



EPS

Escola Politècnica

UdG

Superior

Treball final de grau

Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol: Estudi i caracterització del control de moviment per exteriors del robot mòbil Bigbot

Document: 1. Memòria

Alumne: Julián Alberto García Vélez

Director/Tutor: Albert Figueras Coma

Departament: Enginyeria Elèctrica, Electrònica i Automàtica

Àrea: Enginyeria de Sistemes i Automàtica

Convocatòria (mes/any): juny/2014

Índex

1	INTRODUCCIÓ	3
1.1	Antecedents.....	3
1.2	Objecte	3
1.3	Especificacions i abast	3
2	ROBOT MÒBIL BIGBOT	4
2.1	Estructura interna del robot.....	4
2.1.1	Litestation / Nanostation Loco.....	5
2.1.2	X-Port	5
2.1.3	Switch.....	6
2.1.4	Circuit de control.....	6
2.1.5	Motors i encoders	6
2.1.6	Bateries	7
2.1.7	IMU.....	7
2.2	Sistema de comunicacions	7
2.2.1	Connexió directa.....	8
2.2.2	Connexió indirecta.....	9
3	ENTORN DE TREBALL.....	10
4	DETECCIÓ TERRA	12
4.1	Entorns	12
4.2	Velocitats.....	13
4.3	Entorn de programació	14
5	APLICACIÓ FINAL	15
5.1	Velocitat en funció de la intensitat.....	19
5.2	Altres terrenys	19
5.3	Altres % de frens aplicats	21
6	COMPROVACIÓ CONTROL DE VELOCITAT	23
6.1	Rajola recte	24
6.2	Herba recte.....	26
6.3	Mosaic recte	28
6.4	Sorra recte.....	30

6.5	Rajola gir	32
6.6	Herba gir.....	34
6.7	Mosaic gir	36
6.8	Sorra gir.....	38
6.9	Paviment recte i gir	40
6.10	Interpretació en recte.....	45
6.11	Interpretació en gir.....	45
7	CONTROL DE POSICIÓ	46
8	MILLORES I TREBALL FUTUR.....	50
8.1	Millorar l'estudi.....	50
8.2	Millorar l'aplicació	50
9	RESUM DEL PRESSUPOST.....	51
10	CONCLUSIONS.....	52
11	RELACIÓ DE DOCUMENTS.....	53
12	BIBLIOGRAFIA	54
13	GLOSSARI.....	55
A	PROGRAMA DETECCIÓ DE TERRA.....	56
B	MANUAL COMUNICACIÓ LABVIEW - ROBOT BIGBOT	66
B.1	Com començar	66
B.2	Com moure el robot.....	68
B.3	Altres opcions de la Interfície.....	69
B.4	Controlador PID.....	72

1 INTRODUCCIÓ

Des de sempre l'home ha tingut el desig de posseir el coneixement del seu propi ésser, del seu comportament i el d'altres espècies, amb l'objectiu de crear agents amb la capacitat autònoma amb les quals pugui compartir la intel·ligència, per encarregar-li la realització de tasques que a ell li desagraden per ser monòtones, complicades i perilloses.

Aquests agents poden ser els robots mòbils que són aquells capaços de realitzar moviments en un entorn variable, és a dir, no estan fixats a un entorn físic en concret. Aquests robots utilitzen diferents tipus de navegació i solen tenir sistemes sensorials com sensors de distància i abast, sensors de posició, GPS, càmeres, etc.

1.1 Antecedents

En el laboratori del grup de recerca ARLAB es disposa d'una sèrie de robots mòbils pel rescat de supervivents en catàstrofes. Aquests permeten la comunicació i cooperació amb la gent i amb altres robots i identificar situacions de perill en operacions de rescat.

Són robots de quatre rodes per moure's per exteriors i estan equipats amb diferents sistemes de sensorització com càmeres de localització, GPS, infraroigs i odometria amb encoders.

1.2 Objecte

El present projecte servirà per dur a terme un control de moviment per un robot mòbil de rescat que navegarà per exteriors on el terreny no és uniforme ni únic. Aquest robot ha de poder detectar el terra on s'està movent per tal d'escollir els millors controladors PID per anar recte i girar. Es pretén que tot l'estudi realitzat pugui significar el punt de partida en aplicacions futures on el robot sigui molt més manejable i més eficient en les actuacions en ambients exteriors.

1.3 Especificacions i abast

Es crearà l'aplicació de la detecció de terra que es realitzarà en MATLAB i el càlcul i sintonització dels diferents controladors per cada terreny per millorar el moviment. S'utilitzaran les lectures de les velocitats dels encoders i corrents a partir dels assajos fets per cada terreny.

2 ROBOT MÒBIL BIGBOT

És convenient que s'expliqui l'agent mòbil més en detall ja que serà el que durà la programació i s'encarregarà de fer l'estudi per comprovar la validesa del PID en els diferents terrenys. A continuació es detallaran les seves característiques, que s'hauran de tenir en compte en el moment del disseny de l'aplicació.

El grup de recerca ARLAB disposa de varis tipus de robots mòbils de rescat, on dins d'aquests hi ha el tipus utilitzat en el projecte, l'anomenat Bigbot, del qual se'n disposen tres unitats.

2.1 Estructura interna del robot

A continuació es pot veure una imatge que detalla l'estructura bàsica dels robots.

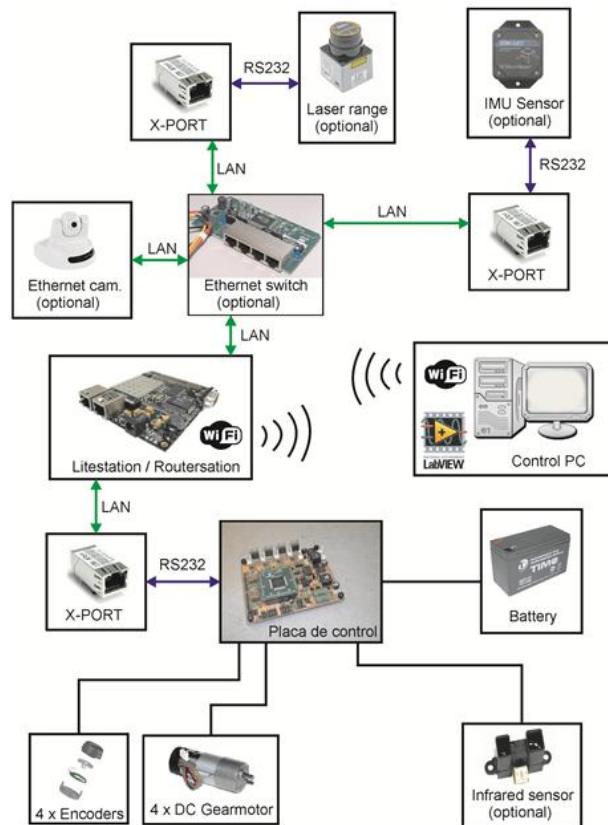


Figura 1. Esquema Robot

El Bigbot 2, és el robot que s'ha utilitzat i és un robot mòbil pel rescat de supervivents en catàstrofes. L'estructura interna bàsica és idèntica a la dels altres robots tot i que entre aquests hi poden haver canvis en quant a sensors i complements.

2.1.1 Litestation / Nanostation Loco

Un dels principals elements que ens trobem al robot és la Litestation, que ens permetrà fer la comunicació entre el robot i l'ordinador i viceversa. Es tracta d'un sistema "embedded" basat en un enrutador wifi. Segons el robot pot tractar-se d'un o altre model, en tot cas les prestacions són molt similars. En la Taula 1 es descriuen les principals prestacions dels diversos models.

Qualsevol dels dos models porta instal·lat OpenWrt, un sistema operatiu basat en Unix dissenyat especialment per aquest tipus de dispositius. Mitjançant aquest sistema operatiu es poden controlar les configuracions de xarxa WiFi i ethernet i altres aplicacions individualitzades.

	Litestation 2	Nanostatin Loco 2
Procesador	Atheros AR2316 SOC, MIPS 4KC, 180MHz	Atheros MIPS 4KC, 180MHz
Memòria	16MB SDRAM, 4MB Flash	16MB SDRAM, 4MB Flash
Interfície sèrie	On board RS232 with DB9 connector	RS232 (3.3 V level)
Interfície ethernet	2 X 10/100 BASE-TX (Cat. 5, RJ-45) Ethernet Interface	1 X 10/100 BASE-TX (Cat. 5, RJ-45) Ethernet Interface
Consum màxim	6.5 Watts	5 Watts
Alimentació	5-24V (18V max recommended)	12V
Interfície WiFi	2.4GHz 802.11b/g	2.4GHz 802.11b/g

Taula 1. Especificacions Litestations / Nanostation

Aquest dispositiu es pot considerar l'element central del robot ja que gestiona totes les comunicacions de les dades provinents i destinades als diferents sensors i actuadors via WiFi de cares a l'exterior del robot i ethernet de cares a l'interior del robot.

2.1.2 X-Port

El X-Port és un altre sistema "embedded" que permet transformar una comunicació sèrie RS-232 o RS-485 en una comunicació ethernet, és a dir, el que ens permetrà traspasar les dades del sensor a l'enrutador.

Donat que els sensors que s'utilitzen, la majoria d'ells, fan servir comunicació en sèrie RS-232 o RD-485 necessitarem aquests sistemes per convertir i fer arribar les dades en ethernet.

2.1.3 Switch

Donat que el robot pot disposar de molts sensors connectats via ethernet s'inclou un switch de 5 ports ethernet per ampliar el nombre de ports de l'enrutador.

En aquest cas es tracta d'un switch Cisco SD205 de cinc ports no gestionables. Donat que no es precisen prestacions especials per aquest dispositiu s'ha seleccionat pel seu tamany i consum de potència.

2.1.4 Circuit de control

El circuit de control s'encarrega de les tasques de control de més baix nivell. El seu funcionament es basa en un dsPIC33FJ256MC on s'executa una aplicació elaborada en C.

Arquitectura	16 bit – 40MHz
Memòria de programa	256 kB
Memòria RAM	30720 Bytes
Encapsulat	100 pin TQFP (85 i/o pins)
Control d'execució	PBOR, POR, WDT
Canals de memòria DMA	8
Entrades analògiques	24 x 12-bit @ 500 (ksps) 2-A/D
Comunicacions digitals	2 x UART, 2 x SPI, 2 x ECAN, 2 x I2C
Canals PWM pel control de motors	8 (16 bit resolution)
Canals Input-Capture	8
Timers	9 x 16 bit, 4 x 32 bit
Interfície QEI	1

Taula 2. Prestacions del dsPIC33FJ256MC

La principal tasca d'aquest circuit és el control dels quatre motors del robot, no obstant s'hi desenvolupen altres prestacions com la lectura dels polsos de l'encoder, mitjançant comptadors d'alta velocitat, i la lectura dels corrents de cada motor, mitjançant cel·les hall connectades a entrades analògiques del dsPIC per tal de controlar la velocitat lineal del robot hi ha implementat un control PID. El circuit inclou fonts d'alimentació commutades de 3.3 i 5V per tal d'alimentar els diversos dispositius del robot.

2.1.5 Motors i encoders

Les rodes dels robots son accionades per quatre moto reductors de corrent continu. Segons el robot varia el model i les especificacions del motor. Cada motor porta acoblat al seu eix un encoder òptic en quadratura d'alta resolució.

Els encoders s'acoblen de forma directa a l'eix del motor de forma que per determinar la velocitat del motor cal tenir en compte el factor de reducció del reductor i el diàmetre de la roda.

Els encoders en quadratura es caracteritzen per tenir dos canals de sortida. Els polsos dels dos canals es troben desfasats 90° entre ells de forma que observant la successió dels flancs de pujada o baixada entre els canals es pot determinar el sentit de gir dels motors, és a dir, si està anant endavant o endarrere.

2.1.6 Bateries

Les bateries del robot estan formades per deu bateries de níquel – metall hidrur de mida estàndard D, les típiques piles grosses dels radiocassets. Tenen un voltatge de 1.2V i donen uns 10.000 mA·h cadascuna. En total s'obté una tensió de 12V i una capacitat de 10A·h.

Serà el dsPIC l'encarregat de veure la tensió que hi ha en tot moment a les bateries mitjançant un convertidor AD.

2.1.7 IMU

Aquest és l'últim dels elements que conformen el BigBog. Com anteriorment hem dit es tracta d'un sensor de mesures inercials, altrament dit giroscopi. Aquest ens permet obtenir la posició angular del robot amb molta precisió. A més a més també disposarem de les acceleracions i velocitats angulars dels tres eixos.

També s'utilitzarà un X-Port per tal de poder-hi accedir.

2.2 Sistema de comunicacions

Donat que el robot i els seus sensors i actuadors formen una xarxa ethernet i/o wifi hi ha diverses opcions per tal d'establir les connexions (socket) per tal de teleoperar el robot mitjançant una aplicació que s'executa en una màquina fins la xarxa dels robots.

A efectes teòrics és indiferent si la màquina on s'executa l'aplicació de telecomandament es troba connectada a la xarxa per WiFi o per ethernet. Donat que la xarxa WiFi és de tipologia Mesh és necessari disposar d'un node compatible amb aquesta tecnologia, és a dir, no és possible connectar-se als robots mitjançant un node WiFi 802.11b/g estàndard.

A efectes teòrics es suposarà que les connexions als robots es realitzaran a través d'un node Mesh connectat per ethernet a la màquina de telecomandament.

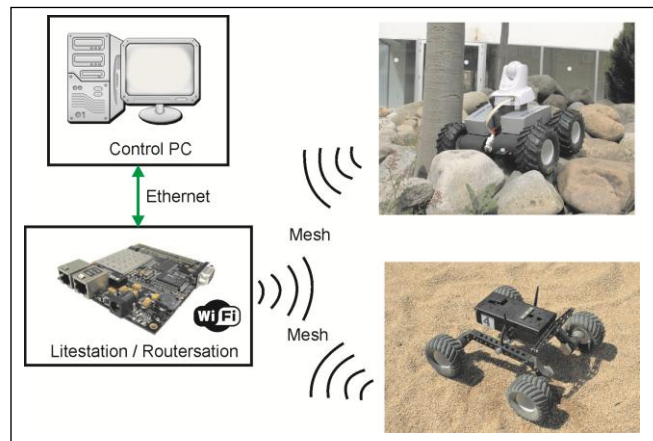


Figura 2. Comunicació amb els robots

2.2.1 Connexió directa

Per tal d'establir una connexió directa cal crear el socket directament amb el X-Port associat al circuit de control. D'aquest mode la Litestation del robot no intervé de forma activa en les comunicacions. La figura 3 representa de forma esquemàtica aquesta connexió. Cal tenir en compte que en aquesta figura el bloc "Control PC" representa de forma resumida l'estructura representada en la figura 2.

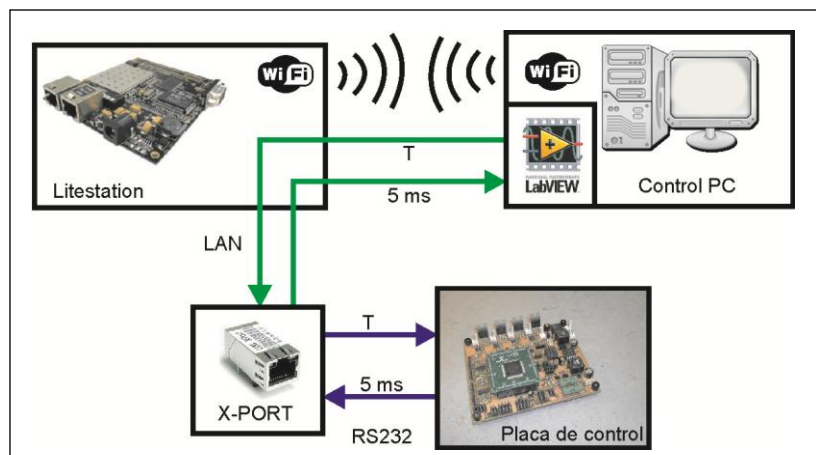


Figura 3. Comunicació directe

Per tal de mantenir la connexió amb el robot cal enviar els paquets de consignes al robot amb una mínima freqüència per tal que aquest no doni la comunicació per interrompuda (tancant el socket). Es recomana un període entre 50 i 150ms, aquest període assegura una continuïtat en les connexions alhora que no satura la xarxa. Per altre banda el període amb el qual es reben les respostes no és controlable i està fixat a 5ms.

Fent servir aquest mode de connexió s'obté la màxima resolució en la resposta del robot, tot i així no és el mode recomanat a no ser que es precisi aquesta resolució ja que la xarxa pot saturar-se fàcilment (sobretot en condicions de mala cobertura).

2.2.2 Connexió indirecta

Si es fa servir aquest mode de comunicació cal executar una aplicació en la Litestation del robot. Aquesta aplicació (comunicació) crearà el socket corresponent amb el circuit de control i es posa a l'espera de rebre una petició de connexió des del sistema de telecomandament.

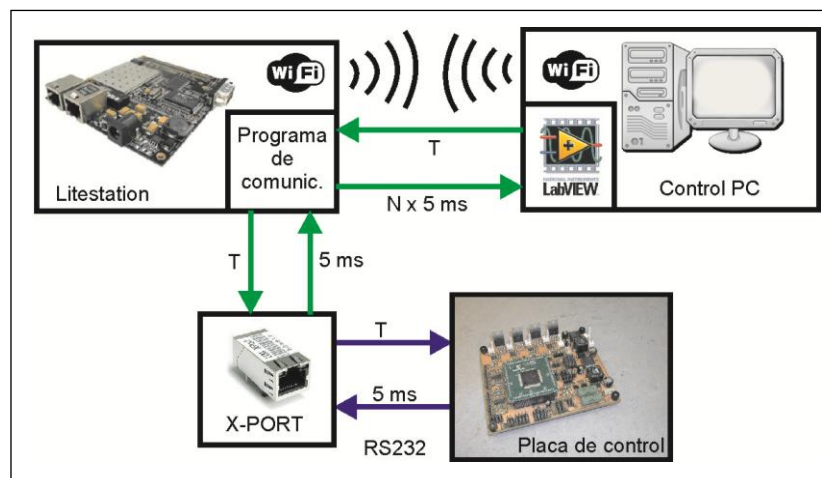


Figura 4. Comunicació indirecte

D'aquesta forma la Litestation intervé de forma activa en les comunicacions. Permetre aquesta intervenció comporta diversos avantatges. En primer lloc es pot controlar la freqüència amb la qual es retransmeten les dades rebudes del dsPIC (cada 5 ms) cap al sistema de control. Mitjançant el paràmetre "N" (Figura 4) es determina quants dels paquets rebuts es retransmeten al sistema de control, per exemple, per $N = 10$ es retransmet un de cada 10 paquets donant un període de $10 \cdot 5 = 50 \text{ ms}$.

Per altra banda aquest sistema permetrà, sobretot de cara al futur, implementar funcions bàsiques de control en el propi robot. Tot i que les comandes pel robot es reben des d'un sistema extern en aquest punt es pot influir sobre aquestes comandes en determinades situacions, per exemple, per evitar xocar amb un obstacle.

La comunicació amb la resta de sensors que pugui tenir el robot s'haurà de realitzar de forma independent.

3 ENTORN DE TREBALL

Per dur a terme el present projecte, el grup de recerca ARLAB ha proporcionat totes les eines necessàries per a fer possible l'estudi.

L'àrea de treball diari ha estat el laboratori de sistemes intel·ligent del P-IV on hi ha l'ordinador i l'antena wifi per a la comunicació amb el robot. En el mateix laboratori i en el pati exterior s'han fet els assajos pertinents als terrenys rajola, sorra i paviment.



Figura 5. Àrea de treball i antena wifi

Per a la comunicació amb el robot, s'ha utilitzat l'aplicació LabView feta per en Rafael Hesse. LabView és un entorn de programació destinat al desenvolupament d'aplicacions, en aquest cas, la que permet la comunicació amb el robot.



Figura 6. Portàtil i antena wifi

A l'hora d'analitzar totes les dades dels assajos, per fer gràfiques i càlculs, s'ha utilitzat el Microsoft Excel, software de full de càlcul. Aquest software ens ha permès dissenyar gràfiques per veure les respostes i fer la comprovació sobre el control de velocitat que

s'explica en l'apartat següent com també efectuar els càlculs per trobar el factor de correcció per a la millora de l'orientació del robot en l'apartat 7.

A l'inici del projecte, abans de començar els assajos es va haver de canviar les bateries del robot, sent l'únic problema que es va presentar abans de la realització del projecte.

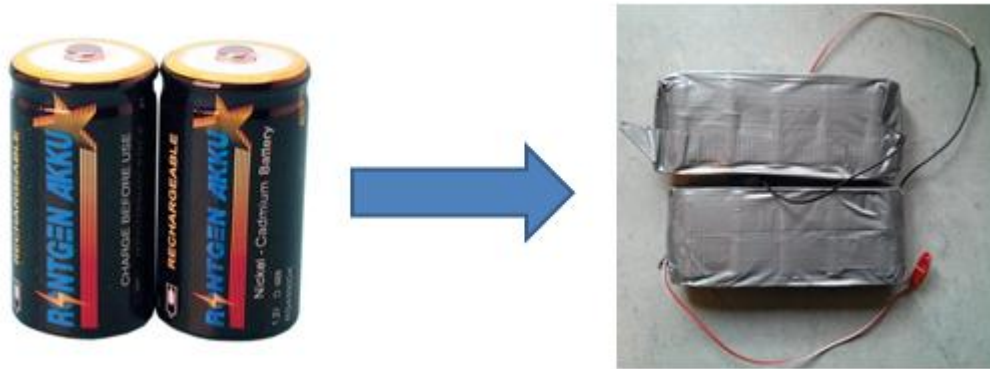


Figura 7. Bateries robot

Vam notar que la vida útil de les bateries era mínim, impeding treballar amb el robot durant molta estona. Així doncs vam procedir a canviar les bateries. Aquestes bateries són les típiques piles grosses dels radiocassets. Tenen un voltatge de 1.2V i donen uns 10.000 mA·h cadascuna. En total s'obté una tensió de 12V i una capacitat de 10A·h. Per aconseguir els 12V aquestes bateries estan connectades en sèrie.

4 DETECCIÓ TERRA

El robot Bigbot és un robot mòbil de rescat capaç d'actuar i interactuar en diferents entorns. Per aquest motiu és convenient que el robot sàpiga en quin terreny exterior està actuant amb l'objectiu de poder millorar la seva maniobrabilitat.

Es pretén que el robot detecti el terreny amb els seus propis sensors. L'ús dels encoders serà la base, aprofitant la lectura de les velocitats i intensitats per dotar-lo de major autonomia sense haver d'interaccionar amb la càmera que porta incorporada, una tècnica anomenada visió per computador.

4.1 Entorns

L'aplicació de detecció de terra està feta només pels quatre entorns que s'han escollit per a la realització del present projecte. Aquests entorns són la rajola, el mosaic, l'herba i la sorra.

RAJOLA



HERBA



MOSAIC



SORRA



Figura 8. Terres de treball

Aquests terres s'han escollit perquè són els més propers que hi han des d'on s'ha realitzat el projecte i perquè s'ha considerat que hi ha prou espai per realitzar els diferents estudis; a més a més a simple vista presenten diferents característiques de propietats físiques com per exemple la textura (rugositat, solidesa, etc.).

4.2 Velocitats

Les velocitats amb les que s'ha treballat han estat a 20 cm/s, 30 cm/s, 40 cm/s i 50 cm/s i amb la seva respectiva reducció del 20% a l'hora de la detecció del terra.

S'ha considerat que el robot a 20 cm/s és la velocitat mínima a la que el robot pot funcionar ja que per velocitats inferiors es podria donar el cas que a causa de la fricció de les rodes amb el terreny el robot no es mogués. Per altra banda la velocitat màxima de treball considerada ha estat 50 cm/s ja que depenent de l'entorn el robot entra en saturació i no pot anar més ràpid.

