



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA TÈCNICA
SUPERIOR ENGINYERIA
INDUSTRIAL VALÈNCIA

TREBALL FINAL DE GRAU EN ENGINYERIA EN TECNOLOGIES INDUSTRIALS

DESENVOLUPAMENT I IMPLEMENTACIÓ DE CONTROLADORS DINÀMICS PER A UN ROBOT CARTESIÀ SOBRE UN CONTROLADOR ENCASTRAT I FPGA

AUTOR: HÈCTOR MARTÍN LÀZARO

TUTOR: MARINA VALLÈS MIQUEL

COTUTOR: ÀNGEL VALERA FERNÁNDEZ

Curs Acadèmic: **2019-20**

Desenvolupament i implementació de controladors dinàmics per a un robot cartesià sobre un controlador encastrat i FPGA

Aquest treball ha consistit en dissenyar el control dels motors d'una impressora 3D. Per a tal fi, s'ha utilitzat un controlador encastrat, una myRIO. El control d'aquests motors en LabView ha estat programat, implantant-se posteriorment en la myRIO. També s'ha simulat el funcionament del robot cartesià mitjançant Simulink i imitant aquest simulador en LabView, s'ha estudiat com realitzar la comunicació entre LabView i Simulink, havent-se implantat després. Per últim, s'han realitzat proves per comprovar el correcte funcionament dels motors.

Paraules Clau: Impresora, myRIO, control, comunicació,

Este trabajo ha consistido en diseñar el control de los motores de una impresora 3D. Para ello, se ha utilizado un controlador encastrado, una myRIO. El control de estos motores ha sido programado en LabView, implantándose posteriormente en la myRIO. También se ha simulado el funcionamiento del robot cartesiano mediante Simulink y imitando este simulador en LabView, se ha estudiado como realizar la comunicación entre LabView y Simulink, implantándose después. Por último, se han realizado pruebas para comprobar el correcto funcionamiento de los motores.

Palabras Clave: Impresora, myRIO, control, comunicació.

This assignment has consisted in designing the control of the motors of a 3D printer. For this reason, it has been used an embedded controller, a myRIO. The control of this motors has been programmed on LabView and subsequently installed to the myRIO. Moreover, the running of the cartesian robot has been simulated with Simulink and being imitated on LabView, the communication between LabView and Simulink has been studied and afterwards implemented. Finally, the motors have been tested in order to check the proper functioning of the motor.

Keywords: Printer, myRIO, control, communication.

Desenvolupament i implementació de controladors dinàmics per a un robot cartesià sobre un controlador encastrat i FPGA

Índex

- Document I: Memòria
- Document II: Pressupost

Índex de la memòria

1.- Introducció.....	pàg. 6
1.1.- Introducció i motivació.....	pàg. 6
1.2.- Controladors actuals en l'indústria.....	pàg. 6
1.2.- Objectius.....	pàg. 8
2.- Components.....	pàg. 9
2.1.- myRio-1900.....	pàg. 9
2.1.1.- FPGA i els seus avantatges.....	pàg. 12
2.1.2.- Comparació amb els PLC.....	pàg. 12
2.2.- Me Stepper Driver.....	pàg. 13
2.3.- Motor pas a pas.....	pàg. 14
2.4.- Simulador pont grua.....	pàg. 15
2.4.1.- Descripció general.....	pàg. 15
2.4.2.- Adquisició del punt objectiu.....	pàg. 16
2.4.3.- Controladors.....	pàg. 17
2.4.3.1.- Avantatges del control en cascada.....	pàg. 18
3.- Control del motor.....	pàg. 21
3.1.- Muntatge del motor.....	pàg. 21
3.2.- Interfície de control.....	pàg. 22
3.3.- Control del motor.....	pàg. 22
4.- Comunicació TCP/IP.....	pàg. 23
4.1.- Enviament de dades.....	pàg. 23
4.2.- Rebuda de dades.....	pàg. 24
5.- Control del simulador.....	pàg. 26

5.1.- Botonera implementada a LabView.....	pàg. 26
5.2.- Controladors implementats a LabView.....	pàg. 27
5.2.1.- Interfície de control.....	pàg. 27
5.2.2.- Control eixos X i Z.....	pàg. 27
5.2.3.- Control eix Y.....	pàg. 30
5.2.4.- Modificacions a Simulink.....	pàg. 30
5.2.5.- Verificació dels resultats.....	pàg. 31
5.2.6.- Ajust dels controladors de l'eix X.....	pàg. 32
6.- Conclusions.....	pàg. 35
7.- Bibliografia.....	pàg. 35

Índex de figures

Figura 1: Stepper Driver A4988.....	pàg. 6
Figura 2: Stepper Driver DRV8825.....	pàg. 7
Figura 3: Stepper Driver LV8729.....	pàg. 7
Figura 4: Stepper Driver TMC2130.....	pàg. 8
Figura 5: Diagrama de bloc myRIO-1900.....	pàg. 9
Figura 6: Entrades i eixides d'un MXP.....	pàg. 10
Figura 7: Entrades i eixides d'un MSP.....	pàg. 11
Figura 8: Vista del driver.....	pàg. 13
Figura 9: Vista del motor pas a pas.	pàg. 14
Figura 10: Connexions del motor.....	pàg. 14
Figura 11: Vista frontal pont grua.	pàg. 15
Figura 12: Vista aèria pont grua.....	pàg. 15
Figura 13: Pantalla principal vrcrane_panel.....	pàg. 15
Figura 14: Bloc del punt objectiu.....	pàg. 16

Figura 15: Punt objectiu modificat.....	pàg. 16
Figura 16: Bloc de controladors.....	pàg. 17
Figura 17: Controlador eix X i Z.....	pàg. 17
Figura 18: Controlador eix Y.....	pàg. 17
Figura 19: Resposta control en cascada i control simple.....	pàg. 18
Figura 20: Controladors sense cascada.....	pàg. 18
Figura 21: Diferents controls en cascada.....	pàg. 19
Figura 22: Control en cascada (velocitat).....	pàg. 20
Figura 23: Control en cascada (angle).....	pàg. 20
Figura 24: Muntatge del motor.....	pàg. 21
Figura 25: Interfície del control del motor.....	pàg. 22
Figura 26: Control del motor.....	pàg. 23
Figura 27: Comunicació TCP/IP LabView.....	pàg. 23
Figura 28: Configuració TCP/IP Client Recieve.....	pàg. 24
Figura 29: Enviament dades des de Simulink.....	pàg. 25
Figura 30: Comunicació TCP/IP LabView.....	pàg. 25
Figura 31: Botonera a LabView.....	pàg. 26
Figura 32: Control per botonera Simulink.....	pàg. 27
Figura 33: Interfície control per referència.....	pàg. 27
Figura 34: Control LabView eix X.....	pàg. 28
Figura 35: Obtenció de l'acció de control.....	pàg. 28
Figura 36: Enviament dades a Simulink.....	pàg. 29
Figura 37: Recuperació de dades.....	pàg. 29
Figura 38: Control eix Y LabView.....	pàg. 30
Figura 39: Nou bloc d'adquisició del punt objectiu.....	pàg. 30
Figura 40: Nou bloc de controladors, eix X.....	pàg. 31
Figura 41: Gràfica de la posició X.....	pàg. 31
Figura 42: Gràfica de l'eix Y.....	pàg. 32
Figura 43: Experiment 1.....	pàg. 33
Figura 44: Experiment 2.....	pàg. 33

Figura 45: Experiment 3.....	pàg. 33
Figura 46: Experiment 4.....	pàg. 33
Figura 47: Experiment 5.....	pàg. 33
Figura 48: Experiment 6.....	pàg. 33
Figura 49: Experiment 7.....	pàg. 34
Figura 50: Experiment 8.....	pàg. 34

Índex de taules

Taula 1: Descripció de les senyals d'un MXP.....	pàg. 10
Taula 2: Descripció de les senyals del MSP.....	pàg.11
Taula 3: Modes del Me Stepper Driver.....	pàg. 13
Taula 4: Valors dels controlados.....	pàg. 32

DOCUMENT I: MEMORIA

1.- Introducció

1.1- Introducció i motivació

En els últims anys el món de la fabricació ha sofert una revolució. Cada vegada més empreses de molts sectors diferents s'estan unint al corrent de la impressió 3D. Des de indústries que l'utilitzen per a fabricar els seus components a usos domèstics. Fins i tot en el àmbit de la medicina i de l'art es fa ús d'aquesta tecnologia.

Es un realitat que s'ha convertit en una part molt important en la fabricació de productes, principalment elements d'una mida reduïda amb un elevat nivell de detall. Aquesta necessitat d'una gran precisió fa que cada vegada es busquen mètodes per aconseguir una millora.

En el nostre cas anem a implementar el control de tres motors pas a pas, degut a la gran precisió que poden arribar a tindre, un gir de 0.11° per cada pas, que seran els motors de la nostra impressora 3D. També es va a utilitzar el dispositiu myRIO, que serveix com a introducció als controladors utilitzats en la indústria.

Per motius de la crisi sanitària viscuda com a conseqüència de la Covid19 , gran part d'aquest treball s'ha realitzat des de casa, impossibilitant el treball directe sobre els motors o la impressora. Es per aquest motiu que s'utilitza el programa Matlab, concretament Simulink, com a simulador d'aquesta. Disposa d'un exemple d'un pont grua que serveix com a imitació del funcionament dels motors de la impressora.

1.2- Controladors actuals en l'indústria

Com ja s'ha comentat abans, per al funcionament dels motors pas a pas es necessari l'ús d'uns controladors que li indiquen quan ha de realitzar cada pas. En l'actualitat existeixen gran varietat de controladors per a les impressores 3D, però anem a centrar-nos en els quatre tipus mes utilitzats en la actualitat:

El primer de tots es el Stepper Driver A4988 (figura 1). Es un controlador molt econòmic i fàcil d'utilitzar i per tant uns dels mes utilitzats. Permeten realitzar fins a 1/16 passos. Con a inconvenient es un soroll elevat en comparació amb altres i un rendiment baix. També han d'estar correctament ventilats per evitar problemes de funcionament.

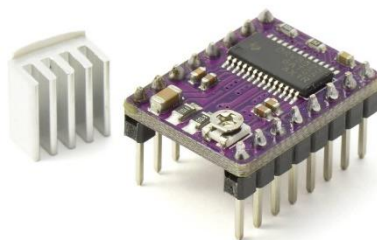


Figura 1: Stepper Driver A4988

A continuació tenim el DRV8825 (figura 2). Es poden considerar una evolució respecte als anteriors. Tenen una millor relació qualitat/preu i produeixen molt menys soroll. També són més eficients tèrmicament. Aquests permeten arribar als 1/32 passos sent molt més precisos que els A4988. En contra, segueixen necessitant una correcta ventilació i poden generar soroll en circumstàncies concretes en algunes impressions.

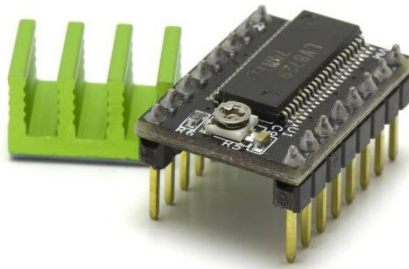


Figura 2: Stepper Driver DRV8825

En tercer lloc tenim els LV8729 (figura 3). Aquests són més avançats i permeten treballar en corrents més elevades i arribar a una precisió major, de 1/128 passos. Són molt eficients i poden moure motors de potència considerable. El inconvenient és que per a poder treballar amb aquesta precisió cal treballar en electrònica de 32 bits, amb 8 bits només poden arribar a 1/32 passos.

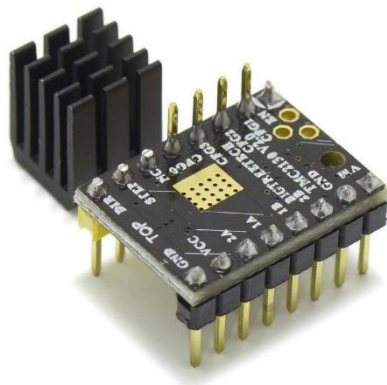


Figura 3: Stepper Driver LV8729

Per últim tenim la gama TMCxxxx. Existeixen molts tipus de controladors TMC, sent els més utilitzats els TMC2130 (figura 4). Son molt silenciosos i permeten detectar la pèrdua de passos. A més tenen diferents protocols de funcionament i poden ser connectats directament a la impressora. També poden fer la funció de finals de carrera. Com a inconvenient el preu, molt més elevat que en els anteriors.

