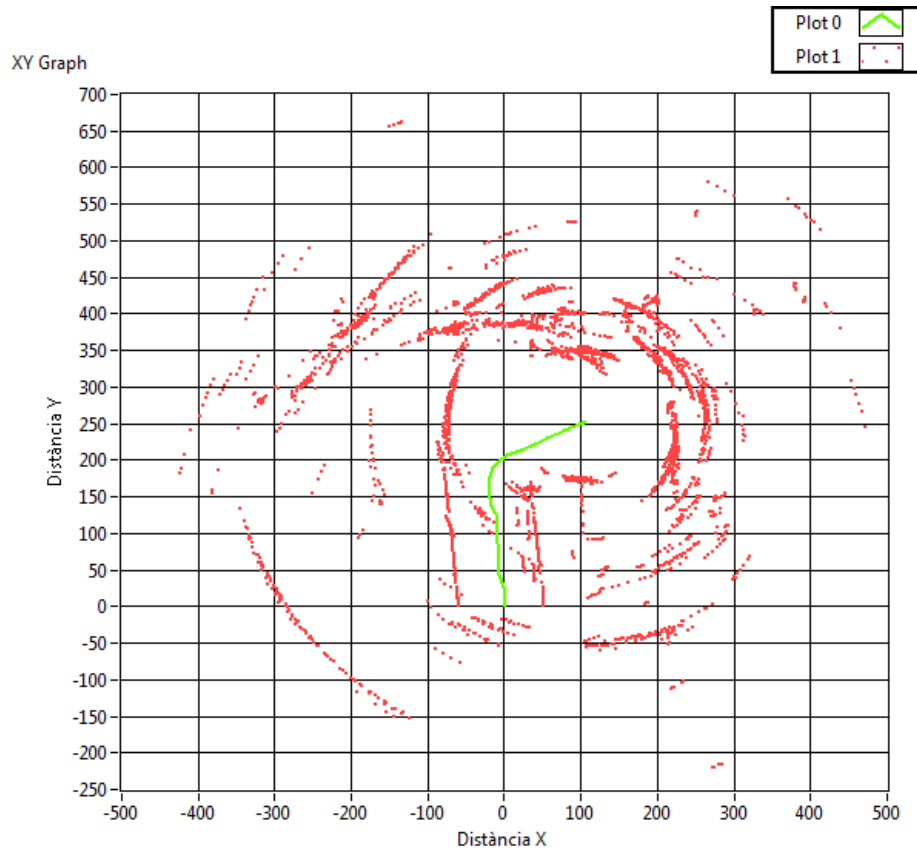


Figura 54. Prova 5

El robot parteix de dins un passadís on aplica l'agent Gothrough i al sortir d'aquest guanya pes l'agent Go to que agafa el control per arribar al punt de destí.

Quan el robot està dins un passadís molt llarg les distàncies del sensors mostren clarament la forma de l'espai per on es mourà el robot, com es pot veure a la imatge següent:

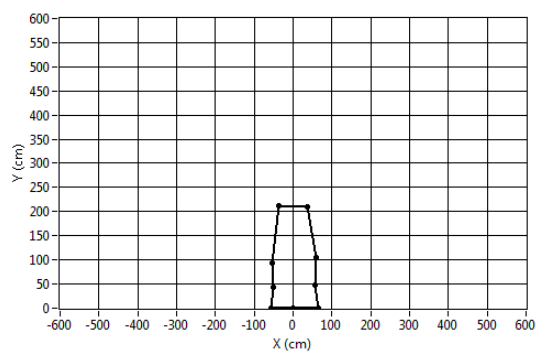


Figura 55. Representació d'un passadís

#### 8.2.4 Conclusions de les proves

S'ha pogut comprovar que el robot compleix amb l'objectiu d'arribar d'un punt fins a un altre esquivant els obstacles, però s'ha observat que el robot es comporta de manera una mica diferent depenent de l'entorn.

Hi ha entorns en els que s'han hagut d'ajustar alguns paràmetres dels agents com les velocitats o els pesos d'aquests, per realitzar moviments més eficients. També s'ha comprovat que depenent dels materials els sensors d'ultrasò reben més soroll i d'altres com la fusta que absorbeixen els ultrasons. Per aquest motiu, també s'hauria d'ajustar la placa de guany dels sensors i així obtenir una millor percepció dels obstacles.

## **9 RESUM DEL PRESSUPOST**

Aquest projecte de substitució de la placa de control i disseny i implementació del sistema de control per aconseguir que el robot es pugui desplaçar d'un punt fins a un altre esquivant els obstacles que hi hagi en el seu recorregut, ascendeix a un cost de tretze mil vint-i-cinc euros sense IVA.

## 10 CONCLUSIONS

El projecte ha complert amb els objectius bàsics que es proposava de substituir la placa de control del robot Pioneer 2DX per el dispositiu MyRIO de National Instruments i dissenyar un sistema de control capaç de que el robot es desplaci d'un punt a un altre esquivant els obstacles que es pugui trobar en el seu camí.

Per fer-ho s'ha dissenyat la FPGA necessària per adquirir les dades i tractar-les segons les necessitats dels agents del sistema de control distribuït que conté els agents necessaris per aconseguir una resposta eficient als estímuls de l'entorn.

D'aquesta manera, es pretén que el robot es pugui continuar fent servir per futurs projectes, ja que, amb el MyRIO i la programació amb LabVIEW obrim la porta que permetrà poder continuar optant a actualitzacions, tant de programació com de nous dispositius compatibles amb el nou dispositiu de control i així treure el robot de l'obsolescència.

Lluís Teixidor Arguimbau  
Graduat en Enginyeria Electrònica Industrial i Automàtica

Girona, 1 de Setembre de 2015

## **11 RELACIÓ DE DOCUMENTS**

Els documents que componen aquest projecte i compren tota la feina realitzada són quatre: el Plec de Condicions, la Memòria, l'Estat d'Amidaments i el Pressupost.

## 12 BIBLIOGRAFIA

ACTIVMEDIA ROBOTICS, LLC. Pioneer 2 / PeopleBot. Operations Manual. 2002.

DOERING, ED. NI MyRio Project Essentials Guide. National Instruments. 2014.

INNOCENTI, BIANCA. A Multi-agent Architecture with Distributed Coordination for an Autonomous Robot. PhD Tesis, Universitat de Girona, 2008.

OLLERO BATURONE, A. Robótica: Manipuladores y robots móviles. España: Marcombo, S. A., cop 1ª Edición. 2001