Aquesta saturació es pot notar en la lectura del senyal de control la qual ens dóna el % del voltatge aplicat al motor. Quan el robot està funcionant sobre recte, el senyal de control per a tots els terres a 50 cm/s pren un valor aproximat del 70% del voltatge, cosa que indica que encara pot anar més ràpid. El problema arriba quan el robot ha de girar, en el terra de mosaic i en el terra d'herba aquest senyal de control a 50 cm/s arriba al 100% i per tant ens indica que per a velocitats més altes ja no pot arribar a la velocitat desitjada i per tant efectuarà un gir no òptim. Per altra banda als únics terrenys que a 50 cm/s el senyal de control no és del 100% és a la rajola i la sorra, on està al voltant del 88%-98%, però s'ha volgut agafar com a velocitat màxima per tots els terrenys.

Velocitat [cm/s]	Reducció 20% [cm/s]
20	16
30	24
40	32
50	40

Taula 3. Velocitats de treball

Per dur a terme la detecció de terra el robot redueix un 20% la velocitat de les rodes del darrera. La necessitat de la reducció d'aquest percentatge de la velocitat és perquè per intensitats dèbils seria complicat aconseguir una variació d'intensitat entre els terrenys. Al frenar aquest 20% a les rodes del darrera, el robot tindrà més parell resistent aconseguint més tracció i en conseqüència les rodes del davant consumeixen més intensitat i això

provoca que es pugui detectar diferències entre els diferents terrenys. Això ho podrem veure més endavant mitjançant gràfiques.

La intenció de l'aplicació és que quan el robot detecti un canvi bruscat en la seva velocitat i per tant hi hagi un possible indicatiu d'una transició d'un terreny a un altre, pugui detectar el terreny en el que està navegant.

Per dur a terme la detecció de terreny es treballa amb el controlador PID de sèrie on la constant $K_p=0,0003$, la constant $K_i=0,05$ i la constant $K_d=0$.

4.3 Entorn de programació

L'entorn de treball utilitzat per desenvolupar el programa ha estat MATLAB que és el nom abreujat de MATrix LABoratory i és un entorn de computació numèrica i un llenguatge de programació propi. Va ser creat per la companyia The MathWorks i permet manipular fàcilment matrius, dibuixar, funcions i dades, implementar algorismes, crear interfícies d'usuari i comunicar-se amb altres programes en altres llenguatges.

5 APLICACIÓ FINAL

L'aplicació de la detecció de terra feta amb MATLAB ha estat realitzada pels terres on el robot ha funcionat gràcies als assajos però està totalment oberta per a possibles modificacions que suposin afegir nous entorns que el robot pugui conèixer. Aquesta aplicació és fàcil de modificar ja que és molt intuïtiva i senzilla.

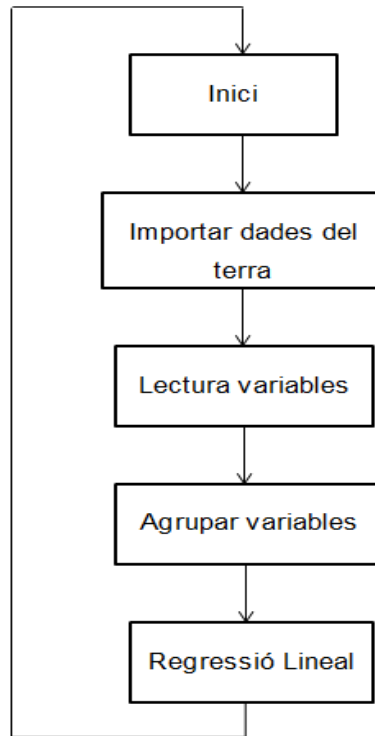


Figura 9. Programació detecció terra

Tal i com es mostra a l'organigrama, no es disposa de cap transició ja que l'estudi està realitzat en estat Offline.

L'estructura de l'aplicació es pot diferenciar en 4 blocs seqüencials començant per la importació de les dades, seguidament de la lectura de les variables necessàries per dur a terme la detecció, l'agrupació de variables i la regressió lineal que és el resultat final de l'aplicació.

S'ha escollit com a tècnica, per estudiar la relació de les variables, la regressió lineal. Ja que s'adapta a una gran varietat de situacions, entre altres per predir diferents aspectes en el comportament del robot.

Es pretén que aquesta aplicació pugui ser incorporada en el robot programant el pic per tal que quan el robot estigui actuant en un entorn i sigui necessari detectar el terra, el robot

llegeixi les dades Online i les vagi processant per tal de determinar el terreny i millorar la maniobrabilitat del Bigbot.

El programa importa les dades dels diferents terres d'assajos des d'un arxiu de disc on s'ha guardat la simulació i tot seguit les gestiona agafant la lectura de les variables que es necessiten per la diferenciació d'aquests terrenys.

```
%Dades Rajola
Imp=importdata('BB1_20_16_PIDFre20\Rajola.txt');
Data=Imp.data;
Imp2=importdata('BB1_30_24_PIDFre20\Rajola.txt');
Data2=Imp2.data;
Imp3=importdata('BB1_40_32_PIDFre20\Rajola.txt');
Data3=Imp3.data;
Imp4=importdata('BB1_50_40_PIDFre20\Rajola.txt');
```

Figura 10. Dades importades

Les variables que es fan servir per la detecció de terra són les velocitats, els corrents i el senyal de control de les dues rodes del davant del Bigbot. També agafa la lectura del nivell de bateria del robot mòbil la qual també es farà servir.

```
%% Separate all variables Rajola data1
speed_FL    = Data(101:200,2);
curr_FL     = Data(101:200,6);
speed_FR    = Data(101:200,3);
curr_FR     = Data(101:200,7);
battery     = Data(101:200,13);
ctl_FL     = Data(101:200,20);
ctl_FR     = Data(101:200,21);
```

Figura 11. Lectura de les variables

Aquestes variables són vectors que emmagatzemen dades cada 50 ms. Com que el treball realitzat ha estat en mode Offline en els assajos realitzats, a l'hora d'agafar les dades, s'ha treballat amb un temps de 10 segons de simulació per cada mostra.

Tal i com es pot veure a la figura 8, la propietat Data (Dades) conté la simulació de les dades emmagatzemades. Aquesta propietat conté totes les dades registrades durant la simulació. Aquesta propietat és una matriu $m \times n$ (Data (m,n)), on m és el número de passos de temps i n és el número de quantitats registrades. Les files de la matriu estan etiquetats pels punts de temps en la propietat Time, i les columnes són etiquetades per les dades en la propietat Data.

Per tant les variables a llegir, fan la lectura a partir del pas de temps 101 fins el 200 ja que s'ha considerat que a partir d'aquest pas de temps la simulació ja està en règim estacionari i tot el règim transitori ja ha desaparegut. És a dir, és el temps en que el robot en la simulació de l'assaig, ha aconseguit arribar a la consigna i per tant, ja està en règim estacionari.

Per a l'agrupació de variables es creen setze vectors nous on s'adjuntaran totes les dades de les quatre velocitats dels diferents terres de treball amb un sol vector. Els vectors nous que es creen són quatre vectors per a l'agrupació de totes les velocitats (AllspeedsFL) dels diferents terres, quatre vectors més per a l'agrupació del nivell de bateria (Allbattery) en la simulació dels diferents terres, quatre vectors més per a l'agrupació del senyal de control (AllctlFL) dels diferents terres i quatre últims vectors més per a l'agrupació dels corrents (AllcurrentsFL) dels diferents terres en les seves respectives simulacions.

```
for z=1:100
    %%Vectors de velocitats FL
    AllspeedsFL(z)=speed_FL(z);
    AllspeedsFL(z+100)=speed_FL2(z);
    AllspeedsFL(z+200)=speed_FL3(z);
    AllspeedsFL(z+300)=speed_FL4(z);
```

Figura 12. Agrupació variables

Tal i com es nota a la figura 9, mitjançant una estructura for, recorrem els quatre vectors, un vector per cada velocitat, i les dades d'aquests les emmagatzemem al nou vector per tal d'agrupar-les. Per tant el vector AllspeedsFL, és un vector que hi conté el vector de la velocitat de 20 cm/s, el vector de la velocitat de 30 cm/s, el vector de la velocitat de 40 cm/s i el vector de la velocitat 50 cm/s.

```
%%Vector de POTFL
AllPotFL(z)=AllctlFL(z)*Allbattery(z)*AllcurrentsFL(z);
AllPotFL(z+100)=AllctlFL(z+100)*Allbattery(z+100)*AllcurrentsFL(z+100);
AllPotFL(z+200)=AllctlFL(z+200)*Allbattery(z+200)*AllcurrentsFL(z+200);
AllPotFL(z+300)=AllctlFL(z+300)*Allbattery(z+300)*AllcurrentsFL(z+300);
```

Figura 13. POT per la diferenciació de terrenys

Per acabar l'agrupació de dades, al final multipliquem els vectors AllctlFL, Allbattery i AllcurrentsFL entre sí per aconseguir el vector POT que ens servirà per trobar el resultat final. El que es vol aconseguir és una gràfica de les velocitats en funció de la potència. Aquest treball és la potència que respon a la fórmula següent:

$$\text{Pot} = V \cdot I = (\text{PWM} \cdot \text{Bateria}) \cdot I = (\text{AllctlFL} \cdot \text{Allbattery}) \cdot \text{AllcurrentFL} \quad (\text{Eq.1})$$

Totes les figures segueixen el treball de programació del terreny rajola. A l'annex es pot veure tota l'aplicació al complet amb tots els terrenys treballats.

Finalment per dur a terme la detecció de terra, es fa una regressió lineal on en l'eix d'abscisses hi figuren les diferents velocitats i en l'eix y la Potència consumida.

Tal i com es veu a la figura, els terres estan diferenciats per colors i cada grup de punts és la lectura de la velocitat en funció de la Potència en el seu temps de mostreig quan el robot ja s'ha estabilitzat a la velocitat de consigna. Cada punt és una mostra agafada al fer-se la lectura cada 50 ms.

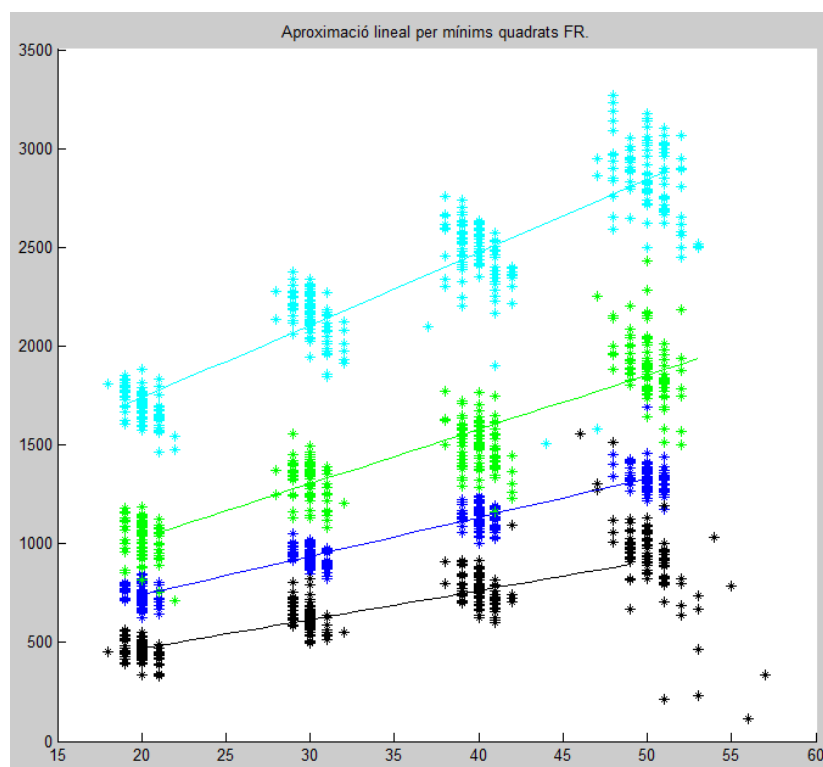


Figura 14. Aproximació lineal terrenys amb fre al 20%

Els punts negre són les lectures dels assajos fets a la sorra, els punts blaus fosc segons l'aplicació es refereixen al terra rajola, els punts verds es refereixen al terra herba i per últim els punts de color blau cel es refereixen al terra mosaic.

El motiu de frenar un 20% les rodes del darrera és perquè només amb aquesta frenada s'aconsegueix una separació clara entre els punts dels diferents terres per poder diferenciar-los.

La idea d'aquesta aplicació, és trobar el % de fre adient i veure gràficament com els terrenys es poden diferenciar entre sí. Així doncs es pretén que aquesta aplicació sigui el punt de

partida per idear la programació en C del pic perquè el robot mòbil en estat online vaig llegint les dades i indiqui en quin terreny s'està movent mitjançant la lectura de la velocitat en funció de la potència.

5.1 Velocitat en funció de la intensitat

L'aplicació de detecció del terreny s'aconsegueix mitjançant l'aproximació lineal de la velocitat en funció de la potència activa. Aquest apartat explica perquè en l'aplicació no s'ha escollit com a mètode final la velocitat en funció de la intensitat, com s'havia pensat inicialment.

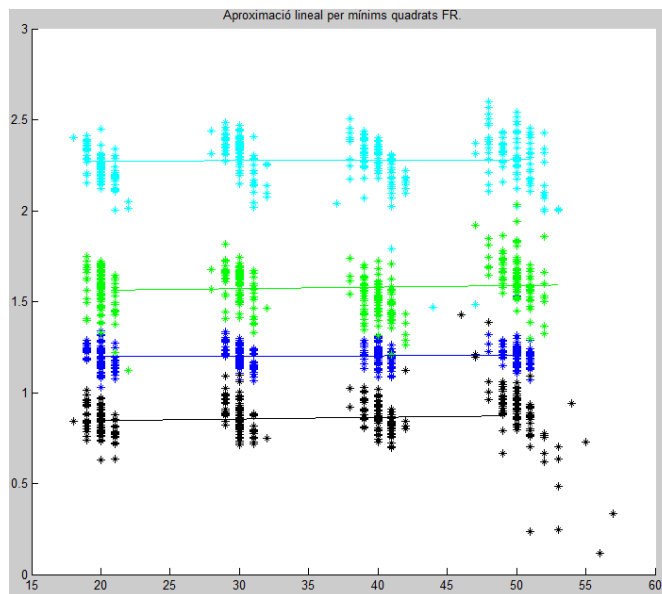


Figura 15. Aproximació lineal velocitat en funció d'intensitat

Tal i com es nota a la imatge, en l'eix y hi figuren els valors d'intensitat consumits pel robot per a cada velocitat i terreny, essent els punts de color blau cel la lectura del terra mosaic, els punts de color verd la lectura del terra herba, els punts blaus fan referència a la lectura del terra rajola i els punts negres a la sorra amb les seves respectives línies de regressió.

La intensitat és l'esforç del robot i és proporcional al parell el qual ens dóna una idea de com evoluciona la potència però manca d'informació i per aquest motiu s'ha volgut treballar en funció del treball (W) que ens dóna el nivell de bateria, agafa la lectura de la intensitat com també la del senyal del control.

5.2 Altres terrenys

A part de l'estudi dels terrenys rajola, mosaic, herba i sorra també s'ha fet l'estudi pel terra paviment de formigó. Aquest terra presenta una rugositat molt superior a la dels altres cosa

que fa complicada la seva lectura de velocitat en funció de la potència. Aquest paviment rep normalment el nom de formigó desactivat el qual presenta avantatges com la resistència, llarga durabilitat i integració perfecta per totes les seves aplicacions (voreres, zones comunes, aparcaments exteriors, places, camins, passejos, etc) però no és aconsellat per trànsit rodat.



Figura 16. Paviment de Formigó

Tal i com es denota a la figura següent, el terreny paviment, que són els punts vermells, té molts més punts de dispersió que cap altre terra, a més a més, el pendent és força semblant al del mosaic.

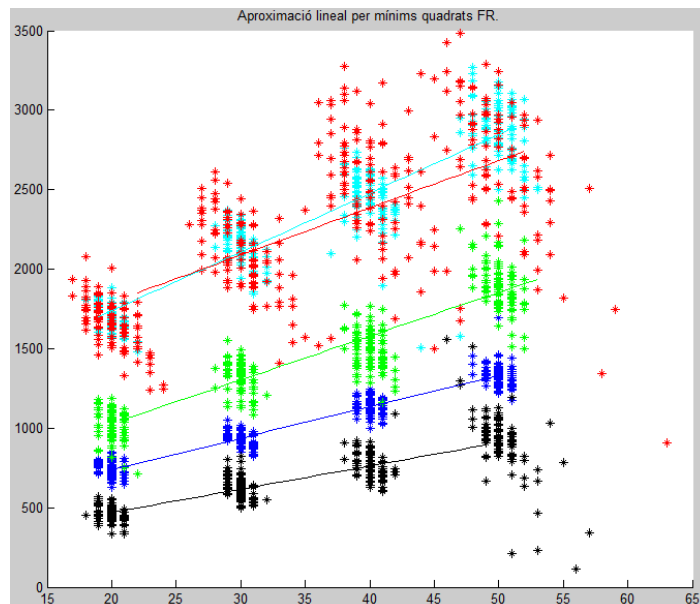


Figura 17. Aproximació lineal 5 terrenys

Tot i que sigui difícil la diferenciació per la lectura de les variables no es pot menysprear el terra ni molt menys considerar com a no apte, ja que en qualsevol actuació de rescat es podria donar el cas que hagués de passar per aquest terra.

S'ha considerat doncs, que el terreny paviment és el mateix que el terreny mosaic.

5.3 Altres % de frens aplicats

En aquest apartat l'objectiu principal és entendre perquè frenar un 20% les rodes del darrera és la millor opció davant altres percentatges.

Velocitat [cm/s]	Reducció 15% [cm/s]	Reducció 25% [cm/s]
20	17	15
30	26	22
40	34	30
50	42	37

Taula 4. Reducció dels diferents valors de velocitat

Tal i com es nota a la taula, els valors reduïts de les velocitats són sempre números enters. En alguns casos, la reducció ha provocat un valor de velocitat amb decimal, però s'ha arrodonit sempre cap a baix, ja que l'objectiu principal que es pretén és frenar les rodes del darrera.

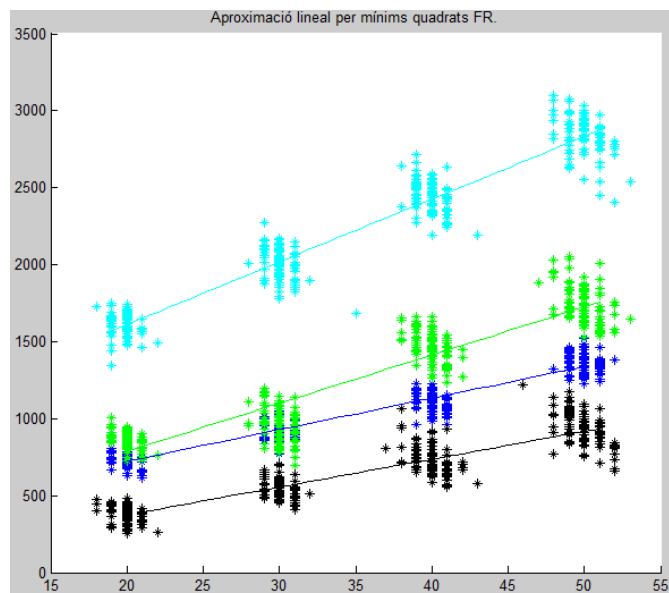


Figura 18. Aproximació lineal terrenys amb fre al 15%

Com es pot veure a la figura, per velocitats petites, no s'aconsegueix una diferenciació clara entre els terrenys rajola i herba solapant-se molts punts entre sí. Per aquest motiu s'ha estimat no convenient frenar les rodes del darrera un 15% per la detecció de terra.

No passa el mateix frenant un 25%, amb el qual sí s'aconsegueix una gran diferenciació entre els terrenys però s'ha estimat més un 20% com a percentatge de frenada ja que el consum serà menor.

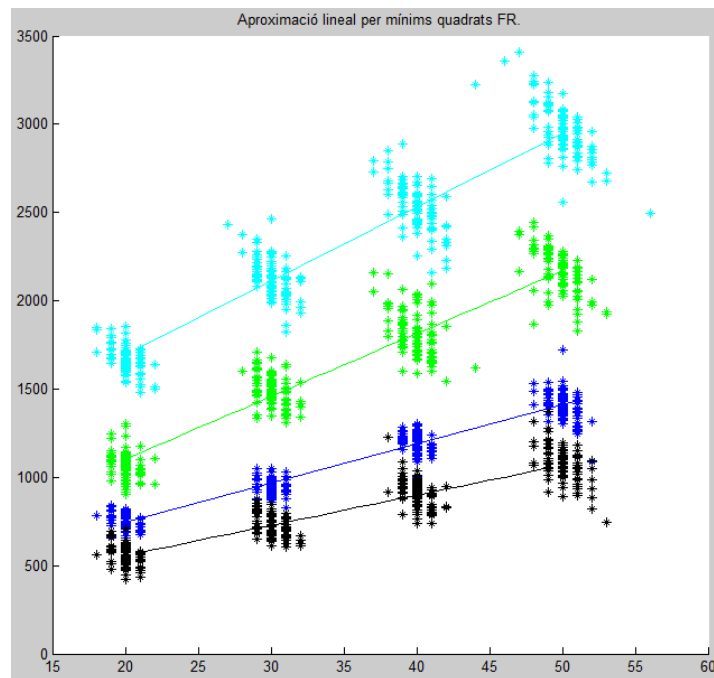


Figura 19. Aproximació lineal terrenys amb fre al 25%

Frenant un 25% les rodes del darrera necessitaran molta més tracció que frenant un 20% i per contra les rodes del davant hauran de consumir molt més si volen arribar a la velocitat consigna. Per altra banda la millora no és molt significativa.

6 COMPROVACIÓ CONTROL DE VELOCITAT

L'objectiu del control de velocitat és que sigui capaç de fer que la sortida segueixi la consigna fixada, segons unes especificacions establertes. Es pot arribar a la consigna de múltiples maneres. Que s'hi arribi d'una forma o d'una altra dependrà del tipus de procés que es tingui i quines són les especificacions de control (que no hi hagi sobrepics, arribar-hi amb el temps mínim...) i per escollir-la s'aconsegueix variant els paràmetres de K_p , K_i i K_d del PID. El fet d'ajustar aquestes constants a uns valors determinats és el que s'anomena sintonitzar el PID.

Aquesta sintonia es poden fer de moltes maneres. Per una banda si es disposa del model del procés a controlar, es poden calcular els paràmetres PID de manera que el sistema global en llaç tancat tingui els pols situats en un lloc del pla complexa adient per a produir una resposta transitòria desitjada. També es poden sintonitzar els PIDs utilitzant mètodes freqüencials.

Però no sempre es disposa del model del sistema, que és el cas del present projecte. Per tant en aquests casos s'haurà de fer una sintonització de forma empírica.

Abans però, caldrà comprovar si el controlador actual és correcte per a tots els terrenys i veure les diferents respostes en els diferents entorns.

El controlador del qual disposa l'agent mòbil és un controlador PI amb un valor K_p de 0.0003 i un valor K_i de 0.05.

Aquest controlador és la solució que es va adoptar quan es va crear el robot. Aquest controlador es va determinar a partir del terra rajola i no es van estudiar altres tipus d'entorns on el robot es pogués moure per aquest motiu s'ha decidit estudiar altres entorns per comprovar si el controlador que hi ha per defecte és adient per aquests nous entorns o s'ha de sintonitzar uns nous paràmetres.

En l'estudi s'han realitzat diferents proves per les diferents velocitats de treball (20 cm/s, 30 cm/s, 40 cm/s i 50 cm/s) i sempre per partida doble, fent la mitja de les dues lectures amb la intenció d'evitar el màxim error possible a l'hora de les lectures.

Les especificacions marcades a l'hora de veure les respostes és que el controlador garantitzi un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3 s, ja que són valors raonables per un robot mòbil de rescat.

L'estudi s'ha fet amb el robot Bigbot funcionant en recte i girant sobre el seu propi eix.

6.1 Rajola recte

L'estudi ha començat pel terreny rajola ja que va ser amb el terra que es va escollir el controlador actual per comprovar el correcte funcionament del controlador. Tal i com es poden veure a les figures, la resposta és bastant correcta. Gràficament es pot comprovar que aquest controlador garanteix un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3s.