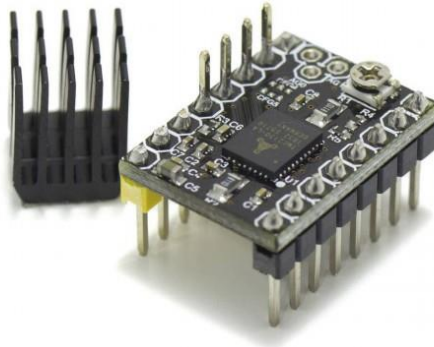


Figura 4: Stepper Driver TMC2130

La informació sobre aquests drivers ha estat extreta de diverses pàgines web centrades en impressores 3D, concretament de *hta3d* i de *deimpresoras3d*.

1.3- Objectius

EL objectiu principal d'aquest treball es analitzar i desenvolupar el control dels motors d'una impressora 3D de manera que siga d'utilitat per a futurs treballs i projectes. Els principals objectius del treball quedarien resumits com:

- Introducció a l'ús de controladors encastrats mitjançant la myRIO.
- Estudi del funcionament dels motors pas a pas i del control d'aquests.
- Programació del control d'un motor pas a pas i posterior implementació a un motor real.
- Verificació del correcte funcionament del motor.

Degut al ja mencionat abans, alguns d'aquests objectius es tornaren impossibles de realitzar, per tant nous objectius han sigut definits per a un millor ajust al treball que era possible realitzar:

- Estudi i implementació de la comunicació TCP/IP entre LabView i Matlab/Simulink.
- Anàlisi d'un robot cartesià gracies al simulador del pont grua de Simulink.
- Desenvolupament de diferents controls per al robot cartesià i tria de la millor opció.

2.- Components

2.1- myRIO-1900

La myRIO-1900 es un dispositiu reconfigurable de E/S (RIO) que s'utilitza en el disseny de sistemes de control, robòtica i mecatrònica. Pel que fa al hardware, la myRIO proporciona entrades i eixides analògiques (AI, AO) i digitals (DIO), entrada i eixida d'àudio, un acceleròmetre i LEDs. Per a la connexió amb un ordinador, disposa de ports USB i connexió inalàmbrica. Consta també de dos controladors, un processador dual-core ARM i una FPGA., de l'anglès *field-programmable gate array*, que significa matriu de portes reprogramable.

Permet executar programes de LabView directament des de l'ordinador o des de la FPGA, ja que aquesta es programable.

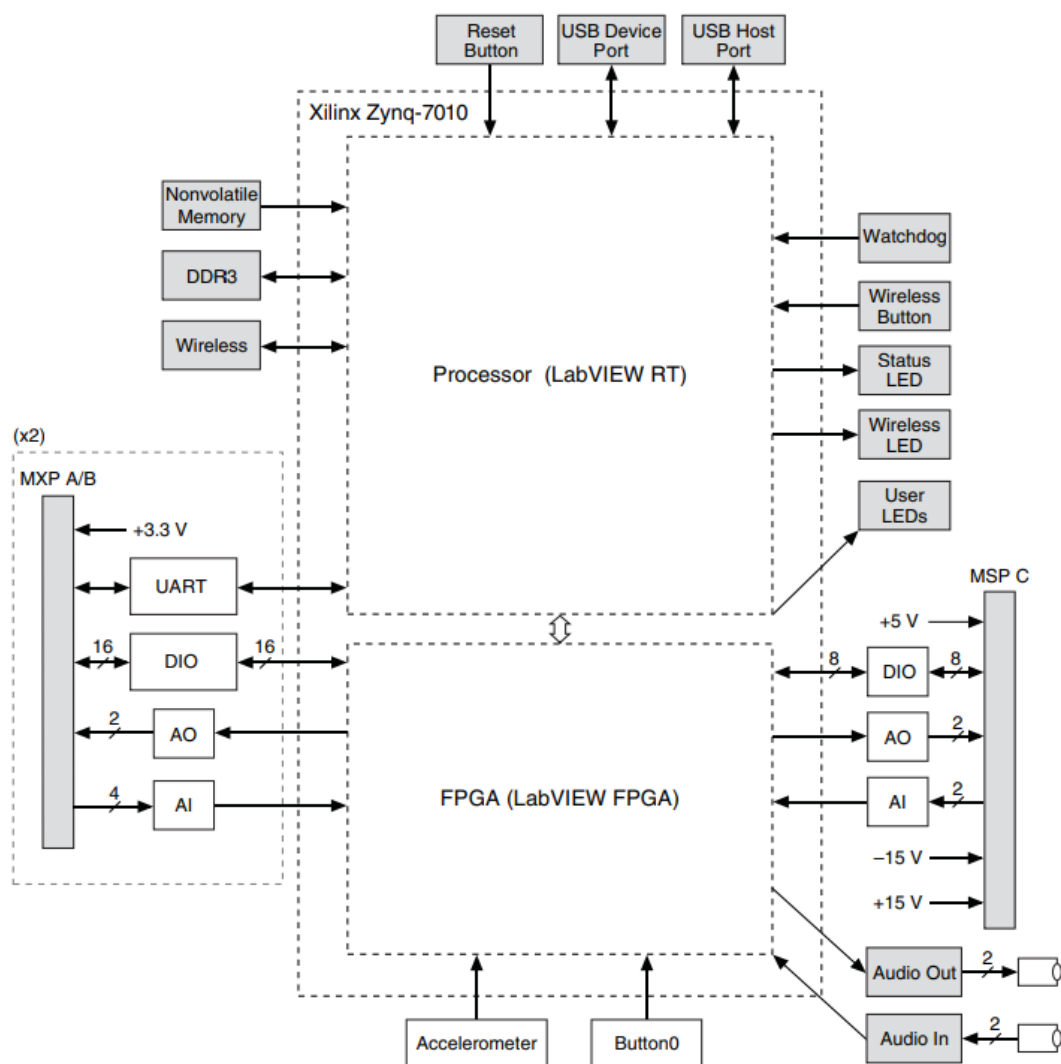


Figura 5: Diagrama de bloc myRIO-1900

Les entrades i eixides estan repartides en tres grups: dos Expansion Port (MXP) amb les mateixes senyals i un Mini System Port (MSP), anomenats connectors A,B i C respectivament.

Descripció de les senyals d'un MXP

NOM	TIPUS	DIRECCIÓ	DESCRIPCIÓ
+5V	Digital	Eixida	Eixida de +5 volts
AI <0...3>	Analògica	Entrada	Entrades analògiques de 0-5 V
AO <0...1>	Analògica	Eixida	Eixides analògiques de 0-5 V
+3.3V	Digital	Eixida	Eixida de +3.3 volts
DIO <0...15>	Digital	Entrada/Eixida	Entrada de 3.3 V, eixida de 3.3/5 V
UART.RX	Digital	Entrada	Entrada UART, elèctricament idèntica a les DIO
UART.TX	Digital	Eixida	Eixida UART, elèctricament idèntica a les DIO

Taula 1: Descripció de les senyals d'un MXP

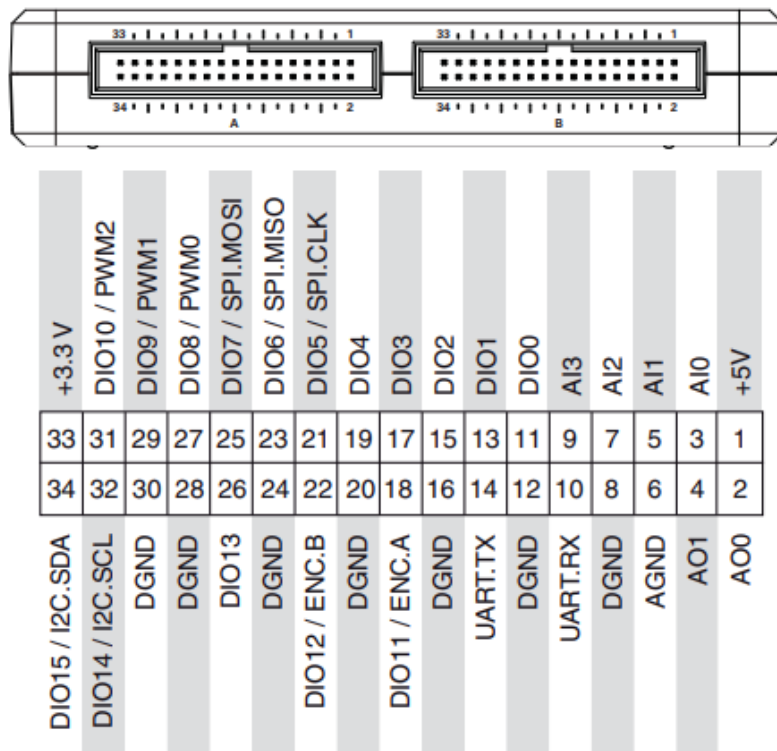


Figura 6: Entrades i eixides d'un MXP

Descripció de les senyals del MSP

NOM	TIPUS	DIRECCIÓ	DESCRIPCIÓ
+15 V / -15 V	Analògica	Eixida	Eixida de +15 V / -15 V
AI0+ / AI0-; AI1+ / AI1-	Analògica	Entrada	Entrades analògiques de ± 10 V
AO <0...1>	Analògica	Eixida	Eixides analògiques de ± 10 V
+5 V	Digital	Eixida	Eixida de +5 V
DIO <0...7>	Digital	Entrada/Eixida	Entrada de 3.3 V, eixida de 3.3/5 V
Àudio IN	-	Entrada	Entrada d'un connector estèreo
Àudio OUT	-	Eixida	Eixida d'un connector estèreo

Taula 2: Descripció de les senyals del MSP

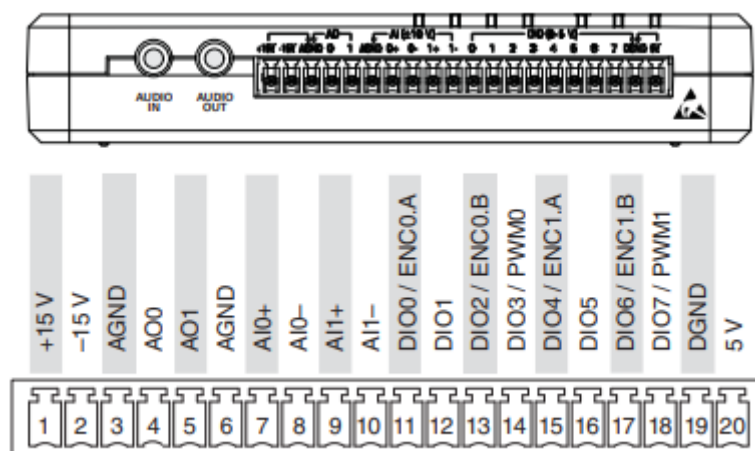


Figura 7: Entrades i eixides d'un MSP

LabView disposa d'un complement per a poder treballar amb la myRIO. Amb aquest complement es pot programar directament en la FPGA i porta per defecte uns models senzills per a facilitar el treball en la myRIO. Aquest models incorporen ja una inicialització del dispositiu i un reinici al acabar el programa.

Tota aquesta informació s'ha obtingut del manual de NI, User Guide and Specifications NI myRIO-1900, que la pròpia empresa proporciona.

2.1.1- FPGA i els seus avantatges

FPGA es el acrònim en anglès de *field-programmable gate array*, que traduït seria matriu de portes reprogramable. Consisteix en un circuit integrat en blanc, es a dir, que pot ser configurat per el usuari. Conte blocs lògics, portes lògiques i biestables sense connectar, permetent realitzar les connexions com es vullga.

El primera avantatge es la flexibilitat que açò proporciona. Al ser reconfigurable es poden modificar els diferents programes sense haver de realitzar canvis físics, reduint el cost.

El segon avantatge i els mes important de tots es l'acceleració que pot aportar. En processador normal es tenen un conjunt fixe d'instruccions i aquest no pot utilitzar mes que les ja definides. Una FPGA en canvi, al no es estar prèviament configurada, es pot configurar de la manera mes òptima per al nostre us. També aquest circuit es capaç de treballar en múltiples senyals a la volta, per tant, pot realitzar diferents funcions simultàniament, estant limitada per l'espai configurable disponible.

Per a aprofitar aquesta millora en la velocitat, s'utilitzen FPGA en companyia de processadors estàndards, ja que les FPGA tenen la memòria volàtil i necessiten elements externs.

En el nostre cas, la myRIO també consta d'una FPGA i d'un microprocessador. Amb LabView es possible realitzar la programació sobre la FPGA o sobre el microprocessador segons es necessite.

Aquesta informació s'ha extret de diverses pàgines com *wikipedia*, *vhdl*, *hardzone* i *planetachatbot*.

2.1.2- Comparació amb els PLC

Un controlador lògic programable, PLC, es una computadora utilitzada en la industria per a automatitzar processos industrials. Els principals avantatges d'utilitzar PLC son:

Un control més precís del procés que s'està realitzant i una resposta ràpida. Permeten afegir modificacions als procés d'una manera còmoda i senzilla. No es excessivament difícil de programar en ells i permet detectar errors que els produeixen durant la execució. Per últim, el sistema es molt més segur i ocupa poc espai.

No obstant això, moltes d'aquests avantatges també els podem trobar utilitzant un controlador encastrat. En canvi, els PLC tenen uns inconvenients diferents a la myRIO:

Primer cal fer un estudi molt complet a l'hora d'instal·lar-los, tenint en compte tots els factors possibles. També el cost del PLC es molt elevat si el nostre objectiu son treballs senzills, com en el nostre cas, que nomes tindria que controlar els tres motors de la impressora. S'estaria desaprofitant la inversió realitzada. Per últim, els PLC necessiten personal qualificat. Tant per al manteniment com per a l'ús, es important que els treballadors sàpiguen com treballar en ells.

La informació sobre els PLC està extreta de les pàgines *scribd*, *gigatecno* i *davidrojasticsplc*.

2.2- Me StepperDriver

El Me Stepper Driver està dissenyat per al control dels motors pas a pas bipolars. Al rebre una senyal, fa que el motor gire un cert angle. Aquest angle pot ser configurat per el usuari, elegint entre diversos modes de funcionament: pas complet, mig pas, 1/4 pas, 1/8 pas i 1/16 pas. Amb estos modes es pot variar els graus que es mou en cada pas, i per tant, els passos necessaris per a realitzar un gir complet. El pas complet es correspon a $1,8^\circ$ per pas, per tant 200 passos per cycle. Per al nostre treball s'ha escollit el mode de 1/4 pas, realitzant 800 passos per cycle.

Pel que fa als components del driver, consta de un connector RJ45 amb 6 pins, 4 d'ells utilitzats com: un pin per a la alimentació, terra, direcció de gir i els passos. El modes son seleccionats amb tres interruptors. També te un potenciòmetre per a ajustar la corrent i modificar el parell del motor.



Figura 8: Vista del driver

Selecció del mode

MS1	MS2	MS3	Mode
L	L	L	Pas complet
H	L	L	Mig pas
L	H	L	1/4 pas
H	H	L	1/8 pas
H	H	H	1/16 pas

Taula 3: Modes del Me Stepper Driver

En aquesta taula es poden observar les posicions dels interruptors per a cada un dels modes, sent L el interruptor baixat i H alçat. La taula i la informació sobre el driver la proporciona MakeBlock en la seua pàgina web.

2.3- Motor pas a pas

El motor utilitzat es un de la marca Makeblock (figura 9). En la pròpia pàgina de l'empresa s'informa de les característiques del motor. Aquest es un motor pas a pas de 2 fases de 1,7 A de corrent per cada fase. L'angle de gir inicial es de $1,8^\circ$, podent-se reduir com ja hem mencionat abans.

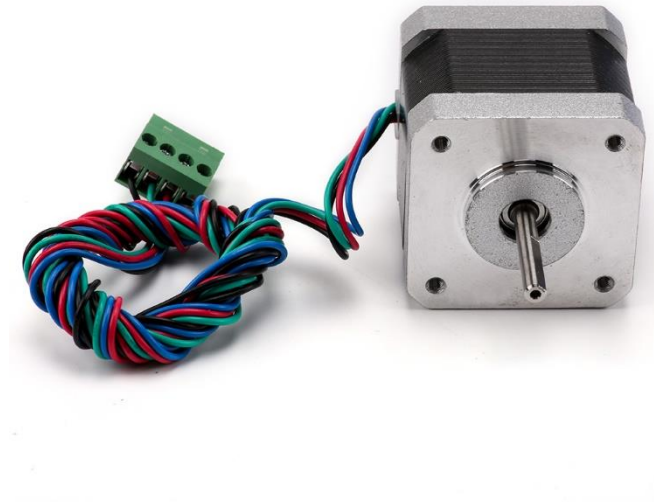


Figura 9: Vista del motor pas a pas

El motor be ja amb els cables necessaris per a la seua connexió (figura 10), que es connectaran a la eixida del driver.

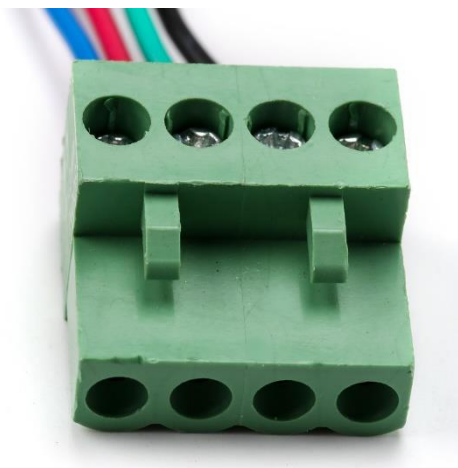


Figura 10: Connexions del motor

2.4- Simulador pont grua

2.4.1- Descripció general

La aplicació Simulink conté un simulador d'un pont grua, que ha sigut utilitzat com a ajuda per a la realització del treball, ja que els principis de funcionament son similars als d'una impressora 3D. Tenim tres eixos, X, Y i Z, cada un controlat de manera independent. En aquest apartat es va a descriure com es el simulador sense cap modificació, mes avant es veuran les modificacions que s'han efectuat.

Per a accedir al simulador cal escriure a la finestra d'ordres de Matlab `vrcrane_panel`. Amb això s'obren les següents finestres:

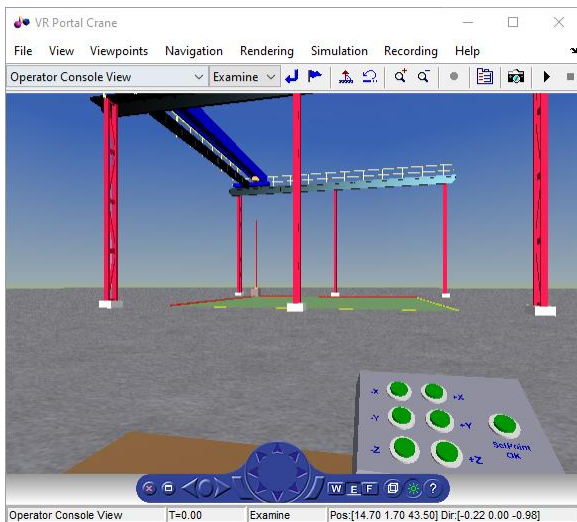


Figura 11: Vista frontal pont grua

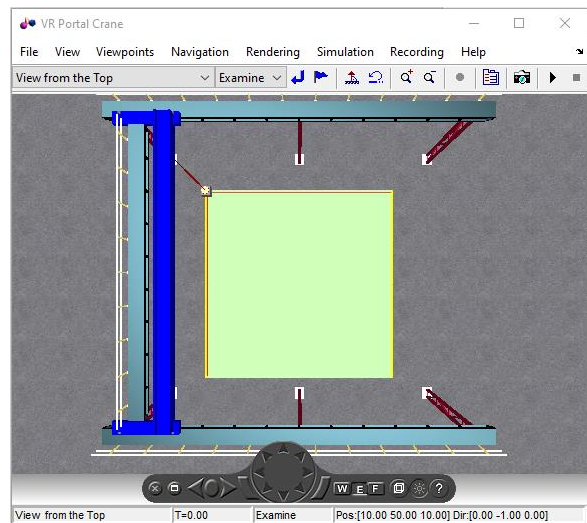


Figura 12: Vista aèria pont grua

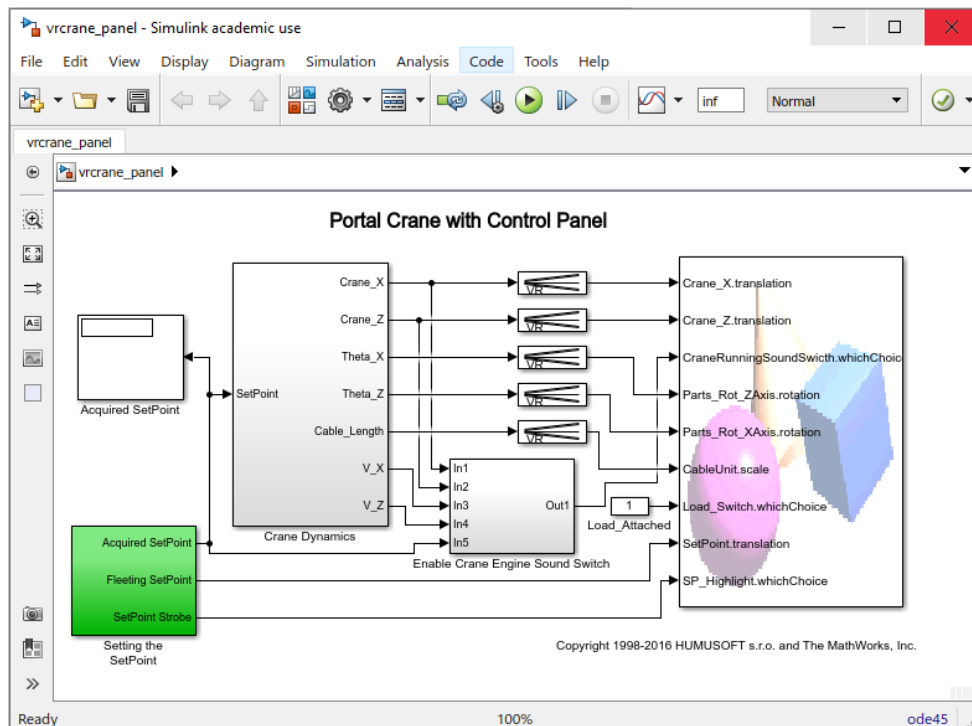


Figura 13: Pantalla principal `vrcrane_panel`

Les figures 11 i 12 son la representació gràfica del pont grua. Amb les fletxes de la part inferior es pot moure el punt de vista i es pot elegir quina de les vistes es vol. La figura 11 te un panell de control, que serveix per a controlar la posició objectiu de la grua i ferla funcionar.

Pel que fa a la figura 13, es el programa complet des de Simulink. Conté quatre blocs, un bloc per a l'adquisició del punt objectiu, un amb els controladors, so i el apartat gràfic.

2.4.2- Adquisició del punt objectiu

A partir dels botons de la vista frontal es modifica el punt objectiu. Al polsar algun d'ells la senyal ix del bloc de gràfics i es transformada amb les funcions que es mostren en la figura 14. Aquestes funcions incrementaran o disminuiran el valor de la posició segons quin botó es pulse, sent 0 la mínima a la que poden arribar i 20 la màxima o 10 en el cas de l'eix Y. Aquest punt objectiu es mostra en les vista com es veu en la figura 15. La grua estarà parada fins que el polse el botó de SetPoint. Una vegada es polse el punt actual serà el objectiu i s'envien les dades al següent bloc per a continuar l'execució.