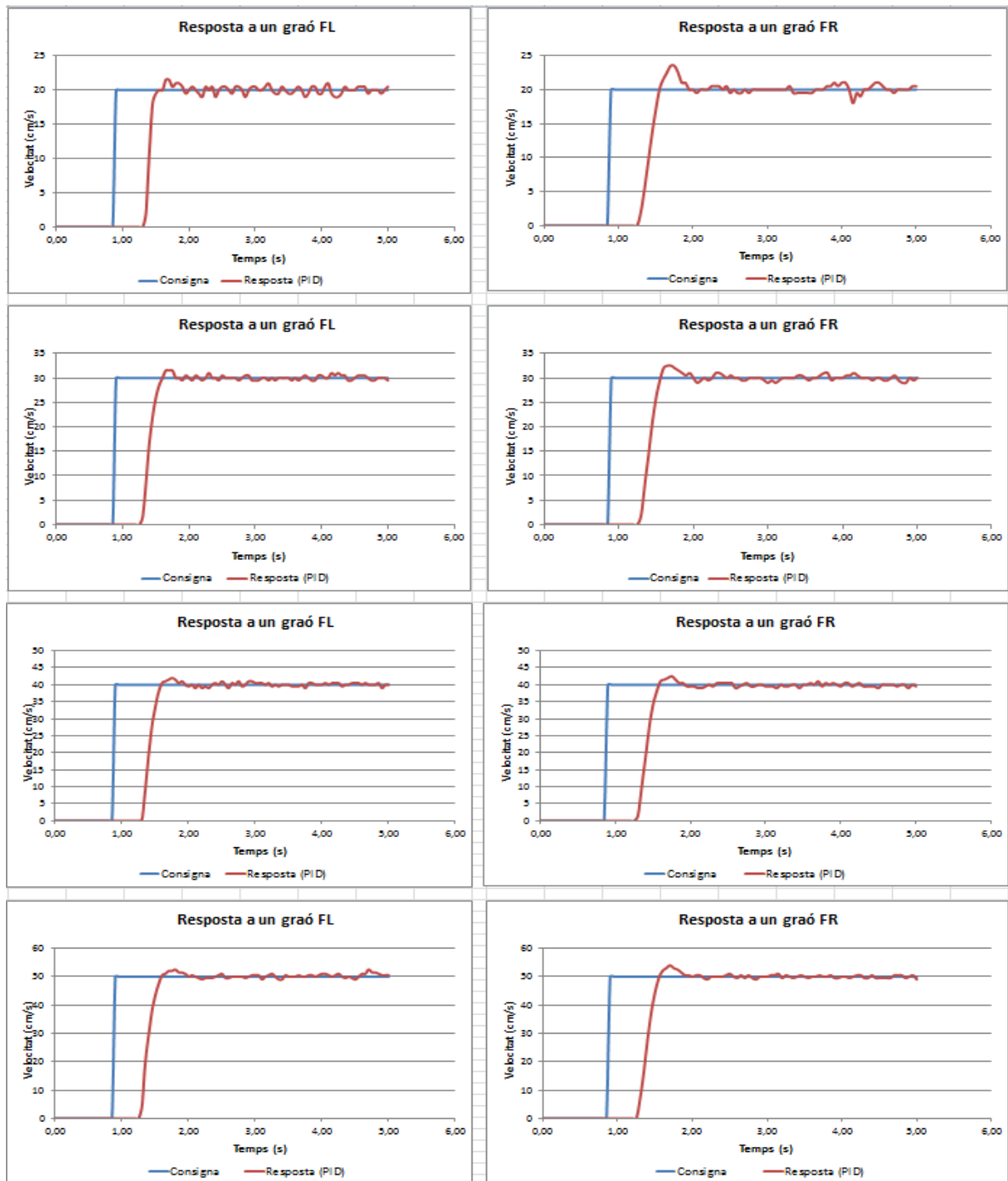


Figura 20. Respostes recte rajola rodes davant

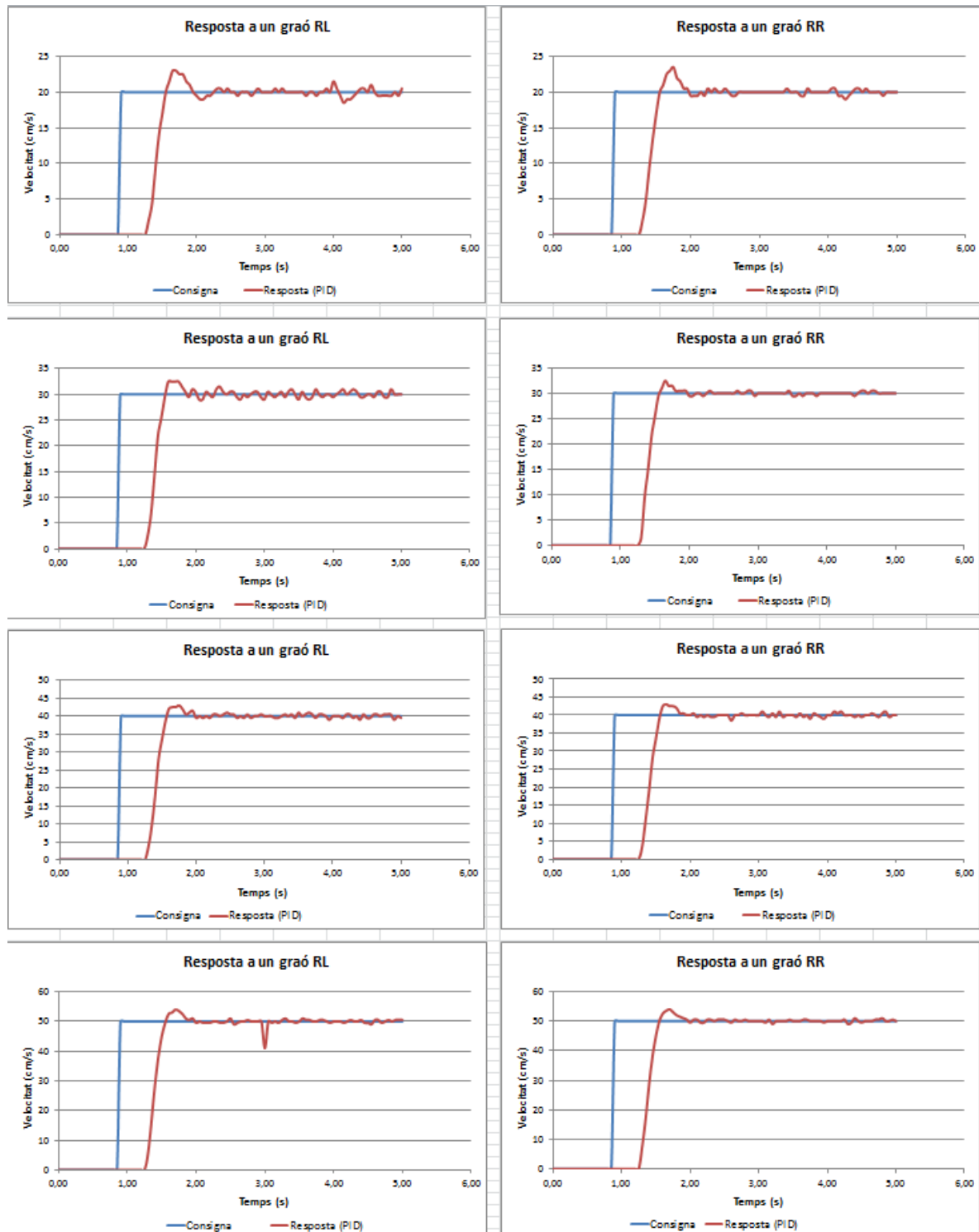


Figura 21. Respostes recte rajola rodes darrera

La lectura de les rodes és la següent, d'esquerra a dreta roda de davant esquerra, roda dreta del davant, roda esquerra del darrera i roda dreta de darrera. I la lectura de dalt cap a baix ens indica les velocitats. Tal i com es veu a la figura 21, la resposta de la roda esquerra de darrera a 50 cm/s hi ha una lectura que significa alguna pertorbació en el seu recorregut.

6.2 Herba recte

L'estudi d'aquest terra s'ha fet a l'herba de les escales davant de l'edifici del P2 de l'Escola Politècnica. La resposta és semblant al terra rajola.

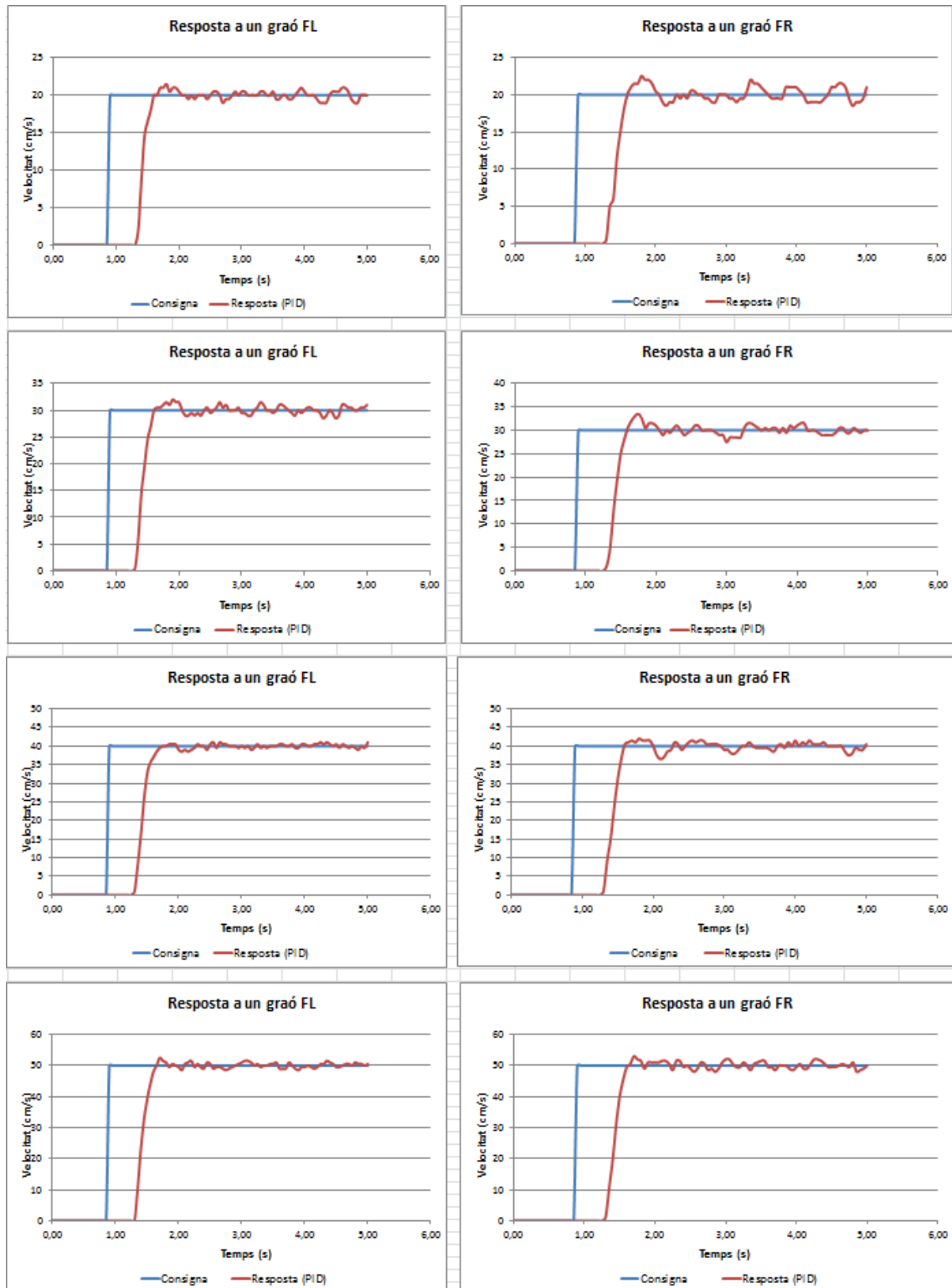


Figura 22. Respostes recte herba rodes de davant

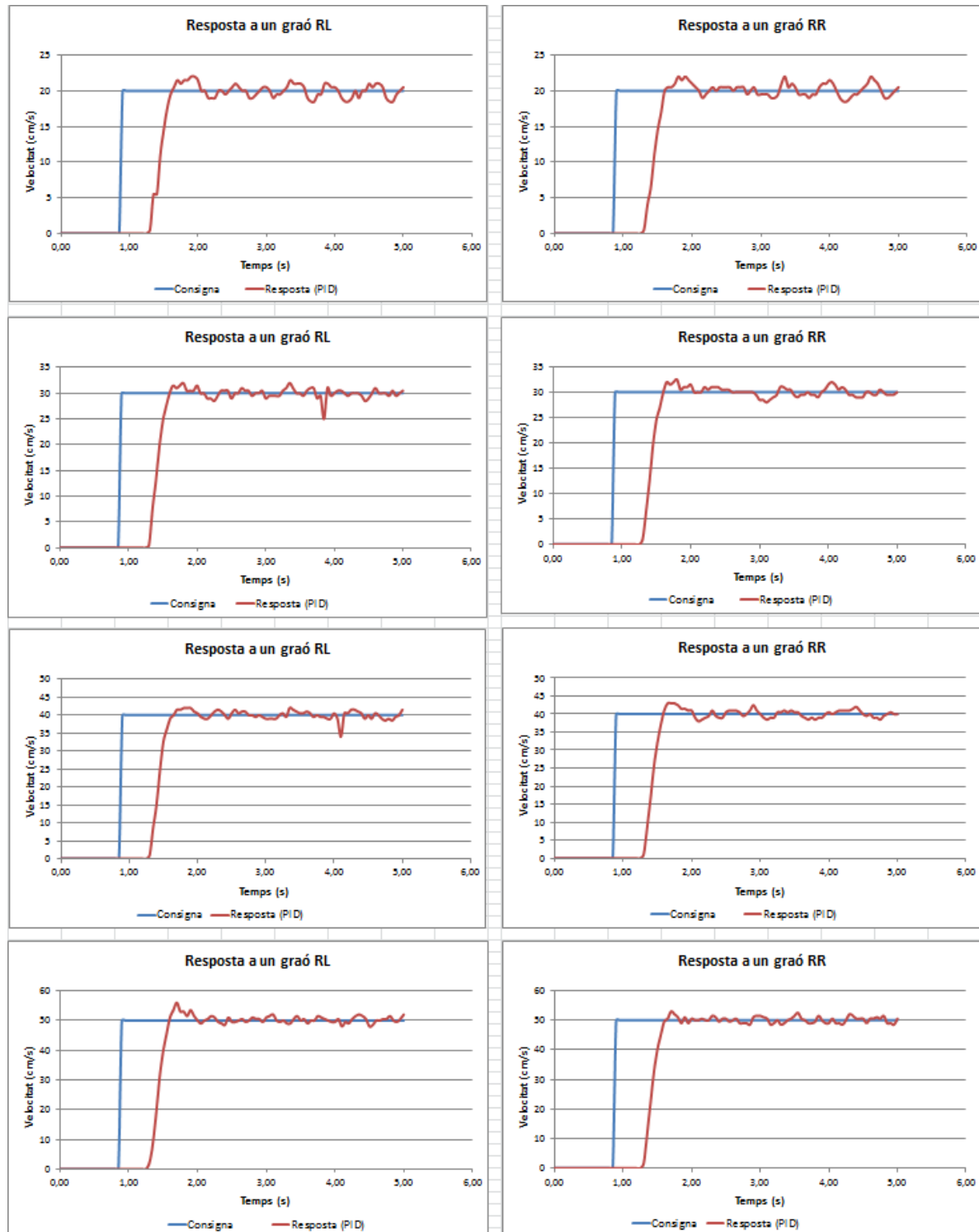


Figura 23. Respostes recte herba rodes de darrera

En les gràfiques de les rodes del darrera es nota algun pic però això és degut a la irregularitat del terreny ja que es pot comprovar a simple vista que les respostes són bastant constants. El controlador per aquest terreny garanteix un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3s.

6.3 Mosaic recte

El terra mosaic és l'entorn exterior més regular juntament amb la rajola dels quatre ambients estudiats. Els assajos d'aquest terreny s'han fet al mosaic que està a l'inici de les escales que comuniquen l'edifici P2 i P4 de l'Escola Politècnica.

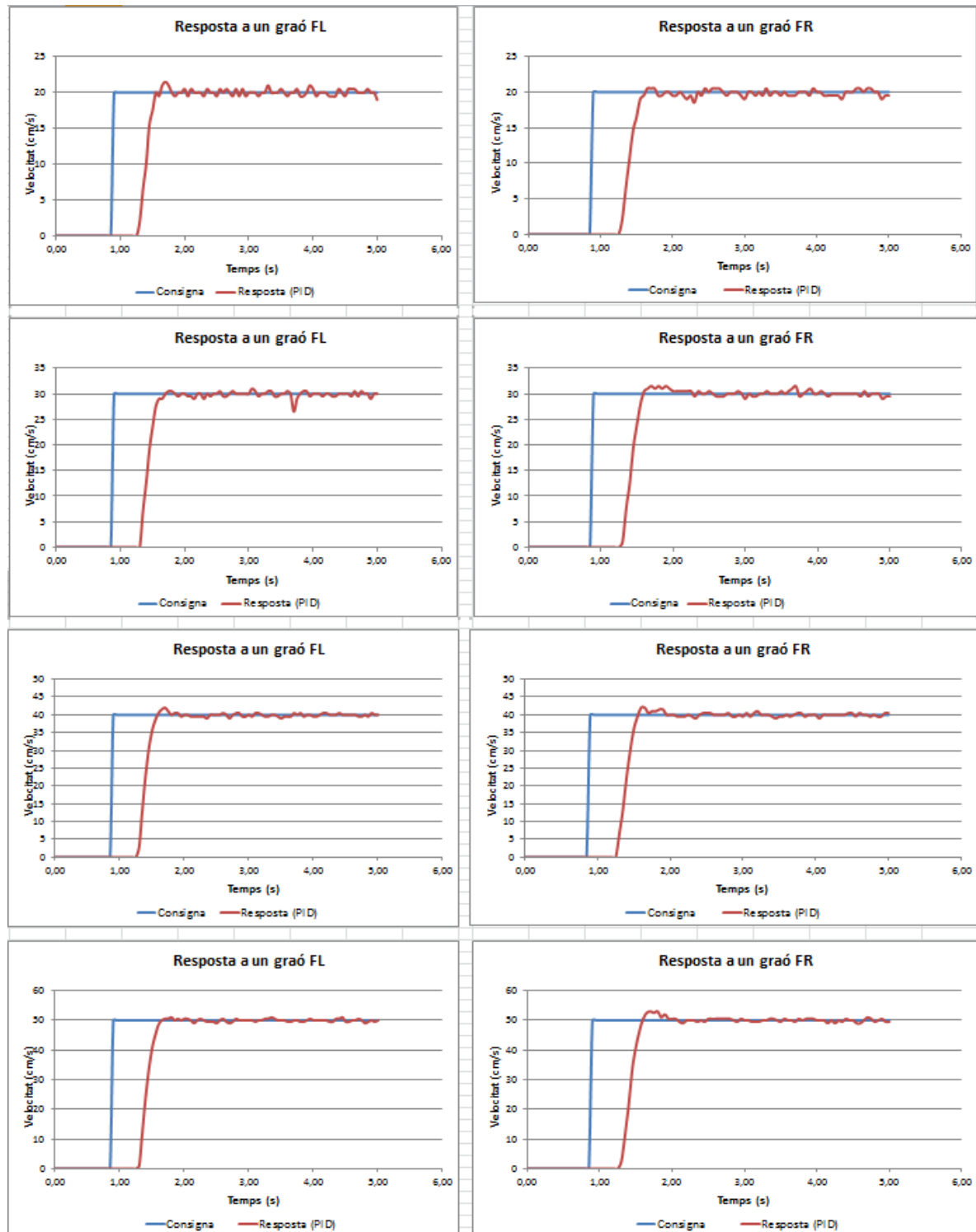


Figura 24. Respostes recte mosaic rodes de davant

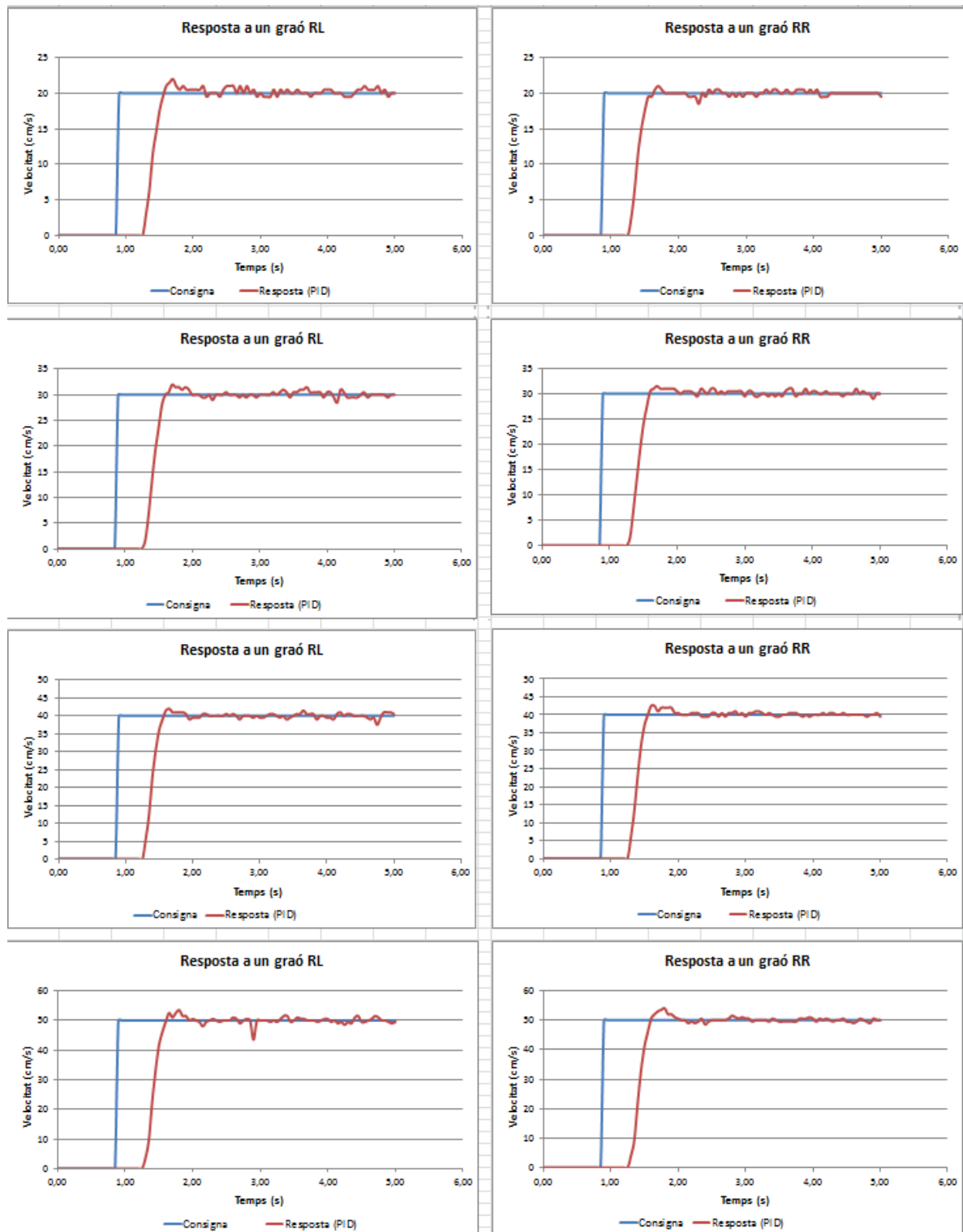


Figura 25. Respostes recte mosaic rodes de darrera

Gràficament es pot veure que hi ha poc soroll i que les respostes són molt semblants a les respostes del terra rajola. El controlador per aquest terreny garanteix un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3s.

6.4 Sorra recte

En aquest entorn les rodes del robot llisquen molt dificultant l'avançament del robot per les velocitats 20 cm/s i 30 cm/s. Per 40 cm/s i 50 cm/s el funcionament és més correcte. El terreny es troba fora del laboratori del grup de recerca ARLAB.