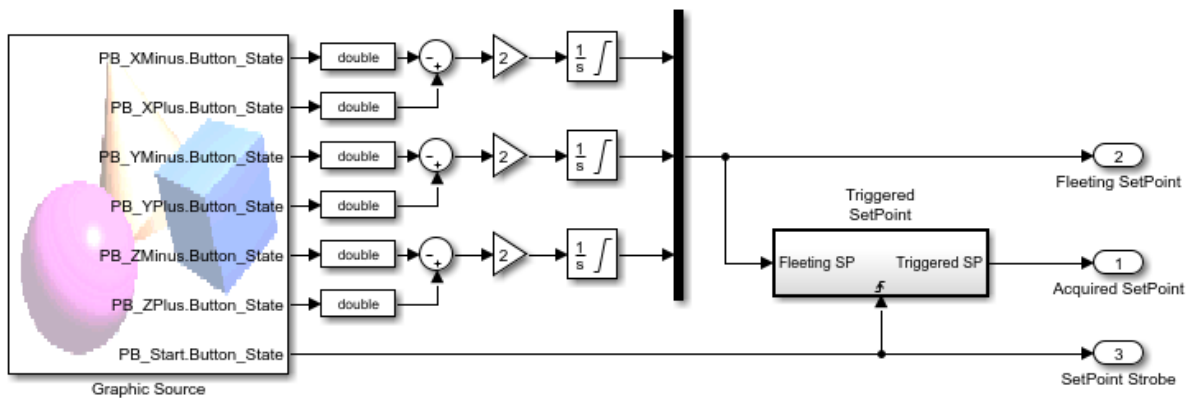


Figura 14: Bloc del punt objectiu

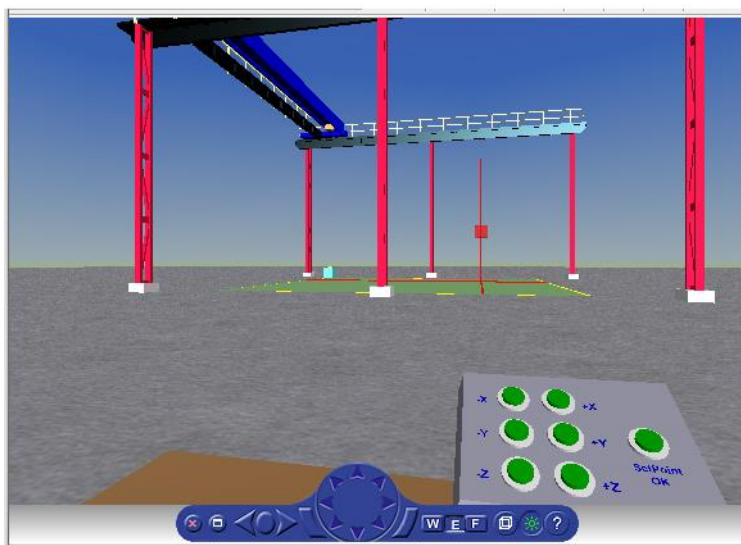


Figura 15: Punt objectiu modificat

2.4.3- Controladors

En aquest bloc es porten a terme tots els càlculs per a obtenir la posició desitjada. Amb el punt objectiu, la massa i la longitud del cable es calculen a cada interval la posició, velocitat i angle per als eixos X i Z dins del bloc de *Crane_Dynamics_X* i *Crane_Dynamics_Z*, i la nova longitud del cable per al eix Y.

Els control dels eixos X i Z es idèntic (figura 17), es realitza amb un control en cascada. L'objectiu d'aquest es millorar el control sobre la variable principal, la posició, gracies a afegir controladors de realimentació. Podem veure que consta de tres controladors tipus P, i un bloc anomenat *Motor_Crane_X*, on es troben les equacions per a obtenir les noves posicions, velocitats i angles, que seran les variables secundaries. A partir d'un valor de referència i d'un error anem obtenint una acció de control, que modifica la posició fins arribar a la desitjada.

Per al eix Y s'utilitza un altre control, molt mes senzill, on a partir d'un integrador de l'error entre la referencia Y transformada i la longitud actual obtenim la nova longitud.

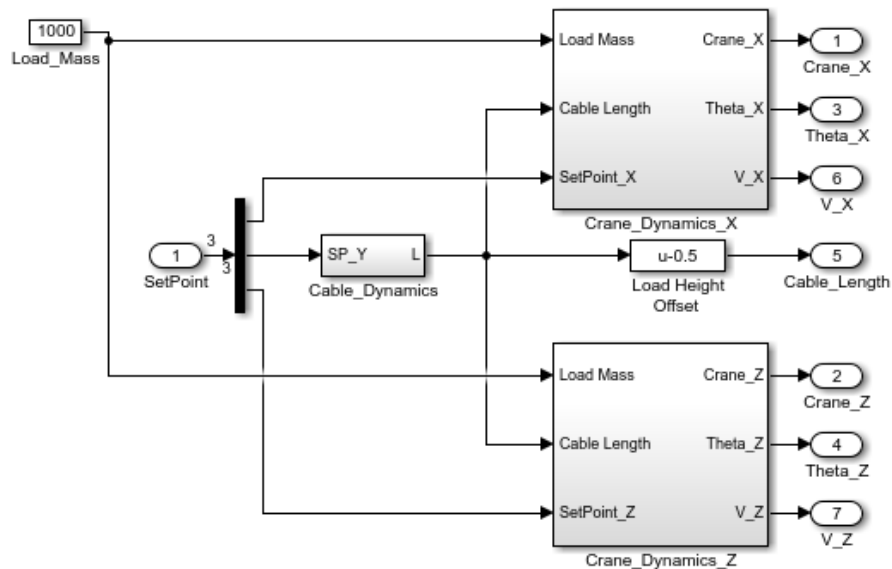


Figura 16: Bloc de controladors

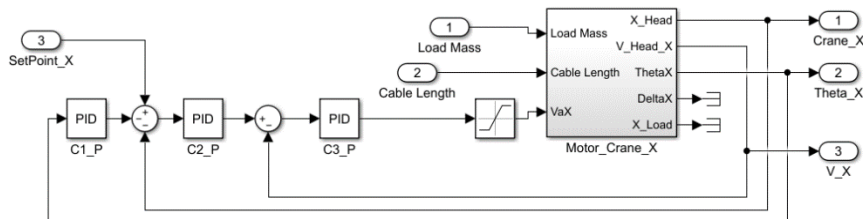


Figura 17: Controlador eix X i Z

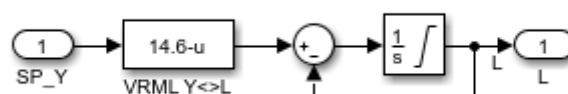


Figura 18: Controlador eix Y

2.4.3.1- Avantatges del control en cascada

Com s'ha dit abans, l'objectiu es la millora del control de la posició. Aquesta millora s'aconsegueix eliminant l'efecte d'algunes perturbacions, fent la resposta del sistema mes ràpida i estable. També s'utilitza per a millorar la dinàmica del llaç de control.

Per tant, com a principal avantatge tenim la velocitat. Amb un control en cascada la resposta es molt mes ràpida. En la figura 19 podem veure una comparació:

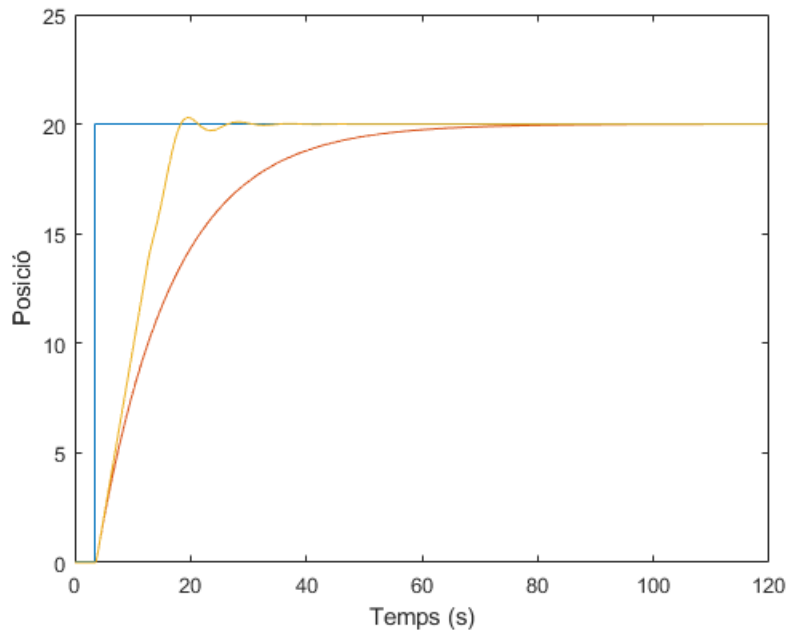


Figura 19: Resposta control en cascada i control simple

En aquesta gràfica tenim la referència, color blau, la resposta del control en cascada, color groc, i la resposta sense cascada, color taronja. Per a fer la comparació hem eliminat els controladors secundaris de la cascada, deixant el valor del principal, 2.5, igual en les dos proves (figura 20).

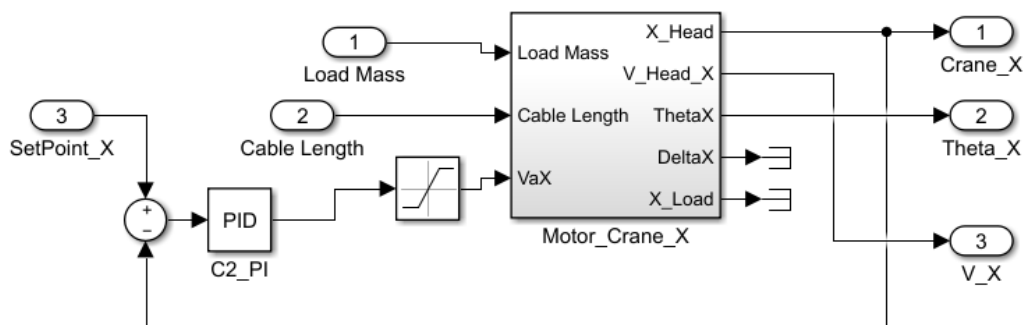


Figura 20: Controladors sense cascada

Es pot veure com la resposta en cascada és clarament molt més ràpida, passant de tardar uns 70 segons en arribar a tardar al voltant de 25 segons.

En segon lloc, el control en cascada també millora la dinàmica del sistema. No obstant això, en la gràfica la resposta amb cascada té una lleugera oscil·lació. Aquesta apareix per culpa del control realitzat sobre l'angle. En la simulació sense cascada el pont grua arriba al punt desitjat però es manté tot el temps oscil·lant al voltant d'eixe punt, degut a com es realitzen els càlculs. Per evitar aquesta oscil·lació s'afegeix també el control sobre l'angle, eliminant-la però creant una lleugera inestabilitat.

Per a observar aquest efecte s'ha realitzat una nova prova (figura 21). En aquest cas ambdues són controls en cascada, però només es controla una variable secundària, la velocitat, línia groga, o l'angle, línia taronja.

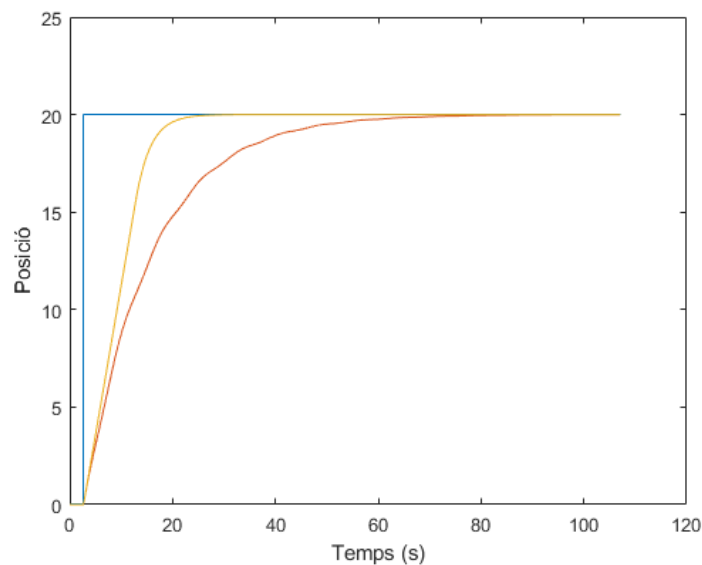


Figura 21: Diferents controls en cascada

Ara també s'han mantingut constants els valors dels controladors, només s'han eliminat la velocitat o l'angle respectivament. Es pot veure com el control sobre l'angle no millora el temps de resposta però evita la oscil·lació infinita en la simulació. Si que s'aprecia com la resposta és més inestable.

D'altra banda en el control sobre la velocitat es veu clarament l'efecte que té el control en cascada sobre el temps de resposta. Com en el cas anterior es redueix d'uns 70 segons a al voltant dels 20 segons, una millora prou quantitativa.

Els controladors de cada prova quedarien igual que el original, eliminant la velocitat (figura 22) o l'angle (figura 23). En cada cas es mante constant el valor principal 2.5, i afegim el control de la velocitat, 5, o el de l'angle, -20, respectivament.

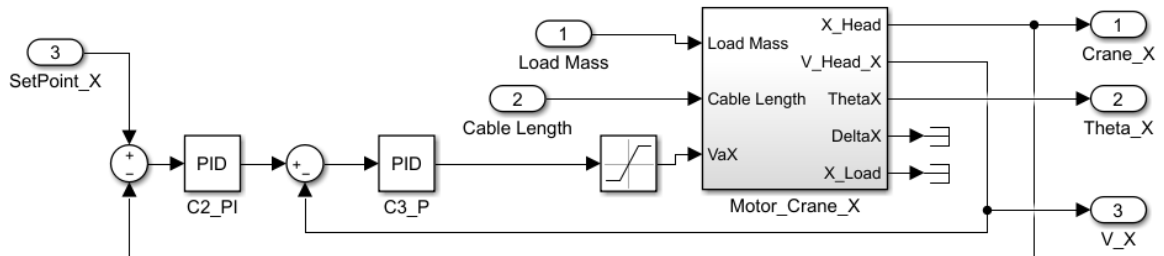


Figura 22: Control en cascada (velocitat)

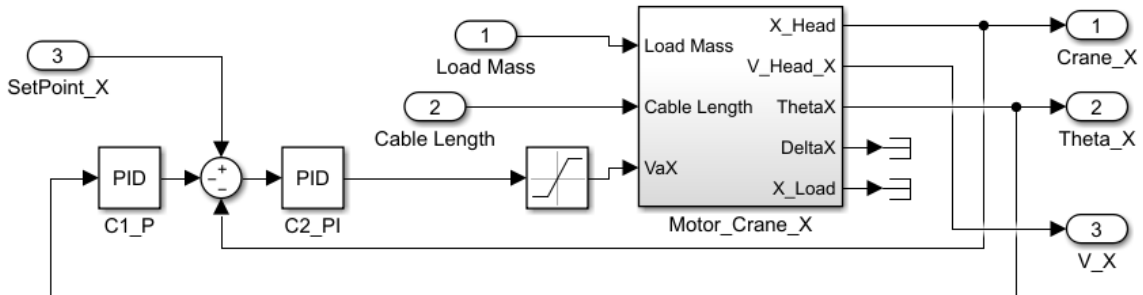


Figura 23: Control en cascada (angle)

3.- Control del Motor

3.1- Muntatge del motor

Primerament, es va connectar la myRIO amb el driver, i aquest al motor (figura 24). Com ja hem comentat, tant la myRIO com el driver es connecten directament a la font d'alimentació. Cal connectar els pins de gir i dels passos del driver, els dos sent entrades digitals. En el nostre cas, es va elegir el grup MSP de la myRIO, elegint el pin DIO0 per al gir i el DIO1 per als passos. S'ha elegit aquest grup ja que es més senzill connectar els cables, però com ja hem vist, els dos tenen entrades digitals i per tant funcionaria igual en els dos.

El motor disposa de cables que es connecten amb el driver mitjançant el connector corresponent. L'eix del motor es connecta a una roda de transmissió de moviment lineal mitjançant una corretja de 40 cm de diàmetre.



Figura 24: Muntatge del motor

3.2- Interfície de control

La interfície del programa es molt senzilla (figura 25). Consta de un controlador per a introduir la distància que desitgem que recorregui, un indicador que ens informa de la posició actual, un LED per a saber si està girant i evitar accidents i un botó per aturar completament el programa.

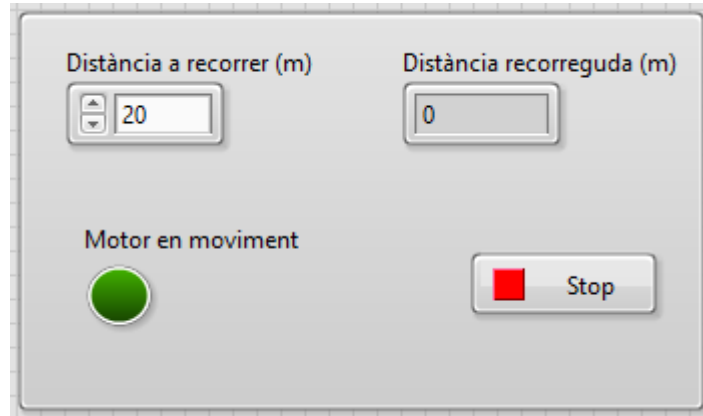


Figura 25: Interfície del control del motor

3.3- Control del motor

Partint d'un dels models per defecte es realitza el programa del motor. L'objectiu es que donant-li un valor el motor es desplaça eixa distància, en el sentit desitjat i una vegada arribe al destí, s'atura esperant la següent instrucció.

El driver no pot donar-li al motor directament la distància, si no que treballa amb els polsos. A cada pols el motor gira un angle determinat, en el nostre cas 0.45° ja que treballem en el mode 1/4 de pas. Per tant, el programa ha de convertir la distància a polsos. Com sabem que una volta completa equival a un desplaçament lineal de 0.125 metres podem obtenir la relació entre distància i polsos, que serà de 6400 polsos/metre.

Pel que fa el funcionament del programa (figura 26), primer introduïm la distància, que es converteix a polsos. El sentit de gir del motor serà determinat pel signe de la distància. Aquets polsos serà el nombre d'iteracions que realitzarà un bucle for, enviant senyal al pin de passos del driver a cada iteració i per tant, movent el motor un pas cada volta. Una vegada finalitzi el bucle, el programa esperarà a que s'introdueixi un nou valor, repetint el bucle a cada nou valor.

El sentit de gir es controla a partir del signe. Al valors positius se'ls assigna una direcció i als negatius la contrària.

Desenvolupament i implementació de controladors dinàmics per a un robot cartesià sobre un controlador encastrat i FPGA

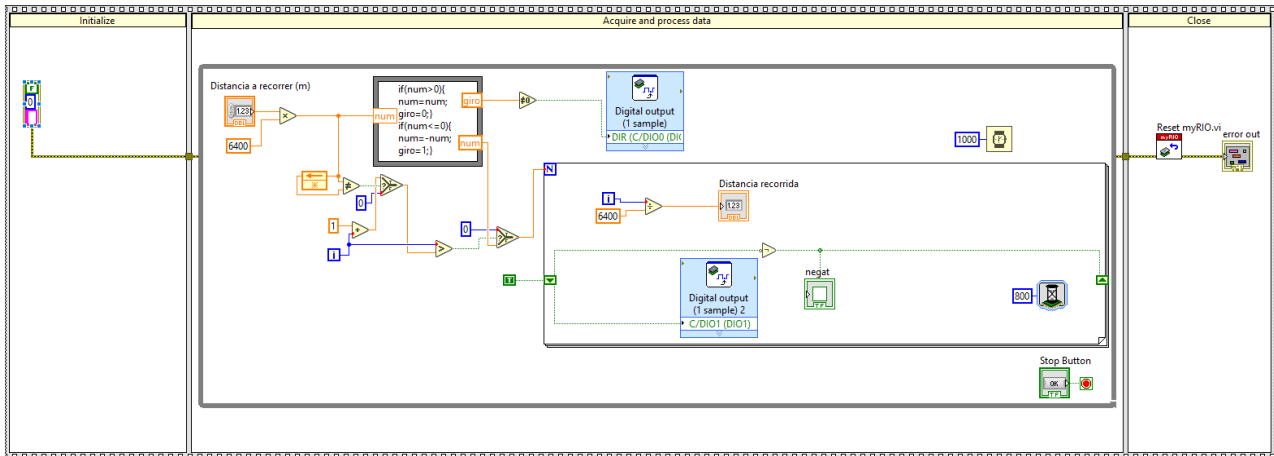


Figura 26: Control del motor

4.- Comunicació TCP/IP

4.1- Enviament de dades

Per a poder realitzar el intercanvi de dades entre LabView i Simulink es necessari utilitzar algun tipus de comunicació, en el nostre cas s'ha fet ús del protocol TCP/IP. En aquesta cal tindre un servidor, que envia la informació, i un client, que la rep. Com les dades son introduïdes en LabView, el servidor ha sigut creat en ell, mentre que Simulink fa el paper de client.

Per a la realització del servidor s'han utilitzat les funcions de *Data Communication*.

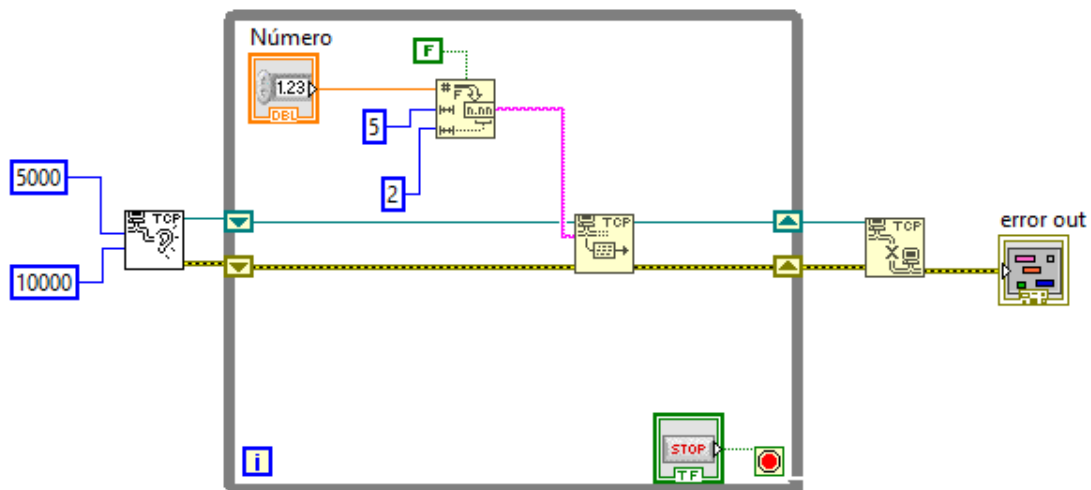


Figura 27: Comunicació TCP/IP LabView

En este exemple (figura 27), s'envia un número amb dos decimals de precisió. LabView no permet enviar números directament, només cadenes de caràcters. Per tant, hem realitzat una conversió a *string*. Es important remarcar que la coma decimal també es un caràcter, per tant al convertir s'ha de tindre en compte. Es per això que enviem cinc dades, dos per a la part entera, dos per a la part decimal i una per a la coma. Una volta convertit es pot enviar a Simulink. En la creació del servidor s'indica el port i el temps d'espera per a connectar-se.

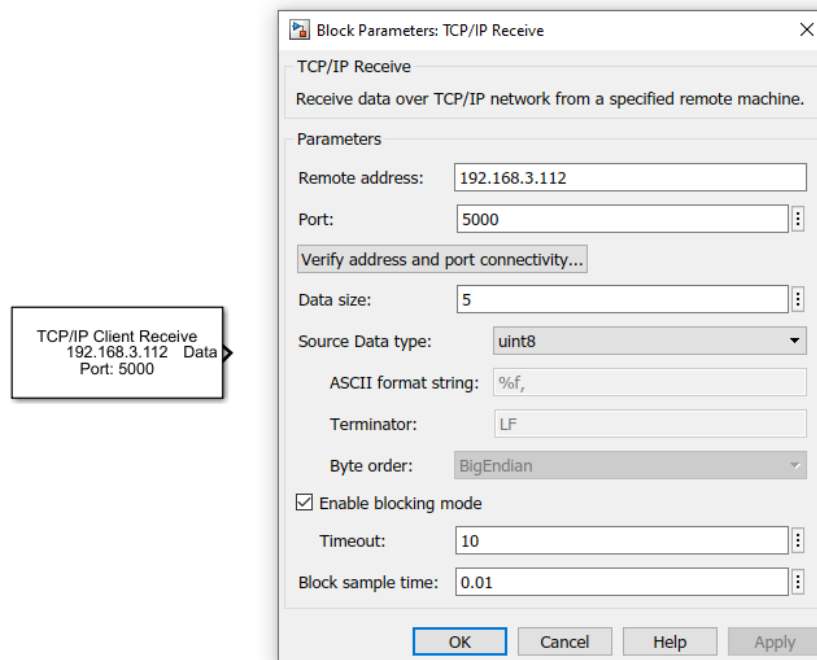


Figura 28: Configuració TCP/IP Client Recieve

Pel que a fa a Simulink, cal instal·lar una extensió anomenada *Instrument Control Toolbox*. En ella es troba la funció *TCP/IP Client Recieve*, que cal configurar (figura 28).

El port ha de coincidir en el indicat en LabView, al igual que la mida de les dades. La adreça serà la IP del ordinador que estem utilitzant.