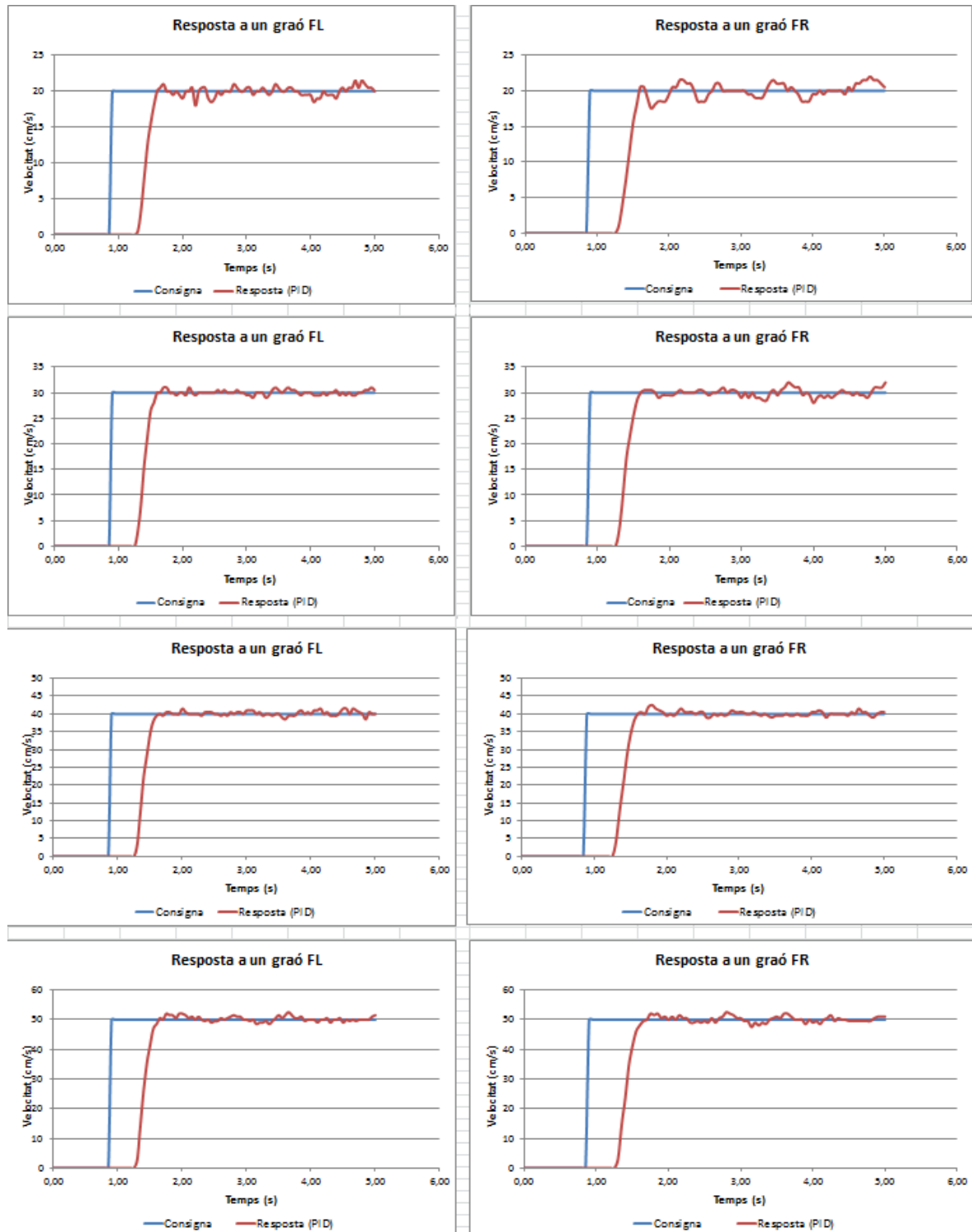


Figura 26. Respostes recte mosaic rodes de davant

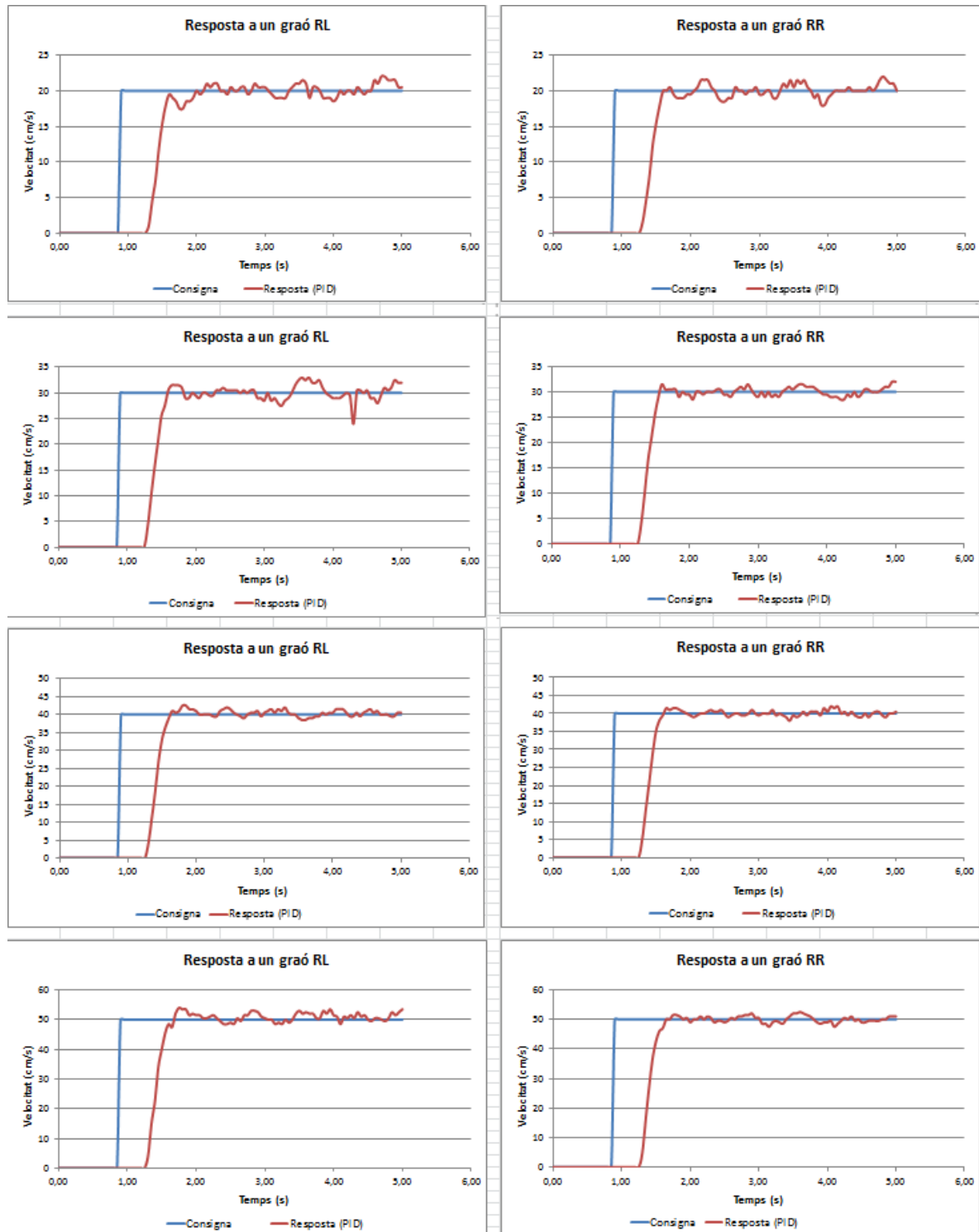


Figura 27. Respostes recte sorra rodes de darrera

Tot i que algunes respostes demostrin moltes oscil·lacions, el controlador PI per aquest terreny garanteix un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3s.

6.5 Rajola gir

En aquest terreny la fricció de les rodes amb el terra és baixa i per tant no hi ha gaire dificultat en que el robot pugui efectuar el moviment.

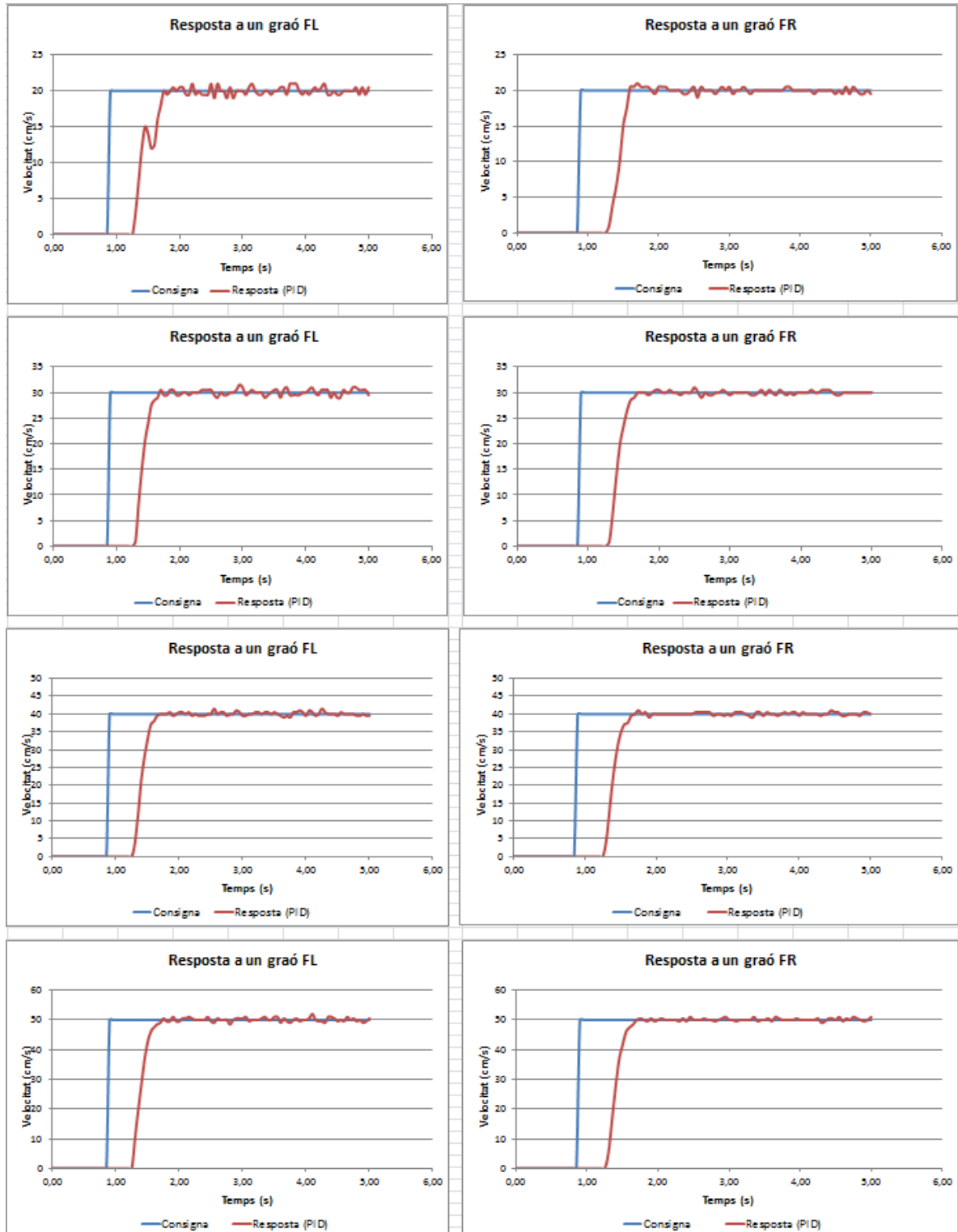


Figura 28. Respostes gir rajola rodes de davant

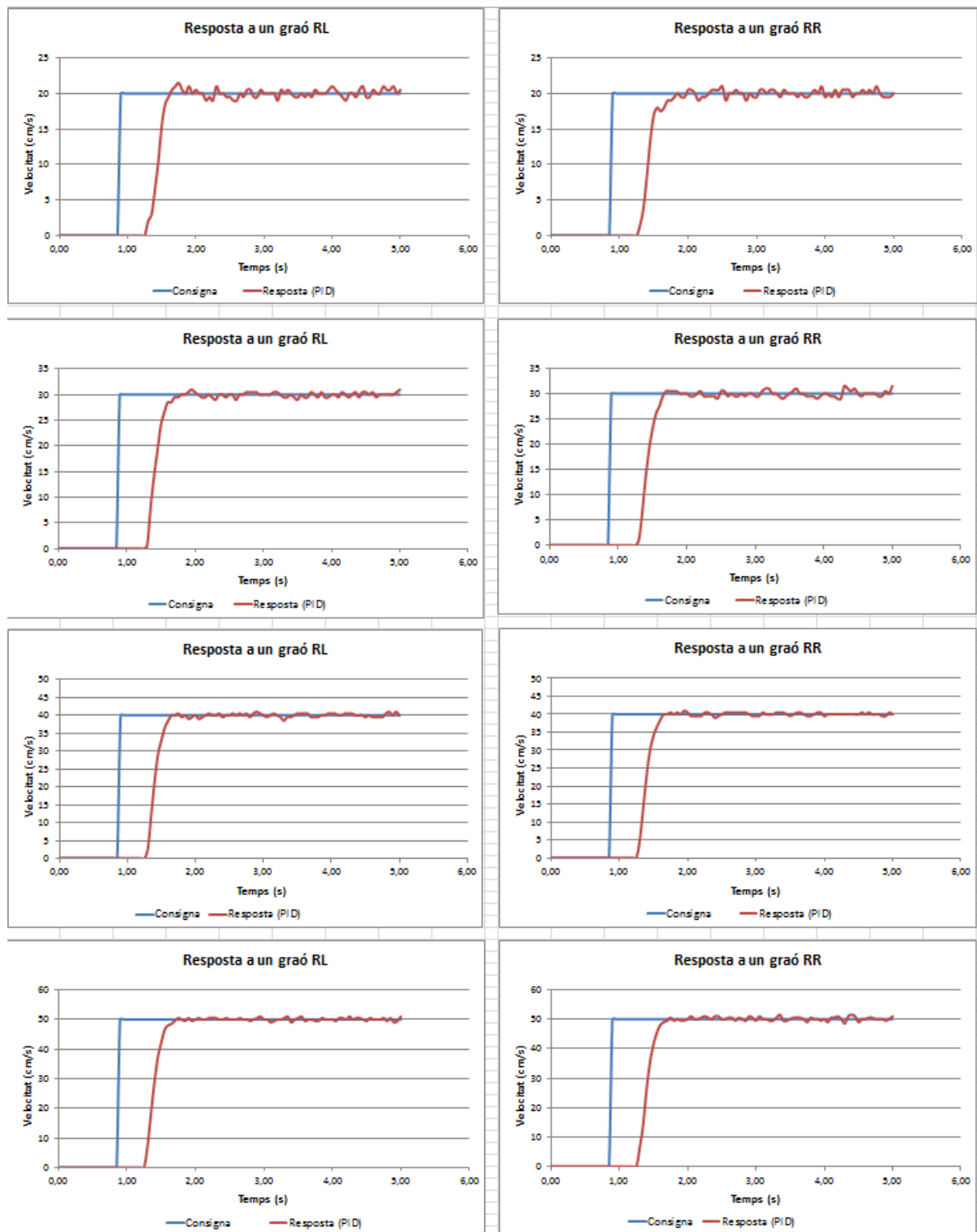


Figura 29. Respostes gir rajola rodes de darrera

Gràficament es pot comprovar que el controlador per aquest terreny garanteix un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3s.

6.6 Herba gir

En aquest terreny es pot veure molta irregularitat en els gràfiques, sobretot en la roda esquerra de davant i en la roda dreta de darrera.

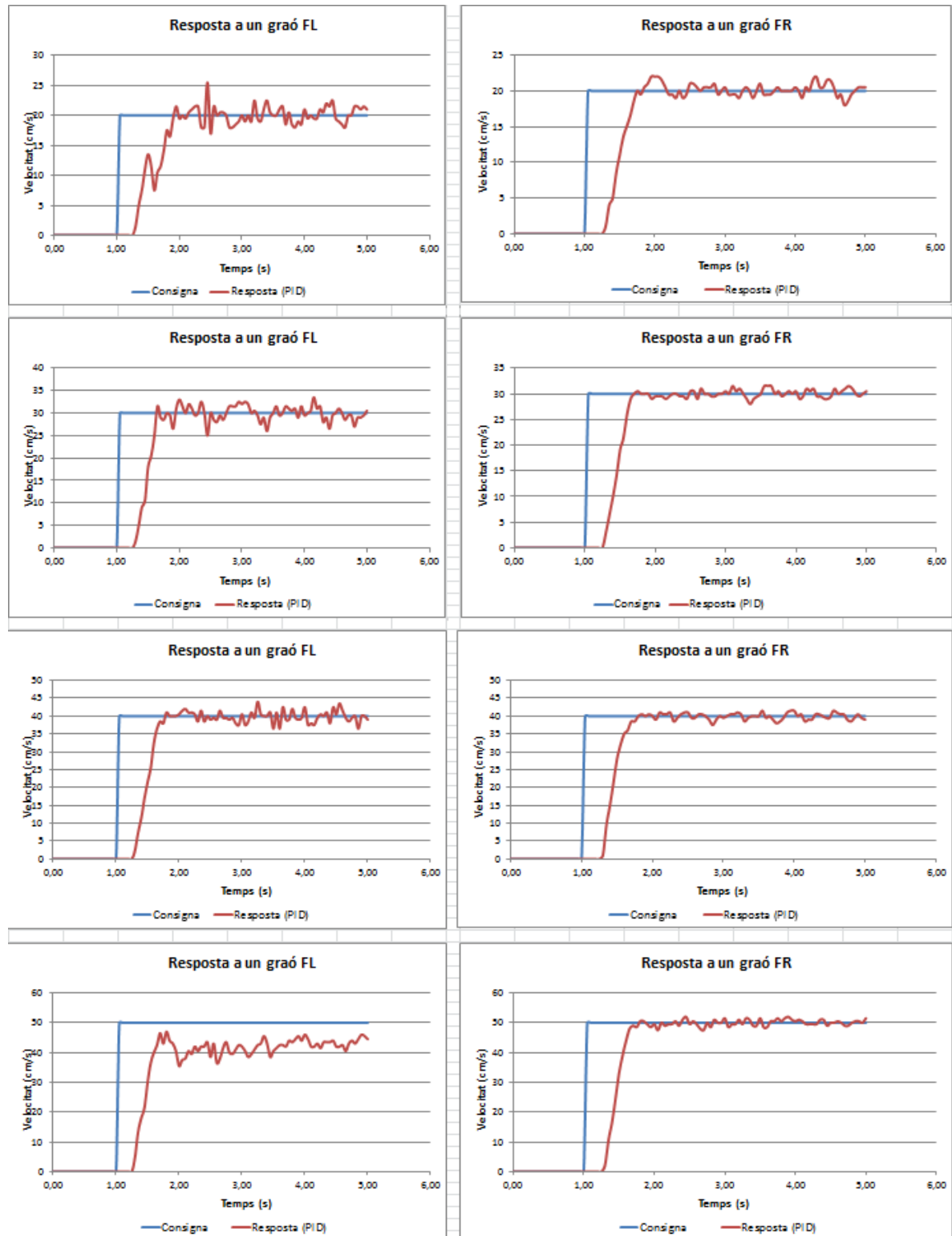


Figura 30. Respostes gir herba rodes de davant

Això es degut a que aquestes dues rodes són les que porten més tracció a l'hora en que el robot gira cap a l'esquerra. Si el robot girés cap a la dreta seria l'altra diagonal (roda dreta de davant i roda esquerra de darrera) quines durien la major tracció.

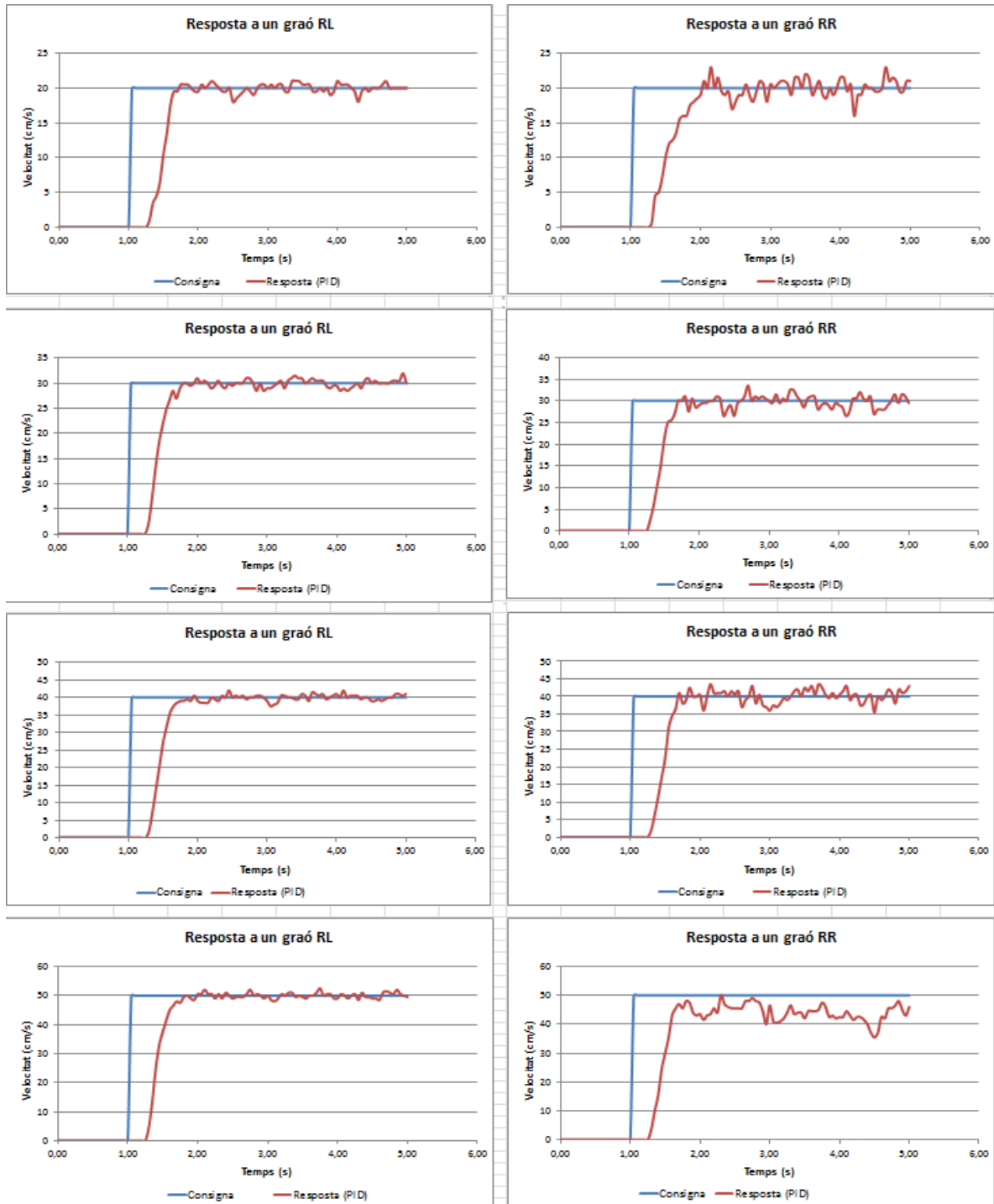


Figura 31. Respostes gir herba rodes de darrera

Per la velocitat de 50 cm/s les dues rodes que porten la major tracció no arriben mai a la consigna desitjada i això és perquè el senyal de control està al màxim i per tant està saturat.

6.7 Mosaic gir

Per aquest terreny passa el mateix que per l'herba, les dues rodes que porten la màxima tracció es saturen a la velocitat de 50 cm/s.