Una vegada es tenen les dades a Simulink cal tornar a convertir-les a números. Simulink rep les dades en el format *uint8*. Aquest format no es apte per a treballar en el simulador, per a això fem us de dos funcions de Matlab: *char()* i *str2double()*. La funció *char()* ens dona una cadena de caràcters, i amb *str2double()* convertim eixa cadena al valor numèric inicial.

Per últim, es necessari afegir la funció *real()* perquè *str2double()* retorna els valors com a nombres complexos, i el simulador treballa en reals.

4.2- Rebuda de dades

Per al cas contrari, enviar dades des de Simulink a LabView, el procés es similar. Partim d'un número en Simulink i el volem convertir a ASCII per a que LabView pugui entendre'l. Ara no es necessari el bloc de funcions com en el cas anterior, ja que Simulink disposa de dos blocs per a convertir-lo, *To String* i *String To ASCII*. En el exemple (figura 29), partim d'una constant i primer *To String* la converteix a un string. A continuació, el bloc *String To ASCII*, converteix aquest *string* a codi ASCII i ja es pot enviar a LabView amb el bloc de *TCP/IP Send*. És important indicar al bloc *String To ASCII* la longitud de les dades que es volen enviar, en el cas del exemple una. Per últim, com en el cas anterior, cal configurar el bloc de la comunicació indicant la direcció IP i el port al que enviem les dades.

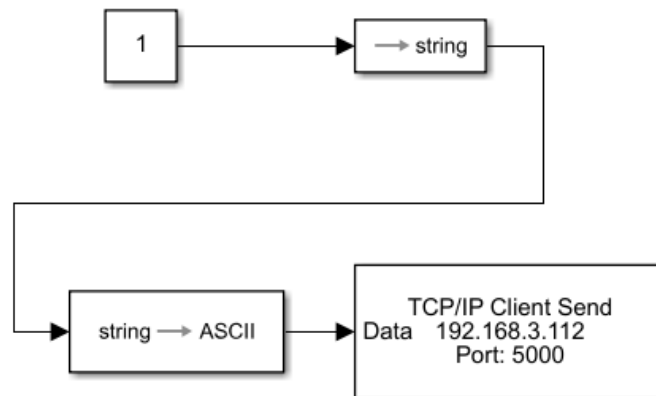


Figura 29: Enviament dades des de Simulink

Pel que fa a LabView afegim un nou bloc als que ja teníem (figura 30). Aquesta funció s'encarrega de llegir les dades. Cal indicar-li el nombre de dades que te que llegir.

Una vegada llegides tenim un *string* amb la informació. Utilitzant un convertidor a decimal recuperem el número i ja podem treballar en ell en normalitat.

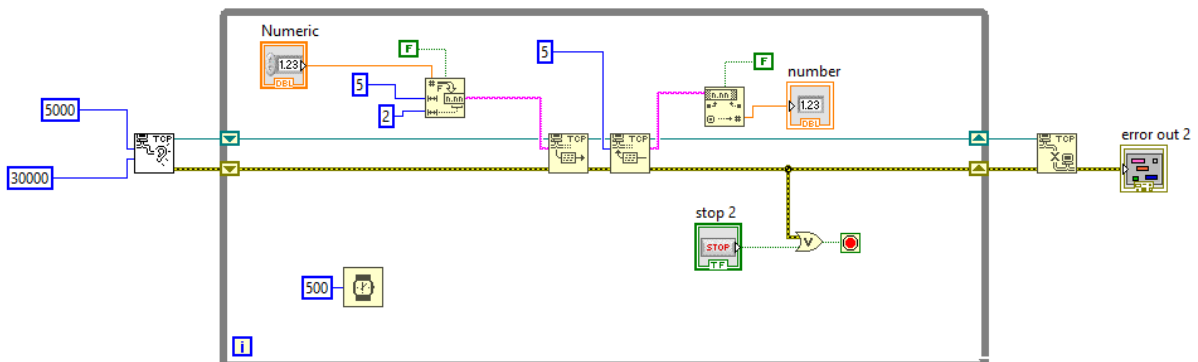


Figura 30: Comunicació TCP/IP LabView

5.- Control del simulador

Una vegada establerta la comunicació cal reproduir el control del simulador en LabView. Per a això es van pensar dos maneres diferents de control. La primera, més similar al control inicial, consistia en reproduir el panell de control en LabView, mentre que la segona imitava millor el comportament real d'una impressora 3D.

5.1.- Botonera implementada a LabView

Com s'ha dit abans, l'objectiu d'aquest control es reproduir exactament el funcionament del simulador original. Per tant, el primer de tot es copiar la botonera. El panell (figura 31) té sis botons per a controlar la posició, dos per a cada eix, un botó per ordenar que vaja al punt desitjat i un botó per a aturar el programa. L'original no disposava d'aquest últim, ja que el propi Simulink permet finalitzar la simulació, però al treballar des de LabView es necessari.

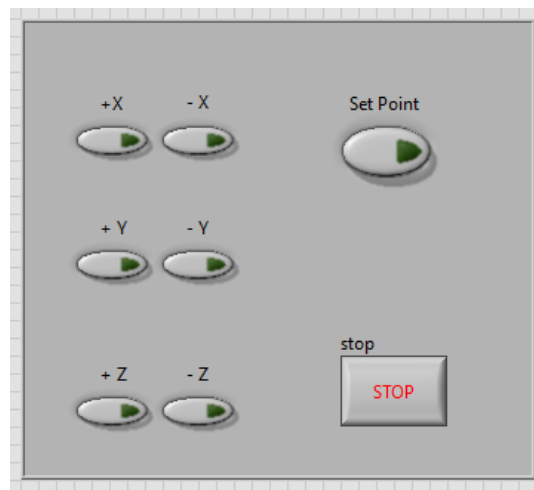


Figura 31: Botonera a LabView

Una volta fet cal realitzar la comunicació amb alguna modificació a la anteriorment explicada. Com els botons son booleans es necessari convertir-los primer a decimal i ja es poden enviar. Cada senyal s'envia per un canal diferent, facilitant la lectura en Simulink però es podria modificar per a que fora tot un sol canal.

Per a la part de Simulink, s'elimina el bloc gràfic dins del punt objectiu i es modifica afegint les set funcions per a rebre la informació (figura 32). Amb les mateixes funcions de Matlab es recupera el valor numèric.

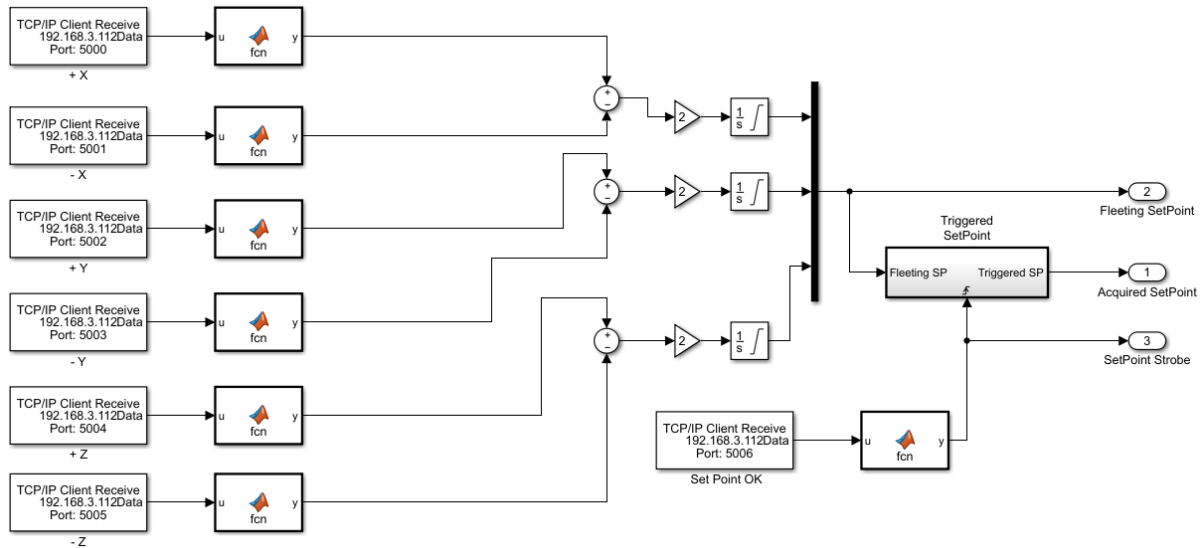


Figura 32: Control per botonera Simulink

5.2.- Controladors implementats a LabView

En el control per referència l'objectiu es enviar punts concrets, la referència, del espai on volem que vagi la grua. A qualsevol canvi d'aquesta referència la grua ha de moure's automàticament.

5.2.1.- Interfície de control

Primer de tot s'explicarà la interfície en LabView (figura 33). De manera similar al control real dels motors, es tenen tres llocs per a introduir la referència, un per a cada eix. Els eixos X i Z funcionen en

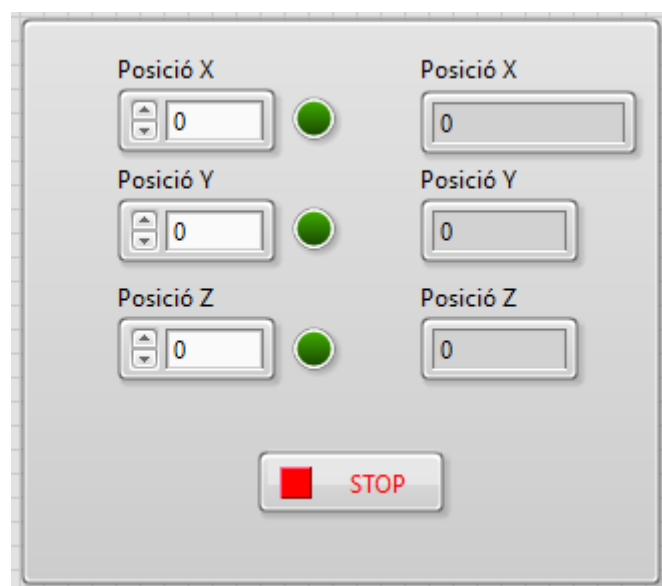


Figura 33: Interfície control per referència

el rang de 0 a 20, mentre que el Y treballa de 0 a 10. També es tenen tres leds que indiquen si estem treballant fora de rang en la seua posició corresponent. En cas que passara, el programa internament ajustaria el valor al rang. La posició actual de la grua apareix en els tres indicadors i es pot veure com es va movent. Per últim, un botó d'aturada per a finalitzar el programa.

5.2.2.- Control eixos X i Z

Primer de tot cal analitzar com funcionen els controladors inicialment. A partir de la referència, la massa de la grua i la longitud del cable s'obté la nova posició, velocitat i angle respecte a l'eix. Com es un bucle tancat, amb les noves dades es calcularan les de la següent iteració mitjançant els tres controladors proporcionals en cascada.

Una vegada ja es sap com funciona es reproduïx en LabView. El nostre objectiu serà enviar l'acció de control a Simulink, on es calcularan els efectes que tindrà sobre la posició, velocitat i angle, i recuperar eixos nous valors per a poder calcular la següent acció de control.

El control complet del eix X seria el següent (figura 34), però l'analitzarem per parts.

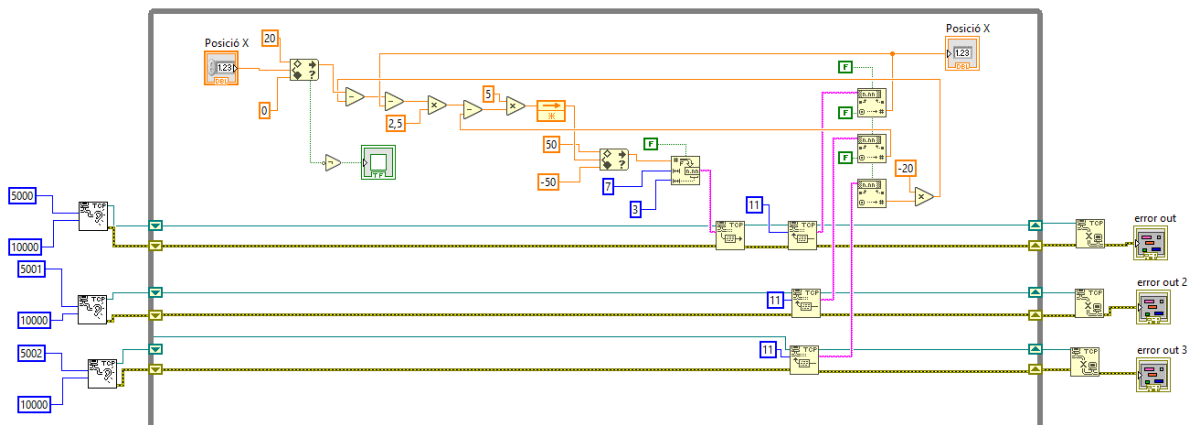


Figura 34: Control LabView eix X

En primer lloc, s'introdueix la posició desitjada i es calcula l'acció de control (figura 35). La posició es limita als valors ja mencionats prèviament i operant amb els valors anteriors de posició, velocitat i angle obtenim l'acció. Els controladors P tenen uns valors de -20, 2.5 i 5.

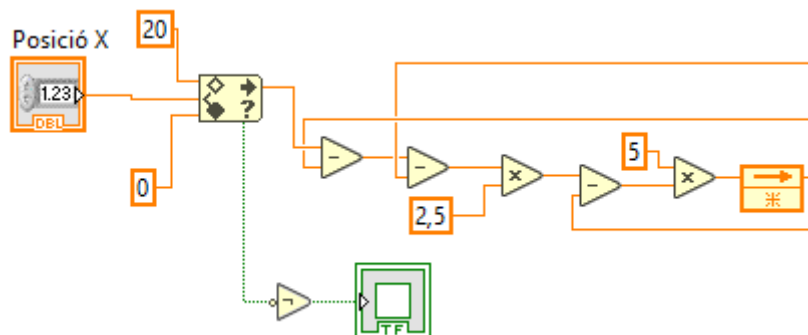


Figura 35: Obtenció de l'acció de control

Aquesta acció es satura entre -50 i 50 i s'envia a Simulink per comunicació TCP/IP (figura 36). És important tindre en compte que poden ser valors negatius, per tant cal donar un caràcter més per al signe. També indicar si estem gastant el punt o la coma com a indicador decimal. Es pot augmentar la precisió del valor però amb tres decimals el programa funciona perfectament.

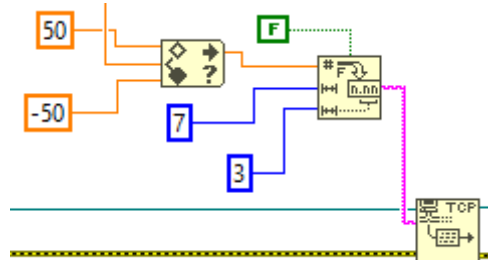


Figura 36: Enviament dades a Simulink

Per últim es recuperen els nous valors mitjançant tres ports diferents (figura 37) per a utilitzar-los en la següent iteració. El valor la posició es el que apareixerà en el indicador. En aquest cas, llegim 11 caràcters per a tindre la màxima precisió possible i fer un control més exacte. Els valors son posició, velocitat i angle en orde descendent.

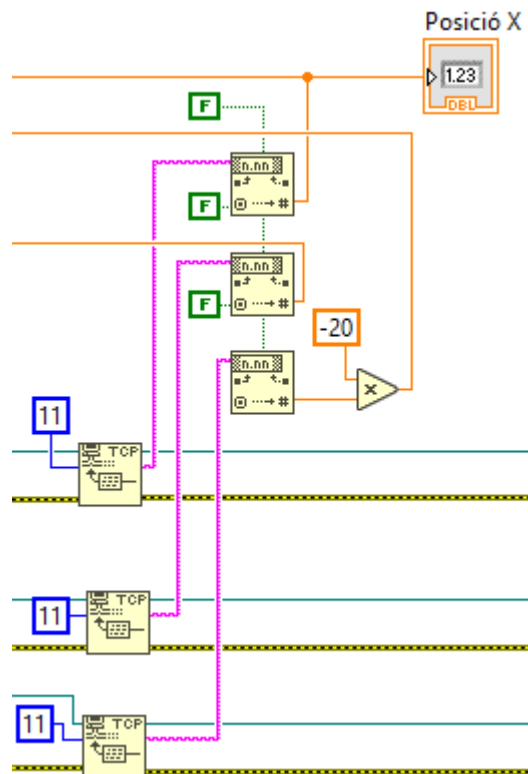


Figura 37: Recuperació de dades

El control de l'eix Z es exactament igual, però amb diferents ports per a la comunicació.

5.2.3.- Control eix Y

L'eix Y es molt més senzill de controlar (figura 38). Simplement s'introdueixen les dades i s'envien a Simulink, on es calcula la nova altura a partir d'una equació. Aquesta altura serà necessària en el càlcul de les accions de control de la resta d'eixos. Una vegada calculada es recupera en LabView per a poder-la mostrar. Cal dir que el simulador realment no treballa en valors d'altura, si no en valors de longitud del cable. Es per això que a pesar de que per a indicar les posicions utilitza valors de 0 a 10, aquests valors corresponen a 14,1 i 4,1 respectivament com a longitud de cable. Per tant per a facilitar la comprensió per part de l'usuari es fa un càlcul per que es mostren com a valors de 0 a 10.

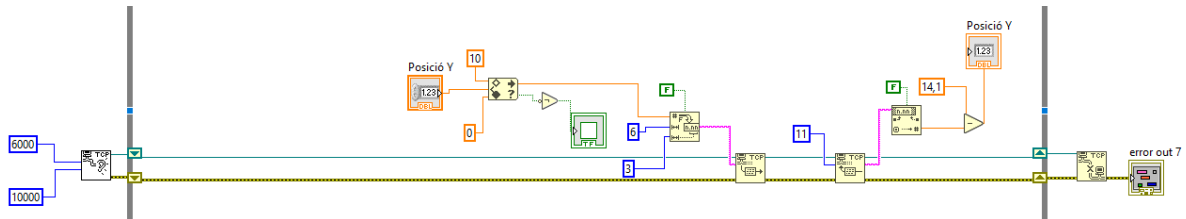


Figura 38: Control eix Y LabView

5.2.4.- Modificacions a Simulink

Pel que fa a Simulink, el bloc d'adquisició del punt objectiu canvia totalment (figura 39). S'elimina tota la part gràfica, i es tenen les tres funcions per a rebre les dades des de LabView. Aquestes dades van al bloc de controladors que també es modifica. Col·loquem un valor constant unitari en la eixida del SetPoint per a que l'apartat gràfic i de so no done problemes però no afecta en res al nostre objectiu.

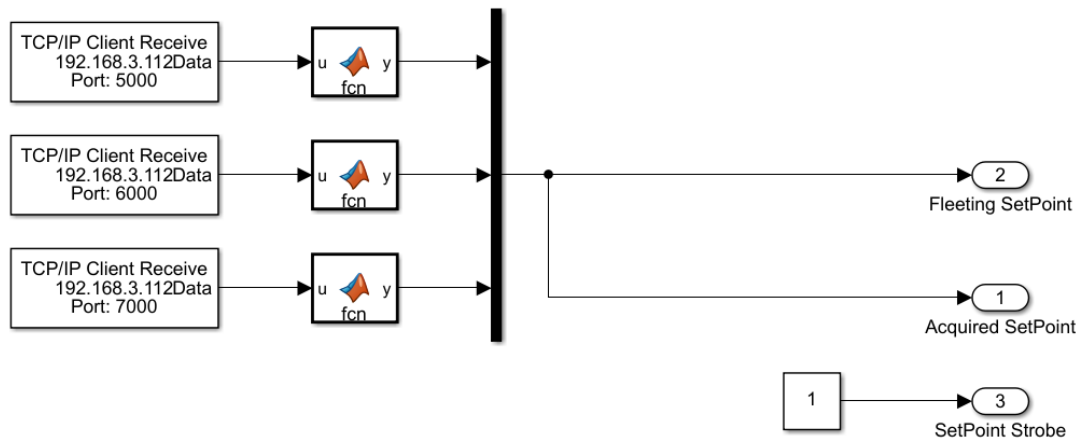


Figura 39: Nou bloc d'adquisició del punt objectiu

En el bloc de controladors X i Z s'elimina el controlador en si deixant les equacions per a calcular la nova posició i s'afegeixen les funcions anteriorment comentades per a enviar dades a LabView (figura 40). Tant en la rebuda com en el manament de dades tenen que coincidir la quantitat amb la de LabView. Per tant, en el nostre cas rebem 7 caràcters i enviem 11.

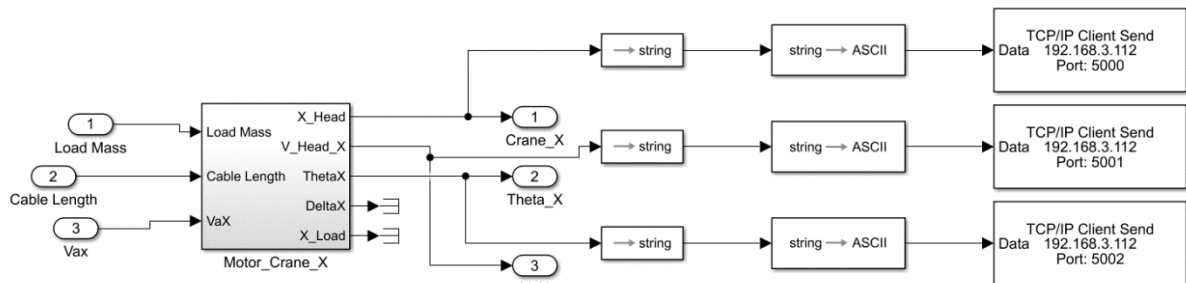


Figura 40: Nou bloc de controladors eix X

Pel que fa al eix Y, simplement afegim les funcions per a enviar les dades una vegada calculada la nova altura.

5.2.5.- Verificació dels resultats

Una vegada es tenen els controladors es necessari comprovar que tot funciona correctament. Per a això es porta a la grua a diferents punts i s'analitzaran els resultats. Primer de tot, com els eixos X i Z son iguals només es prova un dels dos, en el nostre cas el X i s'obté la següent gràfica:

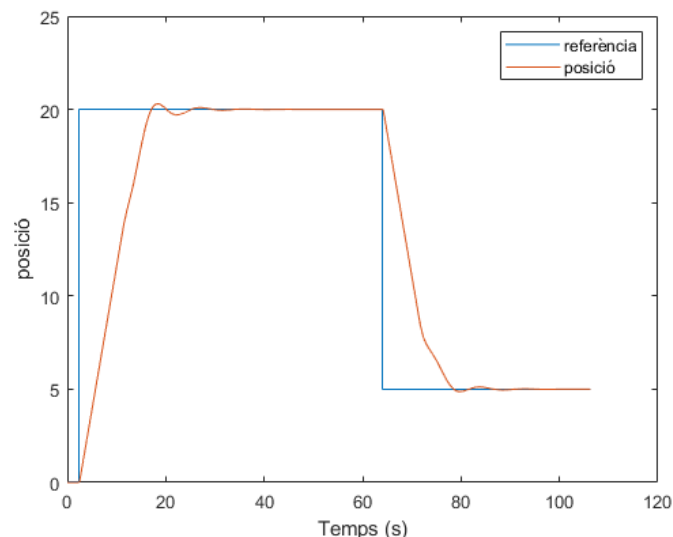


Figura 41: Gràfica de la posició X

En aquest experiment s'han donat dos referències, la línia blava, primer partint de 0 al màxim, 20, i després tornar a 5. La línia taronja representa la posició real i es pot veure com si que segueix la

referència. Cal dir que el temps que apareix en l'eix X es el temps calculat per Simulink, en la vida real tarda aproximadament uns 10 segons en fer tot el cicle.

Per al eix Y es realitza una prova similar. Partint de 0 es porta al màxim 10, i després es torna a 4.