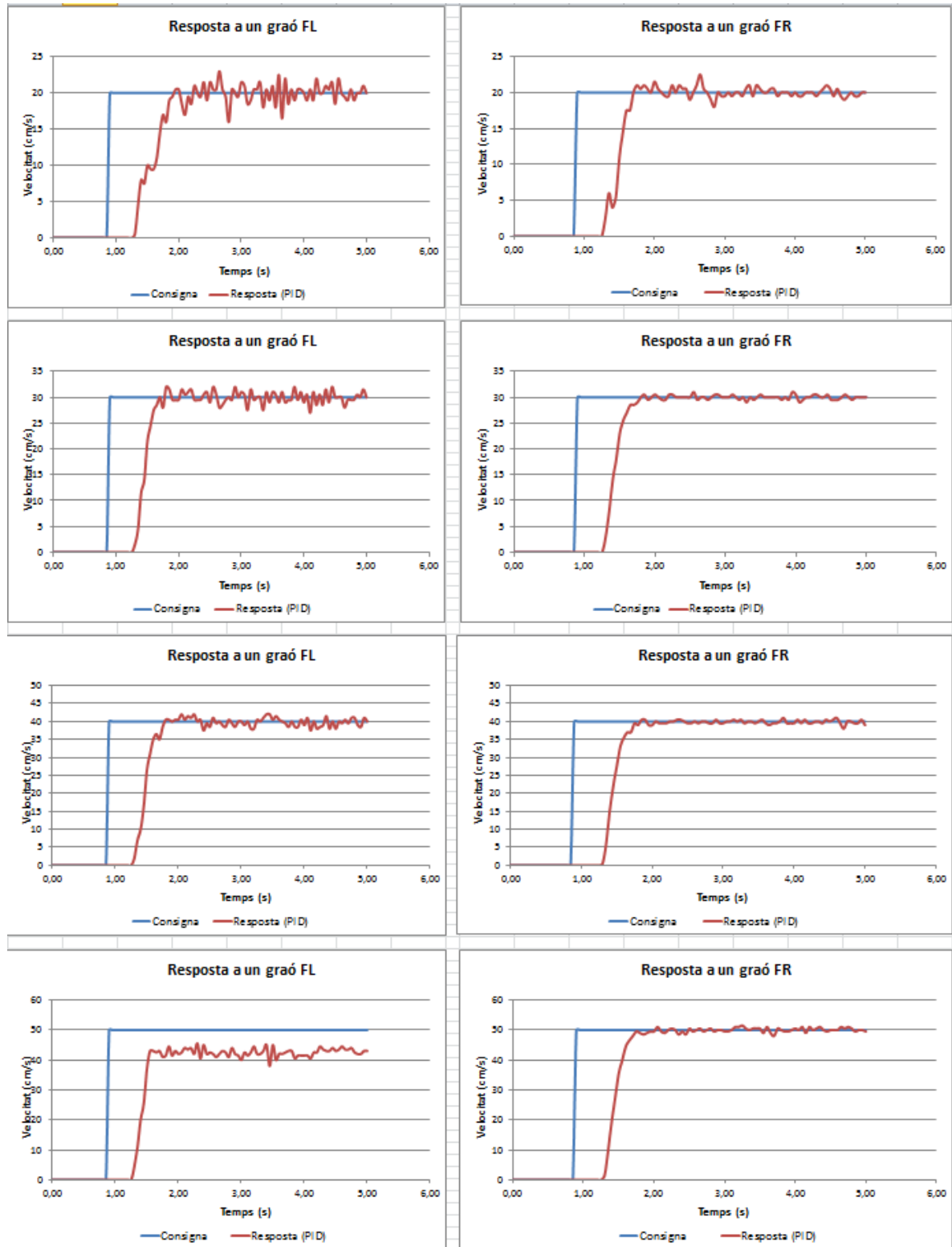


Figura 32. Respostes gir mosaic rodes de davant

Això fa que sigui impossible que el robot pugui arribar a la consigna desitjada. Tot i així, gràficament es pot veure que per les altres velocitats les respostes són bastant correctes i per tant el controlador és adient.

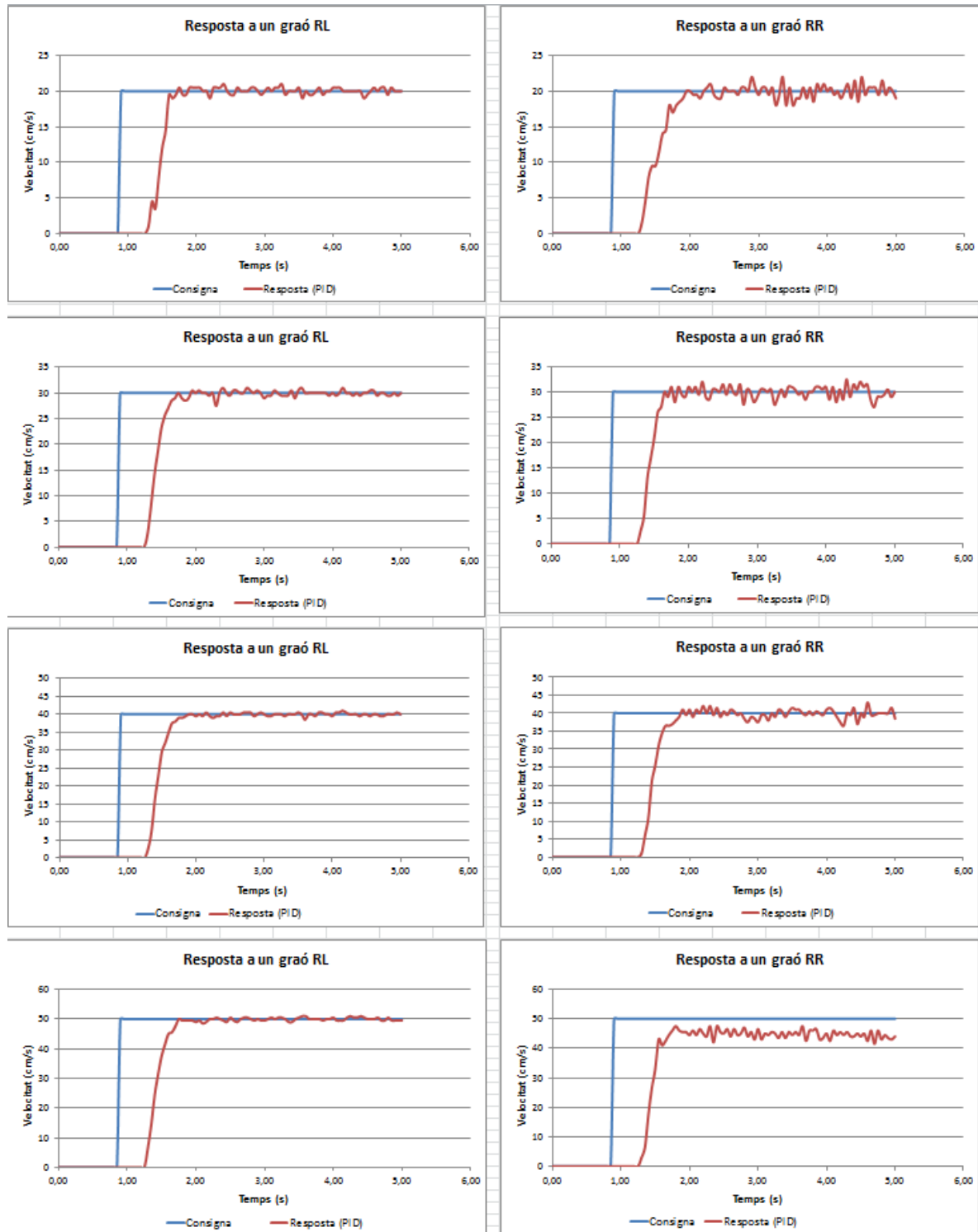


Figura 33. Respostes gir mosaic rodes de darrera

6.8 Sorra gir

En aquest les rodes friccionen molt poc amb el terra pel que fa fàcil el seu moviment de gir.

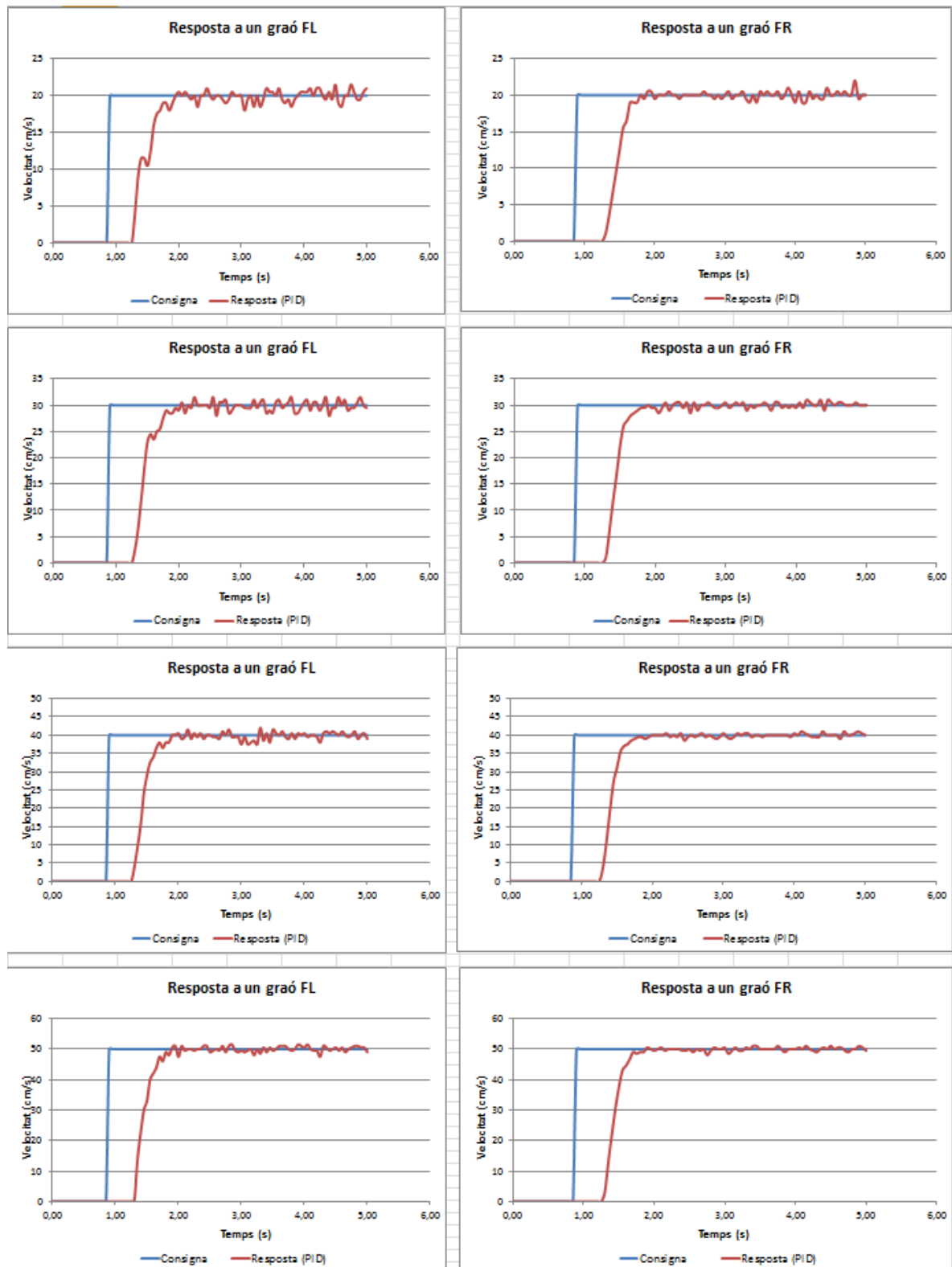


Figura 34. Respostes gir sorra rodes de davant

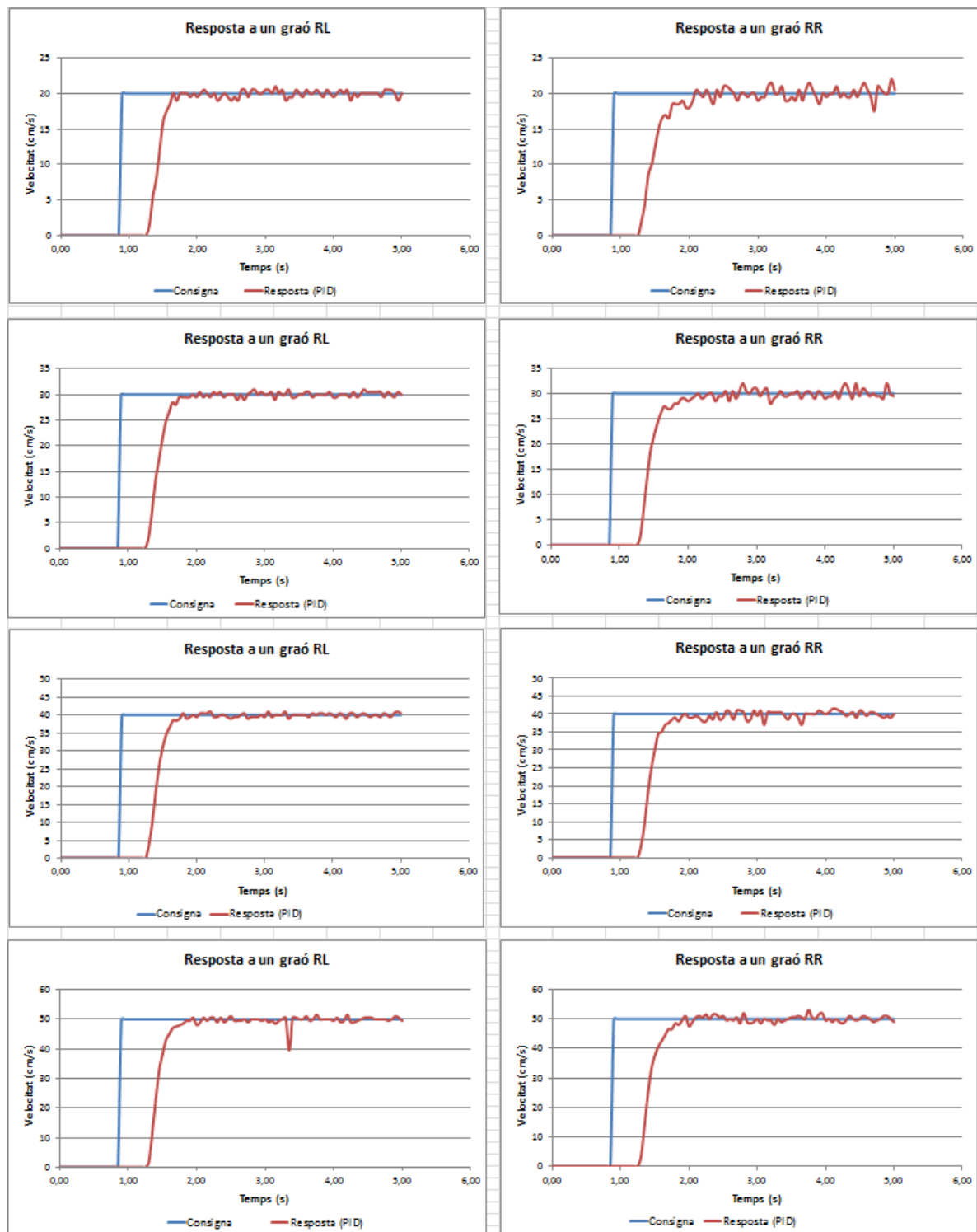


Figura 35. Respostes gir sorra rodes de darrera

Gràficament es pot comprovar que el controlador per aquest terreny garanteix un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3s.

6.9 Paviment recte i gir

El terreny paviment tal com s'ha pogut comprovar a l'hora de la detecció del terreny, la línia de regressió és bastant semblant a la del mosaic. Però tot i així s'ha estudiat perquè les respostes són diferents a la dels altres entorns.

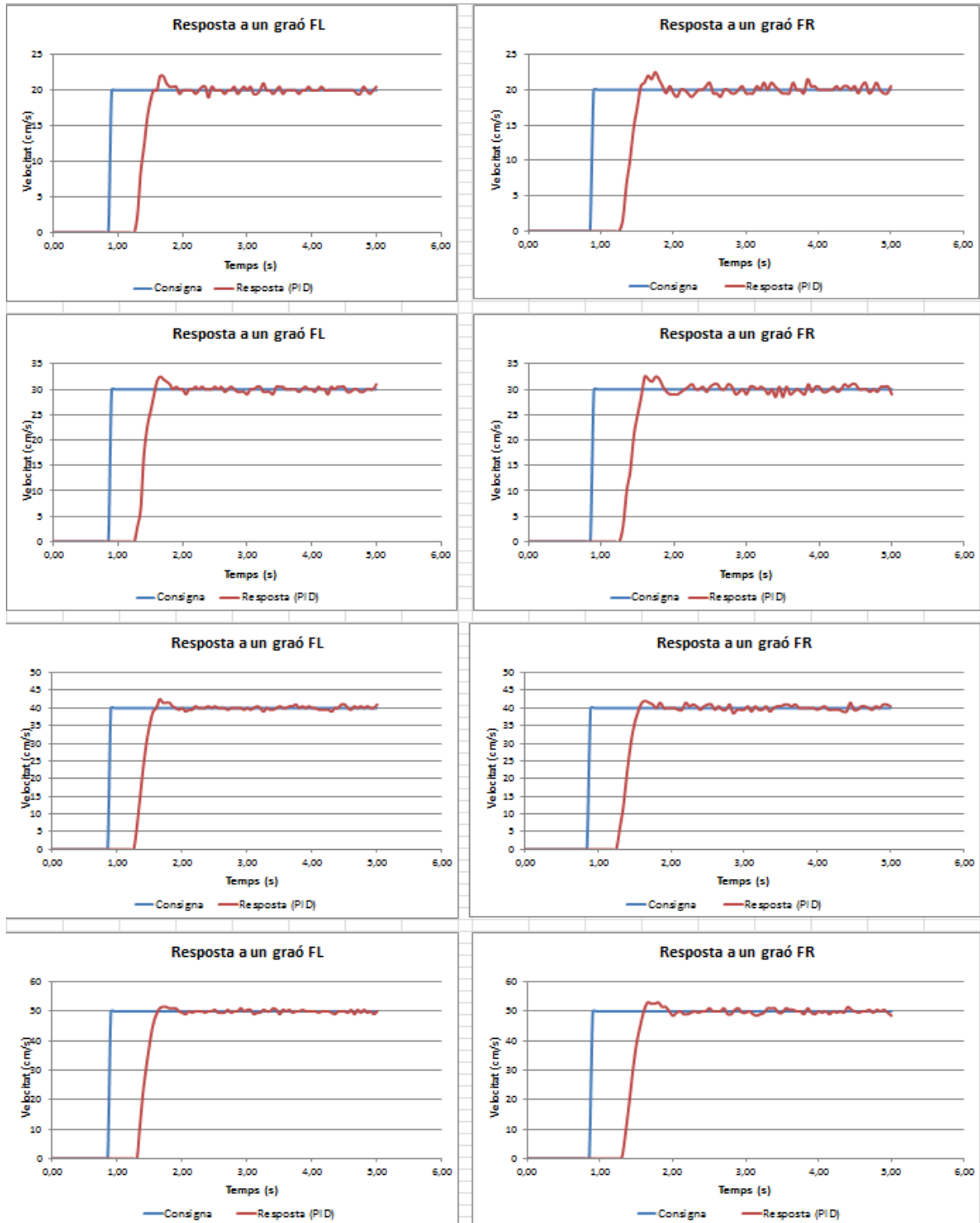


Figura 36. Respostes recte paviment rodes de davant

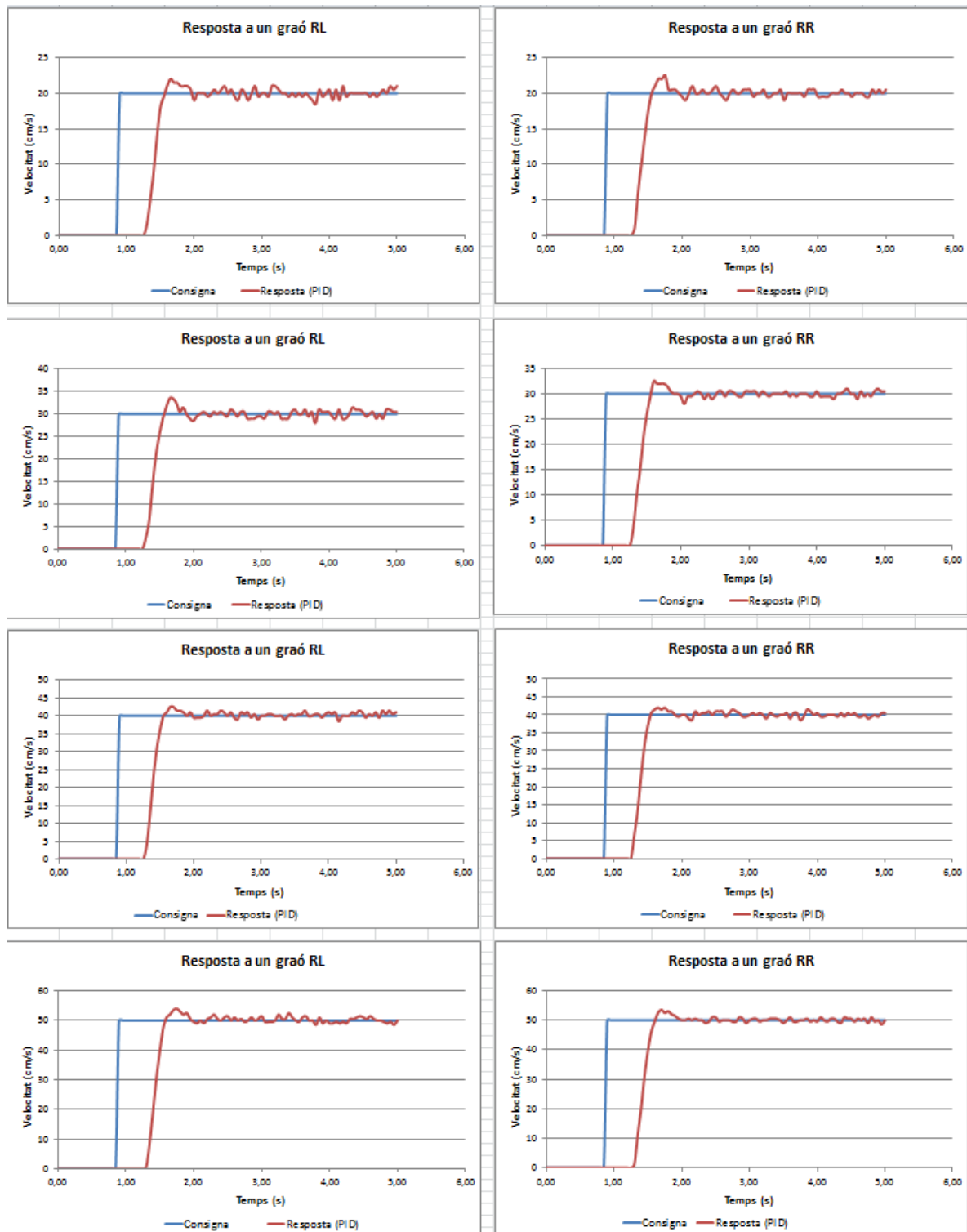


Figura 37. Respostes recte paviment rodes de darrera

Gràficament es pot comprovar que el controlador per aquest terreny garanteix un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3s. Per tant quan el robot funciona anant recte el controlador és adient tot i la rugositat que presenta el terreny.

Tot lo contrari passa quan el robot funciona girant sobre el seu mateix eix, com es pot veure gràficament, hi ha moltes oscil·lacions, sobretot en les dues rodes que aporten la major

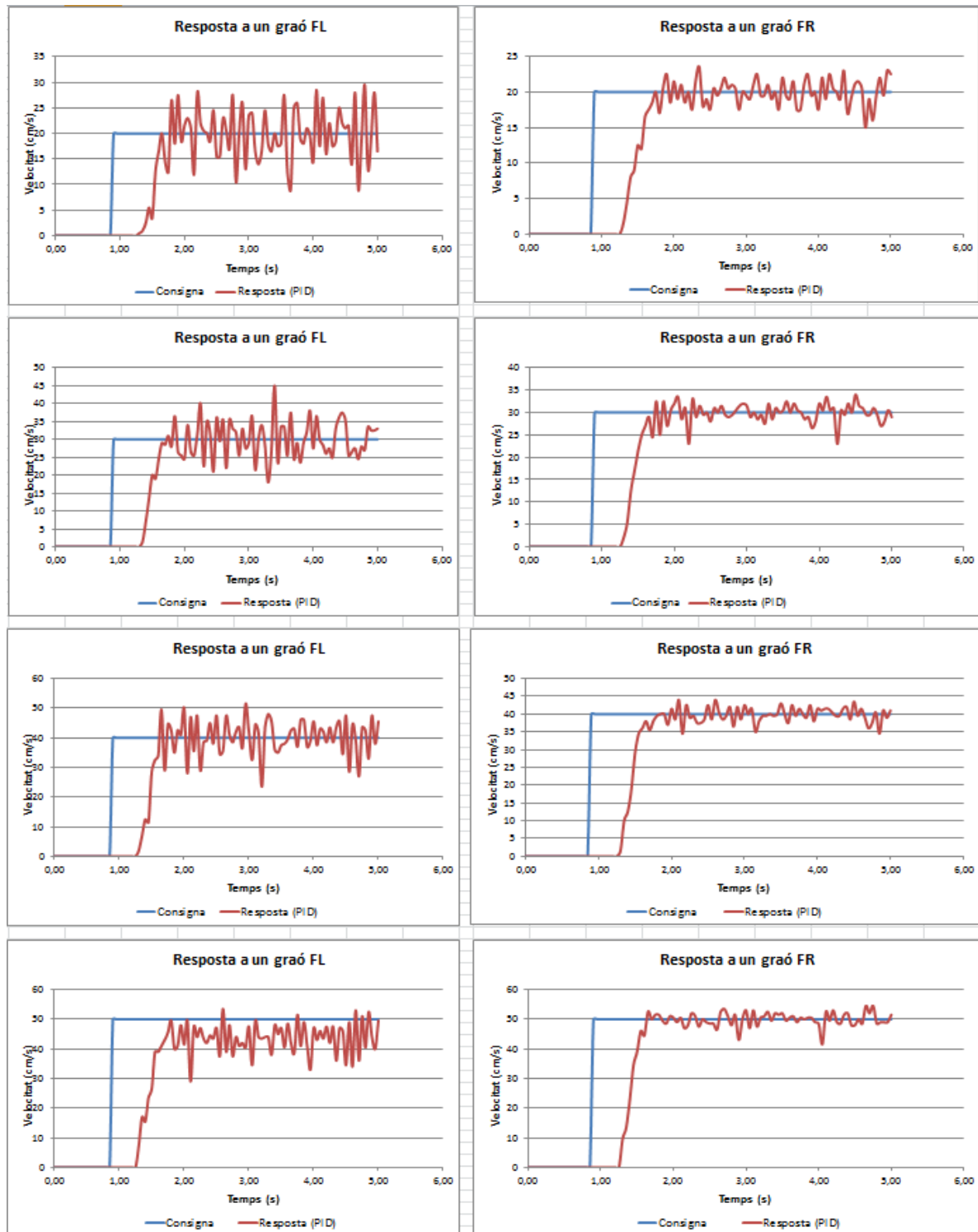


Figura 38. Respostes gir paviment rodes de davant

tracció (roda de davant esquerra i roda de darrera dreta). La resposta de les altres dues rodes és bastant correcte.

També es pot notar quan el robot gira a 50 cm/s, la resposta no arriba a la consigna degut a la saturació, ja que el senyal de control està al 100%.

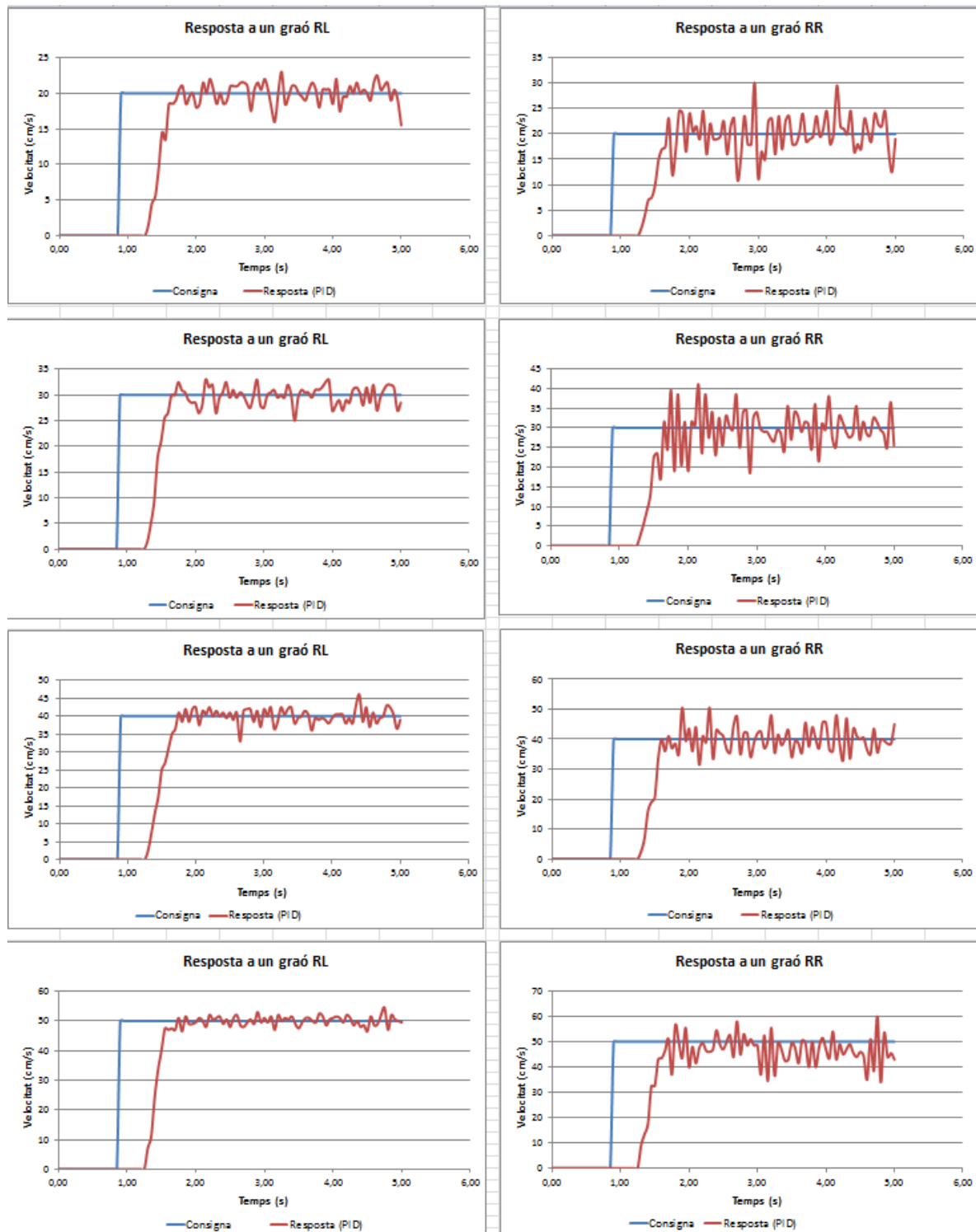


Figura 39. Respostes gir rodes de darrera

S'ha desestimat qualsevol nova sintonització per aquest terreny però s'ha pogut comprovar que el funcionament del robot per aquest terreny no serà òptim.

En la següent figura es pot veure com el senyal de control en aquest terreny arriba al 100% quan està girant a 50 cm/s. La figura mostra la lectura de 2,2 segons de les velocitats de les rodes i els seus respectius senyal de control.

Time	Speed_FL	Speed_FR	Speed_RL	Speed_RR	Ctl_FL	Ctl_FR	Ctl_RL	Ctl_RR
[s]	[m/s]	[m/s]	[m/s]	[m/s]	[%]	[%]	[%]	[%]
22.715.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22.765.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22.815.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22.865.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22.915.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22.965.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.015.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.065.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.115.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.165.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.215.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.265.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.315.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.365.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.415.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.465.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.515.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.565.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.615.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.665.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.715.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.765.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.815.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.865.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23.915.000	0.000	0.000	0.000	0.000	-7.000	7.000	-7.000	7.000
23.965.000	-10.000	10.000	-6.000	10.000	-30.000	30.000	-31.000	30.000
24.015.000	-21.000	13.000	-14.000	14.000	-47.000	49.000	-51.000	49.000
24.065.000	-15.000	23.000	-27.000	21.000	-62.000	66.000	-67.000	66.000
24.115.000	-16.000	36.000	-34.000	34.000	-80.000	77.000	-75.000	78.000
24.165.000	-25.000	39.000	-42.000	36.000	-97.000	83.000	-80.000	84.000
24.215.000	-40.000	48.000	-46.000	44.000	-100.000	87.000	-84.000	92.000
24.265.000	-36.000	44.000	-46.000	35.000	-100.000	90.000	-85.000	94.000
24.315.000	-53.000	56.000	-51.000	56.000	-100.000	92.000	-86.000	100.000
24.366.000	-36.000	47.000	-44.000	36.000	-100.000	92.000	-87.000	100.000
24.415.000	-48.000	54.000	-52.000	32.000	-100.000	93.000	-88.000	100.000
24.465.000	-46.000	54.000	-44.000	60.000	-100.000	90.000	-88.000	100.000
24.515.000	-47.000	50.000	-49.000	53.000	-100.000	90.000	-90.000	100.000
24.565.000	-38.000	45.000	-47.000	44.000	-100.000	89.000	-90.000	100.000
24.615.000	-47.000	52.000	-51.000	58.000	-100.000	91.000	-91.000	100.000
24.666.000	-40.000	52.000	-50.000	44.000	-100.000	90.000	-90.000	100.000
24.715.000	-49.000	47.000	-49.000	48.000	-100.000	91.000	-90.000	100.000
24.765.000	-41.000	51.000	-52.000	43.000	-100.000	93.000	-91.000	100.000
24.815.000	-57.000	50.000	-46.000	36.000	-100.000	92.000	-90.000	100.000
24.865.000	-42.000	51.000	-53.000	56.000	-100.000	92.000	-91.000	100.000

Figura 40. Lectura variables en gir 50 cm/s en paviment

Es nota doncs, que les rodes que porten la major tracció són les que consumeixen més intensitat i per contra ja no arriben a la consigna marcada. També es pot veure en les columnes que ens marca els valors de les velocitats, on es nota que la velocitat no arriba a la consigna.

6.10 Interpretació en recte

Un cop estudiat els diferents entorns proposats pel present projectes, gràficament, es pot dir que el controlador PI que porta incorporat el robot és adient per a tots ells i no cal cap nova sintonització.

L'estudi ha estat important per saber que el funcionament del robot en els altres terrenys és prou semblant tot i la diferència entre aquests per la regularitat.

6.11 Interpretació en gir

Els assajos en el gir sobre el mateix eix del robot ja han presentat una variació important ja que s'ha de tenir en compte la fricció i la tracció de les rodes.

Pel que fa al terra rajola i sorra les respostes del robot mòbil són correctes però la problemàtica s'ha vist en el terreny mosaic i herba on per 50 cm/s no s'arriba mai a la consigna desitjada en les rodes que porten la tracció. A més a més, aquestes dues rodes (roda esquerra de davant i roda dreta de darrera) per les altres velocitats, presenten unes respostes amb moltes més oscil·lacions que les altres dues rodes.

S'ha desestimat qualsevol sintonització PID nova per les dues rodes ja que el controlador per aquests terrenys en el gir garantitza un sobrepic de menys del 25%, un error estacionari de menys del 5% i un temps d'establiment de menys de 3s tot i les pertorbacions que puguin haver-hi.

Tot i que el robot a 50 cm/s en mosaic i herba, en les rodes que traccionen més no arribi mai a la consigna, no desestimarem mai aquesta velocitat, però si cal tenir en compte que el gir no serà òptim.

Per aquest motiu s'ha decidit millorar el control de posició en l'orientació del robot, ja que en el gir, és on el robot depenent de la velocitat, tindrà una orientació òptima o no.

7 CONTROL DE POSICIÓ

La millora en el moviment del Bigbot és essencial per una major maniobrabilitat. Un cop fet l'estudi sobre la comprovació del control de velocitat PID no hem cregut oportú la millora del control de posició amb el robot funcionant en recte, ja que l'error és molt petit. Per altra banda hem determinat que la millora en l'orientació és important, primer per l'error que genera i segon perquè en molts casos el gir amb el control de velocitat PID no és òptim.

Vist en la bibliografia, Anthony Mandow i Wei Yu proposen unes tècniques que fan servir per millorar el gir, utilitzant el càlculs d'odometria.

Obvien la millora en el desplaçament recte, ja que es produeix un error molt baix, això és pot notar amb l'equació plantejada, ja que simplement es fa una mitja sobre el desplaçament recorregut per cada roda.

$$d = \frac{d_R - d_L}{2} \quad (\text{Eq.2})$$

És convenient explicar primer els càlculs d'odometria per a un robot de tipus rotació pura sense lliscament, anomenats differential drive. Són robots amb bandes de rotació d'esquerra i dreta en el pla, és a dir, amb dues rodes, una per cada banda.

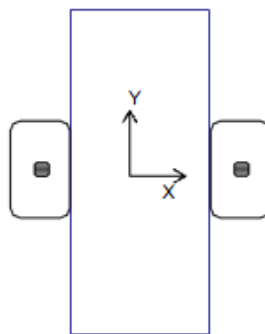


Figura 41. Vehicle de tipologia differential drive

El direccionalment ve donat per la diferència de velocitats de les rodes laterals. La tracció s'aconsegueix també amb aquestes mateixes rodes. Aquest sistema és molt útil presentant la possibilitat de canviar d'orientació sense moviment de translació. Les variables de control d'aquest sistema són les velocitats angulars de les rodes esquerra i dreta.

El procés experimental sobre aquests tipus de robots respon a la fórmula següent:

$$\varnothing = \frac{d_R - d_L}{B} \quad (\text{Eq.3})$$

on d_R és el desplaçament de la banda dreta, d_L és el desplaçament de la banda esquerra, B la distància entre les rodes de cada banda i \varnothing l'angle de rotació en radians.

El càlcul del gir en radians es pot entendre senzillament amb la lectura que fan els encoders a les rodes del robot. Agafarem l'exemple de la roda dreta, la qual seguint l'equació següent s'obté el desplaçament total.

$$\Delta dr = \frac{N_{int}}{N_{volta}} \times 2\pi r \quad (\text{Eq.4})$$

$N_{interval}$ són els polsos de la roda dreta, N_{volta} són els polsos totals en un gir i $2\pi r$ és el perímetre de la roda. Aquest és el càlcul que fan els encoders que cada motor porta acoblar al seu eix. D'aquesta manera es pot determinar el desplaçament de cada roda.

Tot i que aquest sigui el càlcul que es produeix a les rodes gràcies als encoders, a l'hora de la lectura, el robot genera un document informatiu on no s'especifica el desplaçament recorregut, per això pel càlcul del desplaçament s'ha d'agafar la lectura de les velocitats per cada interval de temps de 50 ms.

$$d_{int} = v_{int} * \Delta t \quad (\text{Eq.5})$$

Per tant el desplaçament total serà la suma dels desplaçament de cada interval corresponent a l'equació 5.

Aquest apartat del present projecte té com a objectiu l'estudi teòric per la millora en temps real del control tenint en compte els efectes del desplaçament. Així com pels robots de rotació pura no cal diferenciar el terreny, pel tipus de robot amb el qual treballem pel projecte es necessari, mitjançant la millora de l'equació 3 proposada per Anthony Mandow i Wei Yu amb les seves tècniques experimentals.

Segons els autors esmentats abans, aquest estudi tracta sobre la millora del moviment en rotació del robot depenent del terreny. Per realitzar la millora, es buscarà el factor de correcció pel control d'orientació que s'escau a cadascun dels entorns proposats al projecte.

Un cop fet l'estudi per veure si calia una nova sintonització pel control de velocitat, el següent pas és mirar quin factor de correcció cal pel control de posició quan el robot gira ja que com hem pogut comprovar abans, hi ha molta oscil·lacions a les rodes que tenen més tracció i que per 50 cm/s per alguns terrenys no s'arriba a la consigna. Per tant, al no aconseguir un gir òptim, serà interessant el control de l'orientació del robot.

Aquest estudi es basa en el control de les velocitats relatives de les unitats del costat esquerra i dret.

El robot del qual disposem és de tipus “skid-steer”, ja que es disposen vàries rodes a cada costat del vehicle (en aquest cas dues rodes per banda) i actuen de forma simultània. El moviment és el resultat de combinar les velocitats de les rodes de l'esquerra amb les de la dreta.

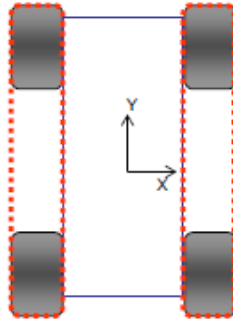


Figura 42. Vehicle de tipologia skid-steer

El procediment experimental per trobar el factor de correcció s'ha regit sota l'equació 6, una millora de l'equació 3.

$$\varnothing = \frac{d_R - d_L}{\alpha * B} \quad (\text{Eq.6})$$

on d és el desplaçament, \varnothing l'angle de rotació en radians, B la distància entre les rodes de cada banda i α el factor de correcció a trobar per cada terreny.

Els assajos fets per cada terreny ha estat un gir de 90° amb les diferents velocitats de treball (20 cm/s, 30 cm/s, 40 cm/s i 50 cm/s).

Un cop fet els assajos, amb un arxiu Excel es calculen les distàncies recorregudes per cada roda. Com que el robot és de tipus “skid-steer” amb dues rodes per banda s'ha fet una mitjana combinant les distàncies recorregudes de les rodes esquerres i el mateix procediment s'ha fet per les rodes de la dreta.

Per fer el càlcul de les distàncies s'ha aprofitat la lectura dels encoders de les rodes. Seguint l'equació següent s'obté el desplaçament de la roda.

$$d = v * \Delta t \quad (\text{Eq.7})$$

Tenint en compte que la lectura dels encoders es produeix amb un temps de mostreig de 50 ms, s'ha de sumar la mitja del desplaçament de les dues rodes de cada banda realitzat per a cada lectura fins que el robot hagi aconseguit el gir complet de 90°.

$$dt = \frac{v_{troda\ davant} + v_{troda\ darrera}}{2} * t_{mostreig} \quad (\text{Eq.8})$$

Un cop obtingut el desplaçament total fet per cada roda, aplicant l'equació 2 s'obté el factor de correcció que s'ha d'aplicar a cada terreny.

Ground / Speed [cm/s]	20	30	40	50
Rajola	1.94	1.99	2.03	2.23
Herba	1.88	1.93	2.01	2.21
Mosaic	1.99	2.01	2.04	-
Sorra	2.72	2.69	2.73	2.88
Paviment	1.97	1.94	2.38	-

Taula 5. Factors de correcció per terreny i velocitat

Aquesta part experimental s'ha fet igual a l'hora d'agafar les lectures que en l'apartat 4 en la comprovació del controlador PID. Els assajos han consistit en fer girar el robot 90° a les diferents velocitats, agafant la lectura de les diferents velocitats en tot el temps en el que el robot ha tardat en fer el gir. Un cop obtinguda aquesta lectura s'ha procedit a fer el càlcul de la distància total recorreguda per cada banda del robot (banda esquerra i banda dreta). Després d'obtenir els diferents desplaçaments, fent ús de la equació 2, s'han pogut trobar els diferents factors de correcció α reflectits en la taula 5.

8 MILLORES I TREBALL FUTUR

Les tècniques i assajos estudiats durant la memòria han permès assolir una crítica de cara a una futura millora per obtenir un millor rendiment. Aquest capítol es centra a recollir algunes possibles millores que es deixen obertes per a un futur treball.

8.1 Millorar l'estudi

Si l'estudi només tracta de cinc terrenys, la millora passa per ampliar aquest estudi i veure nous entorns. Així doncs, abastar aquest estudi, implicaria tenir més informació adient per a la millora del moviment de l'agent mòbil. Els terrenys on es poden fer més assajos poden ser la pedra, herba humida, fusta, asfalt, etc. Amb aquests assajos s'acabaria de determinar si el PID incorporat en el robot és adient per a tots el terrenys o s'hauria de fer una nova sintonia, sempre tenint en compte els paràmetres estàndards marcats des d'un bon inici.

8.2 Millorar l'aplicació

Aquesta millora passa per dues parts, la primera i més fàcil és abastar la detecció del terreny en molts més terres, fent els assajos que s'escaiguin. Així la gràfica de la regressió lineal de les velocitats en funció de la potència seria més ampla.

La segona millora seria ja enfocar aquesta programació en el dsPIC33FJ256MC que incorpora el robot. D'aquesta manera, amb les dades de la regressió lineal feta amb MATLAB, es podria fer una aplicació on el robot, en mode online, pugui anar llegint les dades i detectant el terreny mentre estigui funcionant. Aquesta aplicació es podria fer amb MPLAB que és el programari amb el que està programat el robot.

9 RESUM DEL PRESSUPOST

El cost total per a la realització d'aquest projecte és de cinc mil tres-cents vint euros sense comptar l'IVA.

10 CONCLUSIONS

Un cop acabat aquest projecte es pot concloure que hem estat capaços d'assolir els objectius proposats al principi. S'ha aconseguit desenvolupar una aplicació utilitzant MATLAB. El programa desenvolupat permetrà detectar el terra per on el robot s'està movent en mode online. Així doncs amb la lectura de les velocitats en funció de les lectures de la potència i, fent una regressió lineal, es pot diferenciar els terrenys que s'han proposat per estudiar.

S'han fet els estudis pertinents als terrenys escollits. Primerament estudiant les respostes d'aquests amb el robot funcionant en recte i girant, per tal de determinar si calia una nova sintonització o no.

Al determinar que el gir no era òptim en molts dels casos, s'ha optat per estudiar el control d'orientació del robot, amb el qual s'han aconseguit els factors de correcció adients per a la seva millora en l'orientació.

A partir d'aquest projecte de final de grau es podran començar a estudiar més terrenys amb un treball directe, ja que les eines necessàries, els arxius Excel d'estudi, s'han pogut crear correctament.

Julián Alberto García Vélez
Graduat en Enginyeria Electrònica Industrial i Automàtica

Girona, 10 de juny de 2014

11 RELACIÓ DE DOCUMENTS

Aquest projecte consta de quatre documents: memòria, plec de condicions, estat d'amidaments i pressupost ordenats respectivament.

12 BIBLIOGRAFIA

Aníbal Ollero Baturone, Robótica. Manipuladores y robots móviles. (<http://books.google.es/books?id=TtMfuy6FNCcC&pg=PA31&dq=%C2%BFqu%C3%A9+son+los+robots+%22skid-steer%22?&hl=es&sa=X&ei=9IN8U6m2NenX0QWG2IGADQ&ved=0CEIQ6AEwAA#v=onepage&q=%C2%BFqu%C3%A9%20son%20los%20robots%20%22skid-steer%22%3F&f=false>) , 28 d'abril de 2014)

Anthony Mandow, Jorge L. Martínez, Jesús Morales, José L. Blanco, Alfonso García-Cerezo and Javier González, "Experimental kinematics for wheeled skid-steer mobile robots", Dpto. Ingeniería de Sistemas y Automática, Universidad de Málaga.

Dpto. Ingeniería Mecánica. Ajuste empírico de controladores PID método de Ziegler-Nichols.

Edgar Romero Rodríguez, MÉTODOS NUMÉRICOS. Febrero 2007.

Javier García de Jalón, José Ignacio Rodríguez, Jesús Vidal, Aprende Matlab 7.0 como si estuviera en primero. Madrid. Diciembre 2005.

Wei Yu, Oscar Chuy Jr., Emmanuel G. Collins Jr., and Patrick Hollis, "Dynamic Modeling of a Skid-Steered Wheeled Vehicle with Experimental Verification" In International Conference on Intelligent Robots and Systems, October 11-15, 2009 St. Louis, USA.

13 GLOSSARI

C : Llenguatge de programació

dsPIC : Digital Signal Peripheral Interface Controller

GPS : Global Positioning System

PID : Proporcional Integral Derivatiu

PWM : Pulse With Modulation

USB: Universal Serial Bus

A PROGRAMA DETECCIÓ DE TERRA

```

close all;
%clear all;
%Dades Rajola
Imp=importdata('BB1_20_16_PIDFre20%Rajola.txt');
Data=Imp.data;
Imp2=importdata('BB1_30_24_PIDFre20%Rajola.txt');
Data2=Imp2.data;
Imp3=importdata('BB1_40_32_PIDFre20%Rajola.txt');
Data3=Imp3.data;
Imp4=importdata('BB1_50_40_PIDFre20%Rajola.txt');
Data4=Imp4.data;
%Dades Mosaic
Imp6=importdata('BB1_20_16_PIDFre20%Mosaic.txt');
Data6=Imp6.data;
Imp7=importdata('BB1_30_24_PIDFre20%Mosaic.txt');
Data7=Imp7.data;
Imp8=importdata('BB1_40_32_PIDFre20%Mosaic.txt');
Data8=Imp8.data;
Imp9=importdata('BB1_50_40_PIDFre20%Mosaic.txt');
Data9=Imp9.data;
%Dades Herba
Imp11=importdata('BB1_20_16_PIDFre20%Herba.txt');
Data11=Imp11.data;
Imp12=importdata('BB1_30_24_PIDFre20%Herba.txt');
Data12=Imp12.data;
Imp13=importdata('BB1_40_32_PIDFre20%Herba.txt');
Data13=Imp13.data;
Imp14=importdata('BB1_50_40_PIDFre20%Herba.txt');
Data14=Imp14.data;
%Dades Sorra
Imp16=importdata('BB1_20_16_PIDFre20%Sorra.txt');
Data16=Imp16.data;
Imp17=importdata('BB1_30_24_PIDFre20%Sorra.txt');
Data17=Imp17.data;
Imp18=importdata('BB1_40_32_PIDFre20%Sorra.txt');
Data18=Imp18.data;
Imp19=importdata('BB1_50_40_PIDFre20%Sorra.txt');
Data19=Imp19.data;

%% Separate all variables Rajola data1
speed_FL = Data(101:200,2);
curr_FL = Data(101:200,6);
speed_FR = Data(101:200,3);
curr_FR = Data(101:200,7);
battery = Data(101:200,13);
ctl_FL = Data(101:200,20);
ctl_FR = Data(101:200,21);

%% Separate all variables Rajola data2
speed_FL2 = Data2(101:200,2);
curr_FL2 = Data2(101:200,6);
speed_FR2 = Data2(101:200,3);
curr_FR2 = Data2(101:200,7);
battery2 = Data2(101:200,13);
ctl_FL2 = Data2(101:200,20);
ctl_FR2 = Data2(101:200,21);

%% Separate all variables Rajola data3
speed_FL3 = Data3(101:200,2);
curr_FL3 = Data3(101:200,6);
speed_FR3 = Data3(101:200,3);
curr_FR3 = Data3(101:200,7);
battery3 = Data3(101:200,13);
ctl_FL3 = Data3(101:200,20);

```

```
ctl_FR3      =   Data3(101:200,21);

%% Separate all variables Rajola data4
speed_FL4    =   Data4(101:200,2);
curr_FL4     =   Data4(101:200,6);
speed_FR4    =   Data4(101:200,3);
curr_FR4     =   Data4(101:200,7);
battery4     =   Data4(101:200,13);
ctl_FL4     =   Data4(101:200,20);
ctl_FR4     =   Data4(101:200,21);

%% Separate all variables Mosaic data6
speed_FL6    =   Data6(101:200,2);
curr_FL6     =   Data6(101:200,6);
speed_FR6    =   Data6(101:200,3);
curr_FR6     =   Data6(101:200,7);
battery6     =   Data6(101:200,13);
ctl_FL6     =   Data6(101:200,20);
ctl_FR6     =   Data6(101:200,21);

%% Separate all variables Mosaic data7
speed_FL7    =   Data7(101:200,2);
curr_FL7     =   Data7(101:200,6);
speed_FR7    =   Data7(101:200,3);
curr_FR7     =   Data7(101:200,7);
battery7     =   Data7(101:200,13);
ctl_FL7     =   Data7(101:200,20);
ctl_FR7     =   Data7(101:200,21);

%% Separate all variables Mosaic data8
speed_FL8    =   Data8(101:200,2);
curr_FL8     =   Data8(101:200,6);
speed_FR8    =   Data8(101:200,3);
curr_FR8     =   Data8(101:200,7);
battery8     =   Data8(101:200,13);
ctl_FL8     =   Data8(101:200,20);
ctl_FR8     =   Data8(101:200,21);

%% Separate all variables Mosaic data9
speed_FL9    =   Data9(101:200,2);
curr_FL9     =   Data9(101:200,6);
speed_FR9    =   Data9(101:200,3);
curr_FR9     =   Data9(101:200,7);
battery9     =   Data9(101:200,13);
ctl_FL9     =   Data9(101:200,20);
ctl_FR9     =   Data9(101:200,21);

%% Separate all variables Herba data 11
speed_FL11   =   Data11(101:200,2);
curr_FL11    =   Data11(101:200,6);
speed_FR11   =   Data11(101:200,3);
curr_FR11    =   Data11(101:200,7);
battery11    =   Data11(101:200,13);
ctl_FL11    =   Data11(101:200,20);
ctl_FR11    =   Data11(101:200,21);

%% Separate all variables Herba data 12
speed_FL12   =   Data12(101:200,2);
curr_FL12    =   Data12(101:200,6);
speed_FR12   =   Data12(101:200,3);
curr_FR12    =   Data12(101:200,7);
battery12    =   Data12(101:200,13);
ctl_FL12    =   Data12(101:200,20);
ctl_FR12    =   Data12(101:200,21);

%% Separate all variables Herba data 13
speed_FL13   =   Data13(101:200,2);
```

```

curr_FL13      =   Data13(101:200,6);
speed_FR13    =   Data13(101:200,3);
curr_FR13     =   Data13(101:200,7);
battery13     =   Data13(101:200,13);
ctl_FL13      =   Data13(101:200,20);
ctl_FR13      =   Data13(101:200,21);

%% Separate all variables Herba data 14
speed_FL14    =   Data14(101:200,2);
curr_FL14     =   Data14(101:200,6);
speed_FR14    =   Data14(101:200,3);
curr_FR14     =   Data14(101:200,7);
battery14     =   Data14(101:200,13);
ctl_FL14      =   Data14(101:200,20);
ctl_FR14      =   Data14(101:200,21);

%% Separate all variables Sorra data16
speed_FL16    =   Data16(101:200,2);
curr_FL16     =   Data16(101:200,6);
speed_FR16    =   Data16(101:200,3);
curr_FR16     =   Data16(101:200,7);
battery16     =   Data16(101:200,13);
ctl_FL16      =   Data16(101:200,20);
ctl_FR16      =   Data16(101:200,21);

%% Separate all variables Sorra data17
speed_FL17    =   Data17(101:200,2);
curr_FL17     =   Data17(101:200,6);
speed_FR17    =   Data17(101:200,3);
curr_FR17     =   Data17(101:200,7);
battery17     =   Data17(101:200,13);
ctl_FL17      =   Data17(101:200,20);
ctl_FR17      =   Data17(101:200,21);

%% Separate all variables Sorra data18
speed_FL18    =   Data18(101:200,2);
curr_FL18     =   Data18(101:200,6);
speed_FR18    =   Data18(101:200,3);
curr_FR18     =   Data18(101:200,7);
battery18     =   Data18(101:200,13);
ctl_FL18      =   Data18(101:200,20);
ctl_FR18      =   Data18(101:200,21);

%% Separate all variables Sorra data19
speed_FL19    =   Data19(101:200,2);
curr_FL19     =   Data19(101:200,6);
speed_FR19    =   Data19(101:200,3);
curr_FR19     =   Data19(101:200,7);
battery19     =   Data19(101:200,13);
ctl_FL19      =   Data19(101:200,20);
ctl_FR19      =   Data19(101:200,21);

for z=1:100
    %%Vectors de velocitats FL
    AllspeedsFL(z)=speed_FL(z);
    AllspeedsFL(z+100)=speed_FL2(z);
    AllspeedsFL(z+200)=speed_FL3(z);
    AllspeedsFL(z+300)=speed_FL4(z);

    Allspeeds2FL(z)=speed_FL6(z);
    Allspeeds2FL(z+100)=speed_FL7(z);
    Allspeeds2FL(z+200)=speed_FL8(z);
    Allspeeds2FL(z+300)=speed_FL9(z);

    Allspeeds3FL(z)=speed_FL11(z);
    Allspeeds3FL(z+100)=speed_FL12(z);

```

```
Allspeeds3FL(z+200)=speed_FL13(z);
Allspeeds3FL(z+300)=speed_FL14(z);

Allspeeds4FL(z)=speed_FL16(z);
Allspeeds4FL(z+100)=speed_FL17(z);
Allspeeds4FL(z+200)=speed_FL18(z);
Allspeeds4FL(z+300)=speed_FL19(z);

%%Vectors de velocitats FR
AllspeedsFR(z)=speed_FR(z);
AllspeedsFR(z+100)=speed_FR2(z);
AllspeedsFR(z+200)=speed_FR3(z);
AllspeedsFR(z+300)=speed_FR4(z);

Allspeeds2FR(z)=speed_FR6(z);
Allspeeds2FR(z+100)=speed_FR7(z);
Allspeeds2FR(z+200)=speed_FR8(z);
Allspeeds2FR(z+300)=speed_FR9(z);

Allspeeds3FR(z)=speed_FR11(z);
Allspeeds3FR(z+100)=speed_FR12(z);
Allspeeds3FR(z+200)=speed_FR13(z);
Allspeeds3FR(z+300)=speed_FR14(z);

Allspeeds4FR(z)=speed_FR16(z);
Allspeeds4FR(z+100)=speed_FR17(z);
Allspeeds4FR(z+200)=speed_FR18(z);
Allspeeds4FR(z+300)=speed_FR19(z);

%%Vectors de bateria
Allbattery(z)=battery(z);
Allbattery(z+100)=battery2(z);
Allbattery(z+200)=battery3(z);
Allbattery(z+300)=battery4(z);

Allbattery2(z)=battery6(z);
Allbattery2(z+100)=battery7(z);
Allbattery2(z+200)=battery8(z);
Allbattery2(z+300)=battery9(z);

Allbattery3(z)=battery11(z);
Allbattery3(z+100)=battery12(z);
Allbattery3(z+200)=battery13(z);
Allbattery3(z+300)=battery14(z);

Allbattery4(z)=battery16(z);
Allbattery4(z+100)=battery17(z);
Allbattery4(z+200)=battery18(z);
Allbattery4(z+300)=battery19(z);

%%Vectors de senyal de control FL
AllctlFL(z)=ctl_FL(z);
AllctlFL(z+100)=ctl_FL2(z);
AllctlFL(z+200)=ctl_FL3(z);
AllctlFL(z+300)=ctl_FL4(z);

Allctl2FL(z)=ctl_FL6(z);
Allctl2FL(z+100)=ctl_FL7(z);
Allctl2FL(z+200)=ctl_FL8(z);
Allctl2FL(z+300)=ctl_FL9(z);

Allctl3FL(z)=ctl_FL11(z);
Allctl3FL(z+100)=ctl_FL12(z);
Allctl3FL(z+200)=ctl_FL13(z);
Allctl3FL(z+300)=ctl_FL14(z);

Allctl4FL(z)=ctl_FL16(z);
```

```
Allctl4FL(z+100)=ctl_FL17(z);
Allctl4FL(z+200)=ctl_FL18(z);
Allctl4FL(z+300)=ctl_FL19(z);

%%Vectors de senyal de control FR
AllctlFR(z)=ctl_FR(z);
AllctlFR(z+100)=ctl_FR2(z);
AllctlFR(z+200)=ctl_FR3(z);
AllctlFR(z+300)=ctl_FR4(z);

Allctl2FR(z)=ctl_FR6(z);
Allctl2FR(z+100)=ctl_FR7(z);
Allctl2FR(z+200)=ctl_FR8(z);
Allctl2FR(z+300)=ctl_FR9(z);

Allctl3FR(z)=ctl_FR11(z);
Allctl3FR(z+100)=ctl_FR12(z);
Allctl3FR(z+200)=ctl_FR13(z);
Allctl3FR(z+300)=ctl_FR14(z);

Allctl4FR(z)=ctl_FR16(z);
Allctl4FR(z+100)=ctl_FR17(z);
Allctl4FR(z+200)=ctl_FR18(z);
Allctl4FR(z+300)=ctl_FR19(z);

%%Vector de corrents FL
AllcurrentsFL(z)=curr_FL(z);
AllcurrentsFL(z+100)=curr_FL2(z);
AllcurrentsFL(z+200)=curr_FL3(z);
AllcurrentsFL(z+300)=curr_FL4(z);

Allcurrents2FL(z)=curr_FL6(z);
Allcurrents2FL(z+100)=curr_FL7(z);
Allcurrents2FL(z+200)=curr_FL8(z);
Allcurrents2FL(z+300)=curr_FL9(z);

Allcurrents3FL(z)=curr_FL11(z);
Allcurrents3FL(z+100)=curr_FL12(z);
Allcurrents3FL(z+200)=curr_FL13(z);
Allcurrents3FL(z+300)=curr_FL14(z);

Allcurrents4FL(z)=curr_FL16(z);
Allcurrents4FL(z+100)=curr_FL17(z);
Allcurrents4FL(z+200)=curr_FL18(z);
Allcurrents4FL(z+300)=curr_FL19(z);

%%Vector de corrents FR
AllcurrentsFR(z)=curr_FR(z);
AllcurrentsFR(z+100)=curr_FR2(z);
AllcurrentsFR(z+200)=curr_FR3(z);
AllcurrentsFR(z+300)=curr_FR4(z);

Allcurrents2FR(z)=curr_FR6(z);
Allcurrents2FR(z+100)=curr_FR7(z);
Allcurrents2FR(z+200)=curr_FR8(z);
Allcurrents2FR(z+300)=curr_FR9(z);

Allcurrents3FR(z)=curr_FR11(z);
Allcurrents3FR(z+100)=curr_FR12(z);
Allcurrents3FR(z+200)=curr_FR13(z);
Allcurrents3FR(z+300)=curr_FR14(z);

Allcurrents4FR(z)=curr_FR16(z);
Allcurrents4FR(z+100)=curr_FR17(z);
Allcurrents4FR(z+200)=curr_FR18(z);
Allcurrents4FR(z+300)=curr_FR19(z);
```

```

%%Vector de POTFL
AllPotFL(z)=AllctlFL(z)*Allbattery(z)*AllcurrentsFL(z);
AllPotFL(z+100)=AllctlFL(z+100)*Allbattery(z+100)*AllcurrentsFL(z+100);
AllPotFL(z+200)=AllctlFL(z+200)*Allbattery(z+200)*AllcurrentsFL(z+200);
AllPotFL(z+300)=AllctlFL(z+300)*Allbattery(z+300)*AllcurrentsFL(z+300);

AllPot2FL(z)=Allctl2FL(z)*Allbattery2(z)*Allcurrents2FL(z);
AllPot2FL(z+100)=Allctl2FL(z+100)*Allbattery2(z+100)*Allcurrents2FL(z+100);
AllPot2FL(z+200)=Allctl2FL(z+200)*Allbattery2(z+200)*Allcurrents2FL(z+200);
AllPot2FL(z+300)=Allctl2FL(z+300)*Allbattery2(z+300)*Allcurrents2FL(z+300);

AllPot3FL(z)=Allctl3FL(z)*Allbattery3(z)*Allcurrents3FL(z);
AllPot3FL(z+100)=Allctl3FL(z+100)*Allbattery3(z+100)*Allcurrents3FL(z+100);
AllPot3FL(z+200)=Allctl3FL(z+200)*Allbattery3(z+200)*Allcurrents3FL(z+200);
AllPot3FL(z+300)=Allctl3FL(z+300)*Allbattery3(z+300)*Allcurrents3FL(z+300);

AllPot4FL(z)=Allctl4FL(z)*Allbattery4(z)*Allcurrents4FL(z);
AllPot4FL(z+100)=Allctl4FL(z+100)*Allbattery4(z+100)*Allcurrents4FL(z+100);
AllPot4FL(z+200)=Allctl4FL(z+200)*Allbattery4(z+200)*Allcurrents4FL(z+200);
AllPot4FL(z+300)=Allctl4FL(z+300)*Allbattery4(z+300)*Allcurrents4FL(z+300);

%%Vector de POTFR
AllPotFR(z)=AllctlFR(z)*Allbattery(z)*AllcurrentsFR(z);
AllPotFR(z+100)=AllctlFR(z+100)*Allbattery(z+100)*AllcurrentsFR(z+100);
AllPotFR(z+200)=AllctlFR(z+200)*Allbattery(z+200)*AllcurrentsFR(z+200);
AllPotFR(z+300)=AllctlFR(z+300)*Allbattery(z+300)*AllcurrentsFR(z+300);

AllPot2FR(z)=Allctl2FR(z)*Allbattery2(z)*Allcurrents2FR(z);
AllPot2FR(z+100)=Allctl2FR(z+100)*Allbattery2(z+100)*Allcurrents2FR(z+100);
AllPot2FR(z+200)=Allctl2FR(z+200)*Allbattery2(z+200)*Allcurrents2FR(z+200);
AllPot2FR(z+300)=Allctl2FR(z+300)*Allbattery2(z+300)*Allcurrents2FR(z+300);

AllPot3FR(z)=Allctl3FR(z)*Allbattery3(z)*Allcurrents3FR(z);
AllPot3FR(z+100)=Allctl3FR(z+100)*Allbattery3(z+100)*Allcurrents3FR(z+100);
AllPot3FR(z+200)=Allctl3FR(z+200)*Allbattery3(z+200)*Allcurrents3FR(z+200);
AllPot3FR(z+300)=Allctl3FR(z+300)*Allbattery3(z+300)*Allcurrents3FR(z+300);

AllPot4FR(z)=Allctl4FR(z)*Allbattery4(z)*Allcurrents4FR(z);
AllPot4FR(z+100)=Allctl4FR(z+100)*Allbattery4(z+100)*Allcurrents4FR(z+100);
AllPot4FR(z+200)=Allctl4FR(z+200)*Allbattery4(z+200)*Allcurrents4FR(z+200);
AllPot4FR(z+300)=Allctl4FR(z+300)*Allbattery4(z+300)*Allcurrents4FR(z+300);

end

%% Plot Aprox. Lineal (mínims quadrats) de POT en funció de velocitat de totes les
dades
figure ('Name','Aproximació lineal PWMFL','NumberTitle','off','Color',[0.8 0.8
0.8])
vectoraproxlinealR=[AllspeedsFL;AllPotFL];
vectoraproxlinealM=[Allspeeds2FL;AllPot2FL];
vectoraproxlinealH=[Allspeeds3FL;AllPot3FL];
vectoraproxlinealS=[Allspeeds4FL;AllPot4FL];

save [mR,bR]=mincuadlin(vectoraproxlinealR);
save [mM,bM]=mincuadlin(vectoraproxlinealM);
save [mH,bH]=mincuadlin(vectoraproxlinealH);
save [mS,bS]=mincuadlin(vectoraproxlinealP);

global mR
mR
global bR
bR
global mM
mM
global bM
bM

```

```

global mH
mH
global bH
bH
global mS
mS
global bS
bS

nR=length(vectoraproxlinealR(1,:));
nM=length(vectoraproxlinealM(1,:));
nH=length(vectoraproxlinealH(1,:));
nS=length(vectoraproxlinealS(1,:));

AR=0;
AM=0;
AH=0;
AS=0;
BR=0;
BM=0;
BH=0;
BS=0;
CR=0;
CM=0;
CH=0;
CS=0;
DR=0;
DM=0;
DH=0;
DS=0;

for iii=1:nR;
    AR=AR+vectoraproxlinealR(1,iii);
    BR=BR+vectoraproxlinealR(2,iii);
    CR=CR+(vectoraproxlinealR(1,iii))^2;
    DR=DR+vectoraproxlinealR(1,iii)*vectoraproxlinealR(2,iii);
end

for iii=1:nM;
    AM=AM+vectoraproxlinealM(1,iii);
    BM=BM+vectoraproxlinealM(2,iii);
    CM=CM+(vectoraproxlinealM(1,iii))^2;
    DM=DM+vectoraproxlinealM(1,iii)*vectoraproxlinealM(2,iii);
end

for iii=1:nH;
    AH=AH+vectoraproxlinealH(1,iii);
    BH=BH+vectoraproxlinealH(2,iii);
    CH=CH+(vectoraproxlinealH(1,iii))^2;
    DH=DH+vectoraproxlinealH(1,iii)*vectoraproxlinealH(2,iii);
end

for iii=1:nS;
    AS=AS+vectoraproxlinealS(1,iii);
    BS=BS+vectoraproxlinealS(2,iii);
    CS=CS+(vectoraproxlinealS(1,iii))^2;
    DS=DS+vectoraproxlinealS(1,iii)*vectoraproxlinealS(2,iii);
end

mR=(nR*DR-AR*BR)/(nR*CR-AR^2);
bR=(CR*BR-DR*AR)/(nR*CR-AR^2);
mM=(nM*DM-AM*BM)/(nM*CM-AM^2);
bM=(CM*BM-DM*AM)/(nM*CM-AM^2);
mH=(nH*DH-AH*BH)/(nH*CH-AH^2);
bH=(CH*BH-DH*AH)/(nH*CH-AH^2);
mS=(nS*DS-AS*BS)/(nS*CS-AS^2);
bS=(CS*BS-DS*AS)/(nS*CS-AS^2);

```

```

for iii=1:nR;
    hold on;
    plot
    (vectoraproxlinealR(1,iii),vectoraproxlinealR(2,iii),'*','MarkerEdgeColor','b','LineWidth',1);
end

for iii=1:nM;
    hold on;
    plot
    (vectoraproxlinealM(1,iii),vectoraproxlinealM(2,iii),'*','MarkerEdgeColor','c','LineWidth',1);
end

for iii=1:nH;
    hold on;
    plot
    (vectoraproxlinealH(1,iii),vectoraproxlinealH(2,iii),'*','MarkerEdgeColor','g','LineWidth',1);
end

for iii=1:nS;
    hold on;
    plot
    (vectoraproxlinealS(1,iii),vectoraproxlinealS(2,iii),'*','MarkerEdgeColor','k','LineWidth',1);
end

xaproxlinealR=vectoraproxlinealR(1,1):1:vectoraproxlinealR(1,nR);
yaproxlinealR=mR*xaproxlinealR+bR;

xaproxlinealM=vectoraproxlinealM(1,1):1:vectoraproxlinealM(1,nM);
yaproxlinealM=mM*xaproxlinealM+bM;

xaproxlinealH=vectoraproxlinealH(1,1):1:vectoraproxlinealH(1,nH);
yaproxlinealH=mH*xaproxlinealH+bH;

xaproxlinealS=vectoraproxlinealS(1,1):1:vectoraproxlinealS(1,nS);
yaproxlinealS=mS*xaproxlinealS+bS;

plot(xaproxlinealR,yaproxlinealR,'b',xaproxlinealM,yaproxlinealM,'c',xaproxlinealH,
yaproxlinealH,'g',xaproxlinealS,yaproxlinealS,'k');
title('Aproximació lineal per mínims quadrats FL.');
```

% Plot Aprox. Lineal (mínims quadrats) de POT en funció de velocitat de totes les dades

```

figure ('Name','Aproximació lineal PWMFR','NumberTitle','off','Color',[0.8 0.8 0.8])
vectoraproxlinealRR=[AllspeedsFR;AllPotFR];
vectoraproxlinealMM=[Allspeeds2FR;AllPot2FR];
vectoraproxlinealHH=[Allspeeds3FR;AllPot3FR];
vectoraproxlinealSS=[Allspeeds4FR;AllPot4FR];

save [mRR,bRR]=mincuadlin(vectoraproxlinealRR);
save [mMM,bMM]=mincuadlin(vectoraproxlinealMM);
save [mHH,bHH]=mincuadlin(vectoraproxlinealHH);
save [mSS,bSS]=mincuadlin(vectoraproxlinealPP);

global mRR
mRR
global bRR
bRR
global mMM
mMM
global bMM
bMM
```



```

global mHH
mHH
global bHH
bHH
global mSS
mSS
global bSS
bSS

nRR=length(vectoraproxlinealRR(1,:));
nMM=length(vectoraproxlinealMM(1,:));
nHH=length(vectoraproxlinealHH(1,:));
nSS=length(vectoraproxlinealSS(1,:));

ARR=0;
AMM=0;
AHH=0;
ASS=0;
BRR=0;
BMM=0;
BHH=0;
BSS=0;
CRR=0;
CMM=0;
CHH=0;
CSS=0;
DRR=0;
DMM=0;
DHH=0;
DSS=0;

for iii=1:nRR;
    ARR=ARR+vectoraproxlinealRR(1,iii);
    BRR=BRR+vectoraproxlinealRR(2,iii);
    CRR=CRR+(vectoraproxlinealRR(1,iii))^2;
    DRR=DRR+vectoraproxlinealRR(1,iii)*vectoraproxlinealRR(2,iii);
end

for iii=1:nMM;
    AMM=AMM+vectoraproxlinealMM(1,iii);
    BMM=BMM+vectoraproxlinealMM(2,iii);
    CMM=CMM+(vectoraproxlinealMM(1,iii))^2;
    DMM=DMM+vectoraproxlinealMM(1,iii)*vectoraproxlinealMM(2,iii);
end

for iii=1:nHH;
    AHH=AHH+vectoraproxlinealHH(1,iii);
    BHH=BHH+vectoraproxlinealHH(2,iii);
    CHH=CHH+(vectoraproxlinealHH(1,iii))^2;
    DHH=DHH+vectoraproxlinealHH(1,iii)*vectoraproxlinealHH(2,iii);
end

for iii=1:nSS;
    ASS=ASS+vectoraproxlinealSS(1,iii);
    BSS=BSS+vectoraproxlinealSS(2,iii);
    CSS=CSS+(vectoraproxlinealSS(1,iii))^2;
    DSS=DSS+vectoraproxlinealSS(1,iii)*vectoraproxlinealSS(2,iii);
end

mRR=(nRR*DRR-ARR*BRR)/(nRR*CRR-ARR^2);
bRR=(CRR*BRR-DRR*ARR)/(nRR*CRR-ARR^2);
mMM=(nMM*DMM-AMM*BMM)/(nMM*CMM-AMM^2);
bMM=(CMM*BMM-DMM*AMM)/(nMM*CMM-AMM^2);
mHH=(nHH*DHH-AHH*BHH)/(nHH*CHH-AHH^2);
bHH=(CHH*BHH-DHH*AHH)/(nHH*CHH-AHH^2);
mSS=(nSS*DSS-ASS*BSS)/(nSS*CSS-ASS^2);
bSS=(CSS*BSS-DSS*ASS)/(nSS*CSS-ASS^2);

```

```
for iii=1:nRR;
    hold on;
    plot
    (vectoraproxlinealRR(1,iii),vectoraproxlinealRR(2,iii),'*','MarkerEdgeColor','b','L
    ineWidth',1);
end

for iii=1:nMM;
    hold on;
    plot
    (vectoraproxlinealMM(1,iii),vectoraproxlinealMM(2,iii),'*','MarkerEdgeColor','c','L
    ineWidth',1);
end

for iii=1:nHH;
    hold on;
    plot
    (vectoraproxlinealHH(1,iii),vectoraproxlinealHH(2,iii),'*','MarkerEdgeColor','g','L
    ineWidth',1);
end

for iii=1:nSS;
    hold on;
    plot
    (vectoraproxlinealSS(1,iii),vectoraproxlinealSS(2,iii),'*','MarkerEdgeColor','k','L
    ineWidth',1);
end

xaproxlinealRR=vectoraproxlinealRR(1,1):1:vectoraproxlinealRR(1,nRR);
yaproxlinealRR=mRR*xaproxlinealRR+bRR;

xaproxlinealMM=vectoraproxlinealMM(1,1):1:vectoraproxlinealMM(1,nMM);
yaproxlinealMM=mMM*xaproxlinealMM+bMM;

xaproxlinealHH=vectoraproxlinealHH(1,1):1:vectoraproxlinealH(1,nHH);
yaproxlinealHH=mHH*xaproxlinealHH+bHH;

xaproxlinealSS=vectoraproxlinealSS(1,1):1:vectoraproxlinealSS(1,nSS);
yaproxlinealSS=mSS*xaproxlinealSS+bSS;

plot(xaproxlinealRR,yaproxlinealRR,'b',xaproxlinealMM,yaproxlinealMM,'c',xaproxline
alHH,yaproxlinealHH,'g',xaproxlinealSS,yaproxlinealSS,'k');
title('Aproximació lineal per mínims quadrats FR.');
```

B MANUAL COMUNICACIÓ LABVIEW - ROBOT BIGBOT

Aquest manual de comunicació explica les diferents maneres de monitoritzar el robot Bigbot amb la interfície gràfica de Labview i altres funcions com ara la lectura de variables, la visualització de gràfiques, seqüència de fotos, etc.

B.1 Com començar

Poseu en marxa el motor del robot polsant l'interruptor col·locat a la part lateral esquerra. Noteu seguidament que s'encendrà el LED de color vermell que indica que està encès.

Obriu el programa LabVIEW 2011 i obriu l'aplicació Bot_5.2.vi (>>Escriptori>>Control Robot>> Comunicació wifibot 5.3>>Bot_5.2.vi).

Interfície d'usuari

Un cop oberta l'aplicació, s'obrirà el Panell de Control on hi consten totes les opcions que permet el programa.

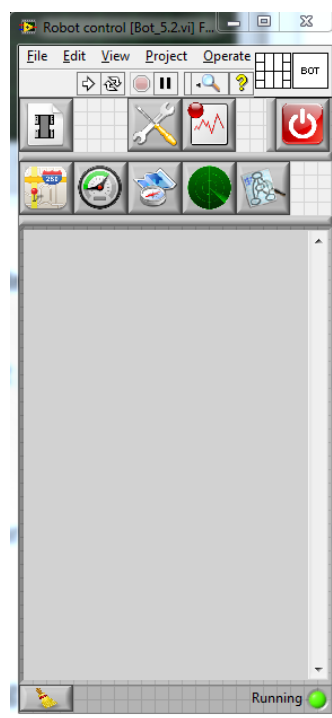




Figura 43. Interfície d'usuari.

Per iniciar el programa, polseu el botó que conté una fletxa de dretes blanca  que està a la interfície d'usuari del control del robot. Veieu que el LED de la part inferior del Panell continua estant de color verd.

Un cop l'aplicació està activada, ja podeu anar a escollir el robot que voleu fer servir prement la opció "Open new robot instance" . Seguidament s'obriran dues pantalles. La de l'esquerra, "Connections options", és on haureu d'escollir quin robot voleu fer servir mitjançant el selector desplegable. Noteu que depenent del robot que s'esculli, la taula d'informació varia juntament amb el color que serveix per diferenciar el robot que s'ha triat.

Cada robot disposa d'unes IP que l'identifiquen, així mateix com les parts importants d'aquest. Noteu que les IP acaven amb el mateix número. Les IP corresponen al Router, el dsPIC, el Giroscopi i el Laser. A la part dreta de la interfície "Connection options", permet habilitar o deshabilitar el Laser i el Giroscopi (veure pàg. 10 de Docu_robots.doc). Tantmateix, podeu escollir quin mode de connexió amb el circuit de control s'hi vol, directe o indirecte.

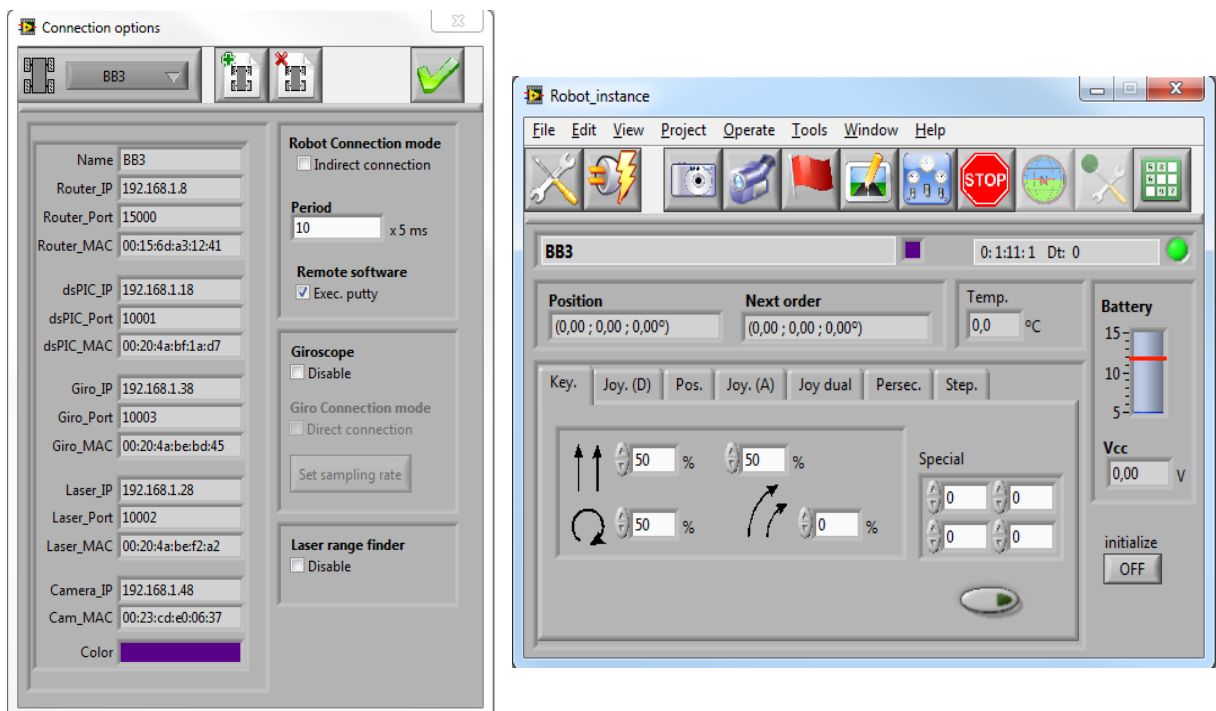




Figura 44. Configuració robot.


Un cop heu escollit el robot i les opcions amb les que voleu treballar feu "click" al tic per acceptar i passeu a treballar a la pantalla de la dreta a la qual li direm a partir d'ara Robot_instance.

Per poder iniciar el moviment amb el robot, heu de connectar el robot mitjançant el botó "Connect or disconnect from de robot"  de la pantalla Robot_instance.

Un cop s'ha fet la connexió del robot, la mateixa pantalla, permet tornar a les opcions de connexió mitjançant el botó . Noteu que a la part dreta hi ha un cilindre el qual indica el nivell actual de la bateria.

B.2 Com moure el robot

Al panell Robot_instance, hi ha disponible la opció amb la qual voleu moure el robot. Pot ser amb el teclat, amb un joystick digital, joystick analògic, joystick dual i pas a pas. El teclat disposa de quatre paràmetres a tenir en compte, treballant per defecte en llaç obert, i per tant, aquests paràmetres treballen en %.

La opció "Pos", us dirigeix a un únic selector de valor anomenat speed, és a dir, moureu el robot definint una trajectòria a la velocitat que haureu posat dintre la pestanya "Pos". Per definir una trajectòria, dintre el mateix panell de control, haureu d'anar a "Define position orders for the robot" . Noteu que s'obrirà la següent finestra:

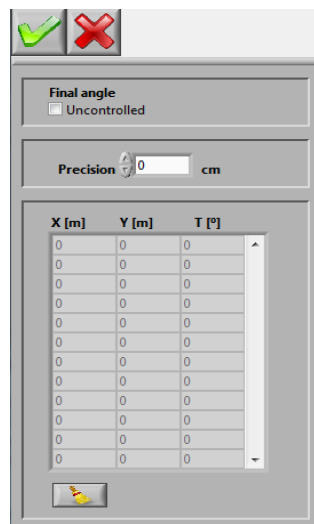


Figura 45. Definició posicions.

Veieu que la finestra disposa d'una taula on podeu posar les posicions on el robot ha d'anar. Observeu que podeu activar la opció de controlar l'angle i ajustar la precisió amb la qual ha d'anar a la posició desitjada. Un cop establerta la posició, només us queda definir la velocitat a la qual voleu que es desplaci el robot si és que no l'heu definit abans a la pestanya "Pos" del comandament del robot.

La pestanya "Step" del comandament del robot, permet que aquest es desplaci pas a pas. Es tracta d'anar marcant les velocitats de la roda a les que voleu que hi vagin. Noteu que per aquest fet, a cada pas se li ha de definir en quin temps voleu que el realitzi. Veieu també, que cada columna està referida a cadascuna de les rodes que fa que el robot es mogui. Així

doncs, enteneu que FL i FR són les rodes frontals esquerra i dreta respectivament i que RL i RR són les rodes posteriors esquerra i dreta respectivament.

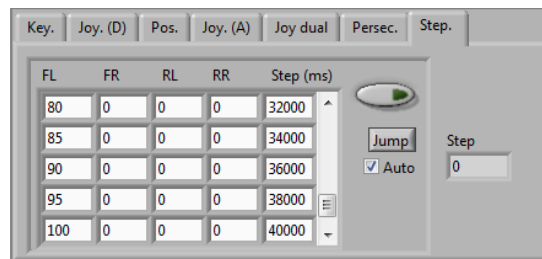


Figura 46. Comandament pas a pas.

B.3 Altres opcions de la Interfície.

La mateixa pantalla disposa de quatre paràmetres més, al requadre anomenat especial, referit a les quatre rodes del robot. Aquests paràmetres es fan servir majoritàriament com a prova, és a dir, en situacions on el robot hagi de detectar un canvi de superfície. Els quatre selectores numèrics encaixats referencien les quatre rodes, així doncs, si canviu el paràmetre superior esquerra, automàticament el valor s'associa amb la roda davantera esquerra del robot.

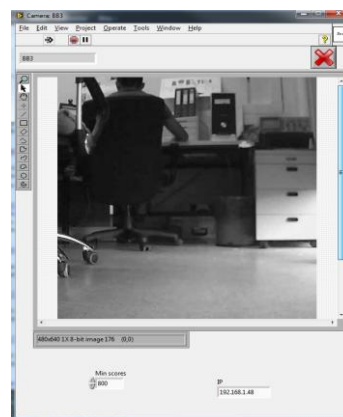



Figura 47. Finestra seqüència fotogràfica.

Amb el robot, ja en marxa, podeu iniciar una seqüència d'imatges prement el botó  del panell de control Robot_instance, el qual obrirà una finestra alternativa que mostrarà el que veu el robot des de la càmera que té incorporada.



El botó , permet una visió directa, des de l'ordinador, a través de la càmera, del que està veient el robot, no obstant, la imatge té un retard d'una mica menys d'un segon.



Figura 48. Finestra visió directa del robot.

Al prémer el botó amb la icona de la càmera, es desplega una finestra com la de la figura 4.

La icona "Correct the actual position from the robot" , permet precisar la posició actual a la qual volem que el robot comenci.

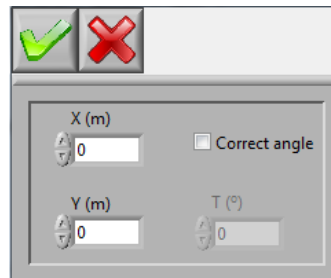



Figura 49. Posició Correcta del Robot.

Veiem que la posició és tant en x com en y i que també podeu ajustar el desfasament amb el que voleu el propi robot.

Disposeu d'una opció Stop , en la qual al pulsar, es detindrà qualsevol ordre que li pugueu enviar al robot sense detenir l'aplicació.

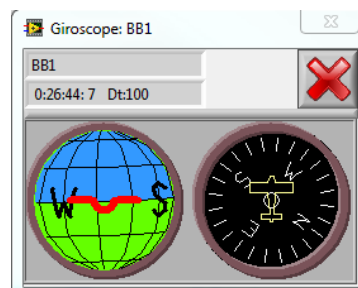




Figura 50. Brúixola

Si heu activat el giroscopi a les opcions de configuració del robot, podeu saber cap a on està el nord i el sud per mitjà del botó .

A la interfície gràfica, podeu escollir també, que us mostri una gràfica on es dibuixa la trajectòria del robot. Aneu a “View a position graph of all robot instances”  i observeu que se us mostrarà una finestra com la de la figura 8.

Veieu que ja hi ha una trajectòria dibuixada com a exemple.

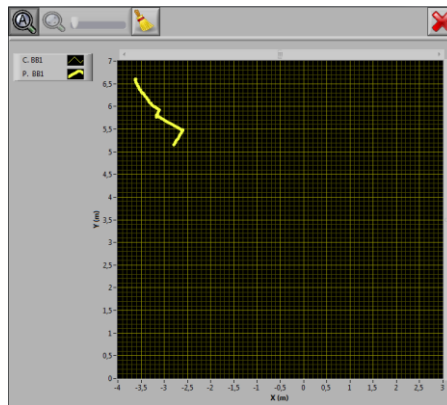




Figura 51. Gràfica de Posició.

Noteu que a la part superior esquerra hi ha una lupa amb una A. Només clicant, podreu escalar la gràfica manualment.

Noteu que a la interfície d'usuari també disposeu de dues opcions per veure una gràfica referida al giroscopi i l'altra al làser. Per accedir-hi, clicant al botó de la gràfica que voleu que dibuixi des de la interfície (Open a plot window for the gyroscope data ) o (Open a plot window for laser range data ) , se us podran obrir dues finestres com les següents:

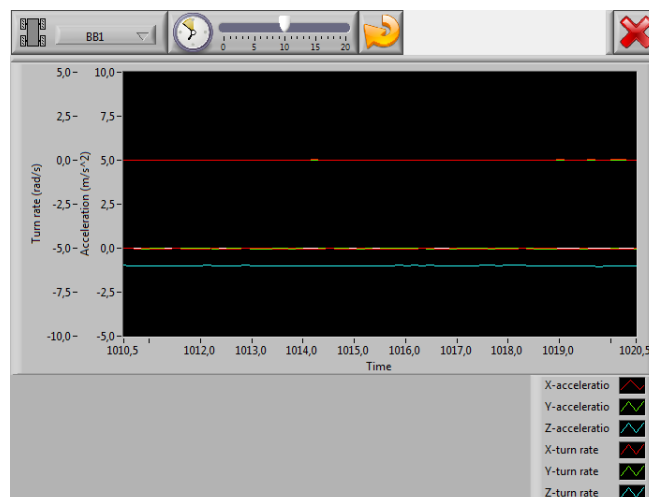


Figura 52. Gràfica Giroscopi

Observeu que obteniu les dades exposades a la llegenda que està a la part inferior dreta.

Pel que fa a la gràfica del làser, veieu que us mostra en punts, els objectes que detecta. Els objectes fora del marge de visió, són perquè el robot inicialment estava en una altra posició.

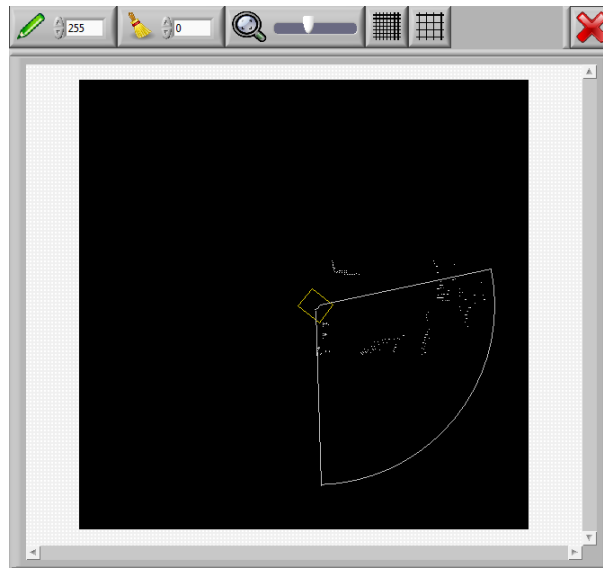



Figura 53. Gràfica Làser

Fixeu-vos però, que abans que s'obri la finestra, us demanarà de quines dimensions voleu el mapa del làser.

També podeu canviar les opcions del làser a la configuració del robot fent un clic a "Display de virtual horitzont for this robot" .

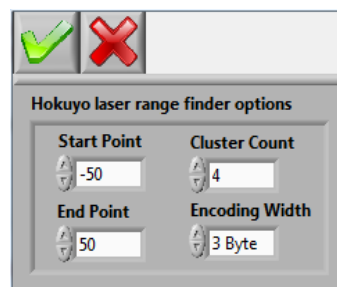



Figura 54. Opcions Làser.

B.4 Controlador PID

El control del robot és en llaç obert principalment, cosa pel qual, no hi ha realimentació i per tant, el robot no retorna dades, així doncs, és molt difícil que la consigna de velocitat coincideixi a la que va el robot sigui exacte al no poder ser corregida. A diferència del control en llaç obert, el controlador PID disposa de realimentació la qual permet calcular l'error entre

un valor mig i el valor que es vol obtenir. Per activar el controlador, heu de fer un clic a “control parameters”  . Noteu que s’obrirà una finestra tal com aquesta:

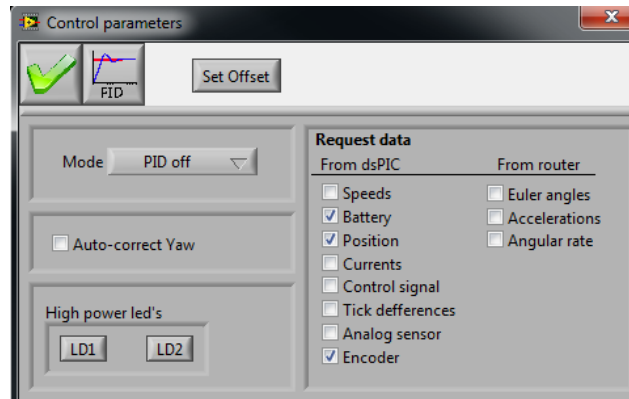
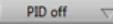



Figura 55. Controlador PID.

Podeu escollir el mode de funcionament del PID mitjançant la selecció desplegable anomenat Mode  . Es possible que el PID controli dues rodes mitjançant velocitat, quatre rodes mitjançant velocitat i quatre rodes mitjançant corrent.

El controlador també permet canviar les variables K_p , K_i i K_d prement el botó situat al costat del tic d’acceptació  . Noteu que s’obrirà una finestra tal que aquesta:

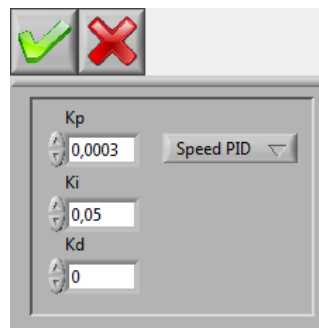



Figura 56. Paràmetres PID.

Observeu que K_p , K_i i K_d ja tenen una valors predeterminats que depenen per on es mogui el robot aniran bé o no. A l’igual que abans, podem definir si aquests paràmetres són pel controlador PID de velocitat o d’intensitat.

A la finestra “Control Parameters” podeu escollir les dades que voleu que el robot us enviï des del dsPIC i des del router.

Request data	
From dsPIC	From router
<input type="checkbox"/> Speeds	<input type="checkbox"/> Euler angles
<input checked="" type="checkbox"/> Battery	<input type="checkbox"/> Accelerations
<input checked="" type="checkbox"/> Position	<input type="checkbox"/> Angular rate
<input type="checkbox"/> Currents	
<input type="checkbox"/> Control signal	
<input type="checkbox"/> Tick differences	
<input type="checkbox"/> Analog sensor	
<input checked="" type="checkbox"/> Encoder	

Figura 57. Request data.

Per visualitzar aquestes dades, heu d'anar a la interfície d'usuari principal i accedir a la finestra de les gràfiques de les velocitats i corrent del robot . Noteu que s'obre una finestra com la següent:

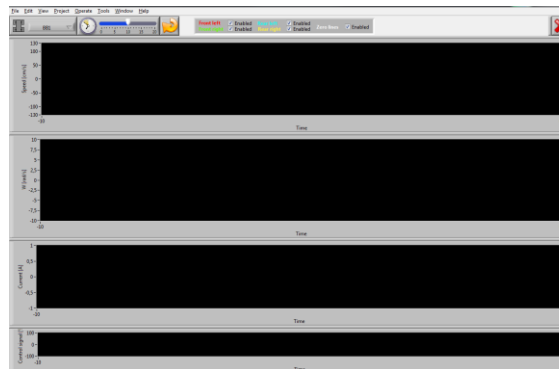


Figura 58. Gràfiques del Controlador.