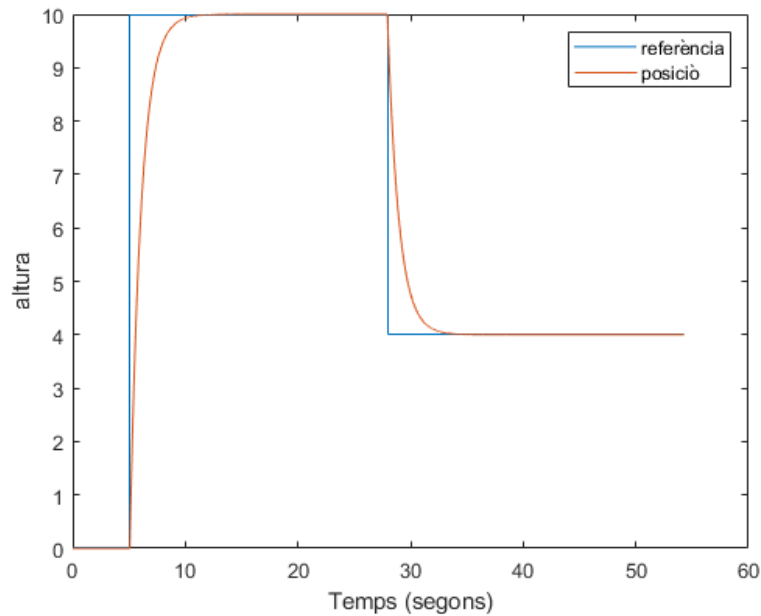


Figura 42: Gràfica de l'eix Y

Aquest control es més ràpid que els anteriors, ja que s'assoleix la referència molt més abans. També destacar la absència d'ondulacions, reduint en gran manera l'error.

5.2.6.- Ajust dels controladors de l'eix X

Per tant de veure com afecta modificar els paràmetres dels controladors s'han realitzat diversos experiments, primer variant cada paràmetre de manera aïllada i després tots els paràmetres a la vegada.

	P1	P2	P3
Original	-20	2.5	5
Exp. 1	-10	2.5	5
Exp. 2	-30	2.5	5
Exp. 3	-20	5	5
Exp. 4	-20	2	5
Exp. 5	-20	2.5	7.5
Exp. 6	-20	2.5	2.5
Exp. 7	-30	5	7.5
Exp. 8	-10	2.5	2.5

Taula 4: Valors dels controladors

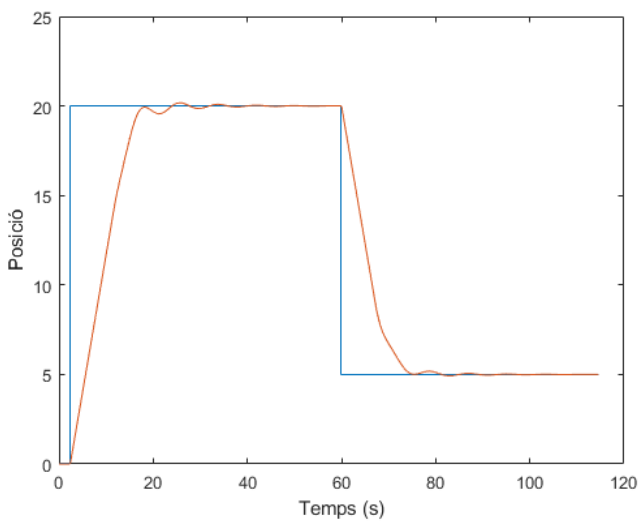


Figura 43: Experimento 1

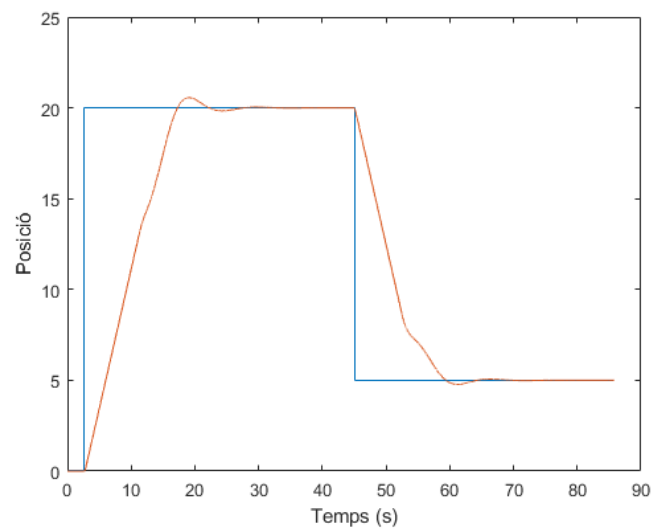


Figura 44: Experimento 2

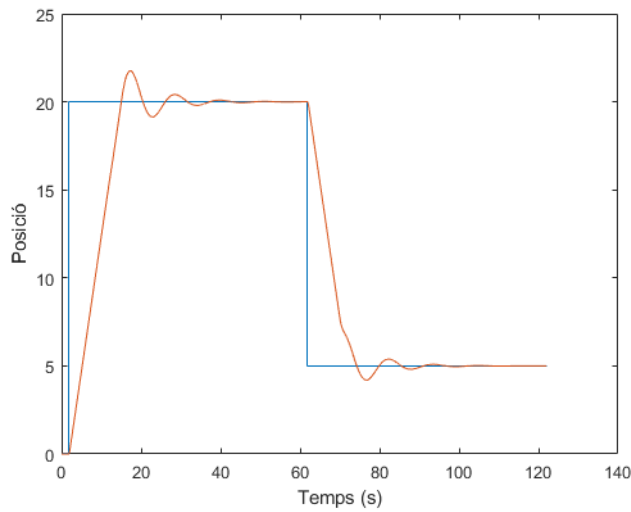


Figura 45: Experimento 3

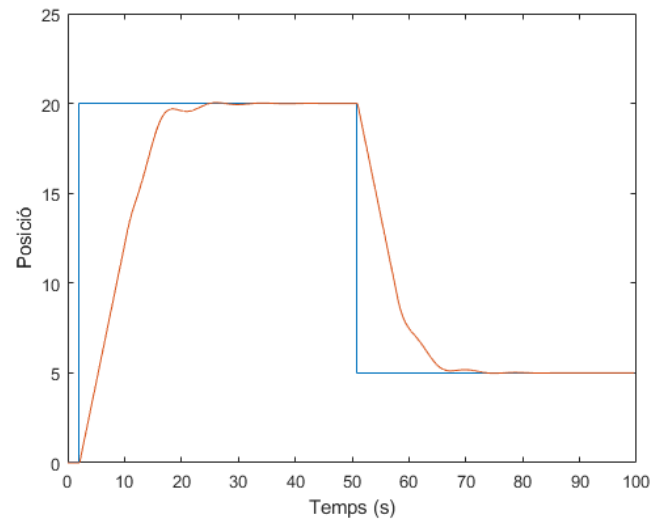


Figura 46: Experimento 4

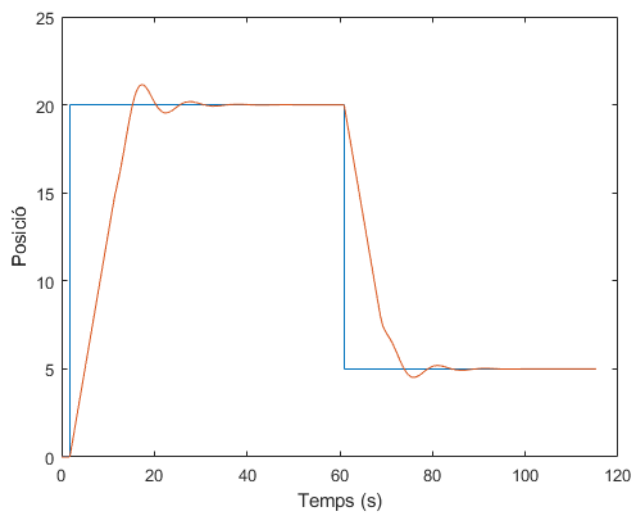


Figura 47: Experimento 5

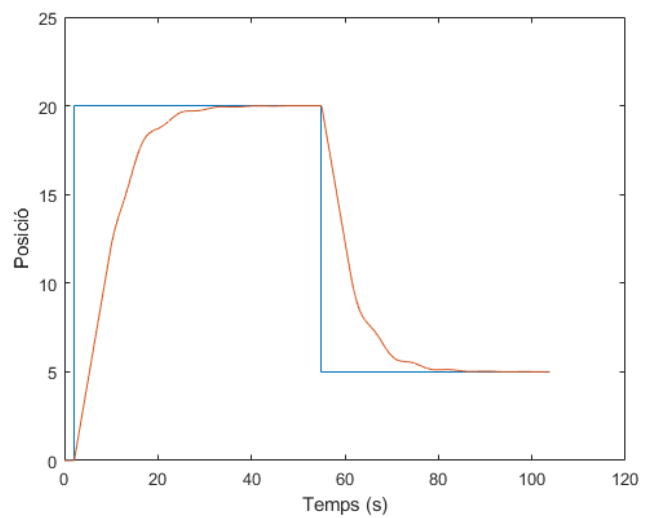


Figura 48: Experimento 6

Es pot veure com en augmentar el valor de cada paràmetre de forma independent, apareixen majors oscil·lacions, principalment en els que afecten a la velocitat i la posició (figura 45 i figura 47). En canvi, al disminuir la oscil·lació disminueix notablement, no obstant això es tarda més en arribar al punt desitjat.

Per a observar aquests comportaments, en els dos últims experiments (figura 49 i figura 50) variem els valors tots a la vegada. En la figura 49 els valors han sigut augmentats i s'observen més oscil·lacions. D'altra banda, en la figura 50 han estat disminuïts i les oscil·lacions pràcticament han desaparegut a canvi de retardar l'arribada al punt desitjat.

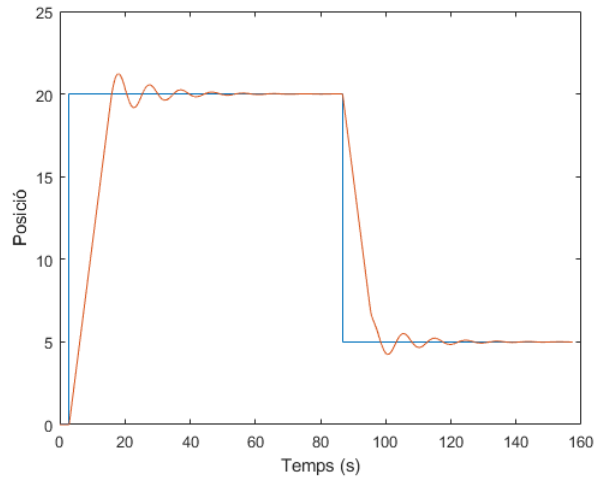


Figura 49: Experiment 7

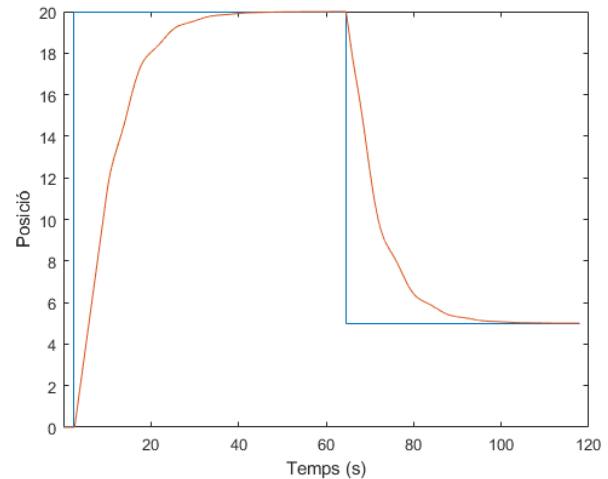


Figura 50: Experiment 8

Com a conclusió, els valors inicials eren prou acurtats, podent-se millorar reduint-los. No obstant això, aquest millora vindria acompanyada d'un increment dels temps de treball que podria produir problemes. Per tant caldria valorar que té més pes si la velocitat o l'absència d'oscil·lacions segons els nostre objectiu. Dir també que reduir en gran manera el control sobre el angle pot originar problemes en el simulador, fent que la simulació arribi al punt però no acabe mai de balancejar-se.

6.- Conclusions

Com a conclusió del treball es pot afirmar que s'han aconseguit els objectius buscats.

Primer de tot, s'ha pogut realitzar la comunicació entre LabView i Simulink, solucionant tots els errors que podien aparèixer i comprnent com treballa cada un dels programes.

En segon lloc, el control d'un robot cartesià s'ha pogut realitzar, amb diferents versions del mateix i s'ha compres el funcionament, ja que ha sigut indispensable per a poder realitzar el control. També s'han analitzats els resultats per a poder confirmar el seu bon funcionament.

Pel que fa als motors pas a pas, s'ha assolit un bon nivell de coneixement del seu funcionament, permetent realitzar un control d'un d'aquests i veure com complia amb el propòsit desitjat. També ha estat introduïda la myRIO com a exemple de controlador encastrat.

Deguda a la situació actual, no ha estat possible veure implementats els controls sobre la impressora ni treballar físicament amb ella. Es per això que quedaria pendent com una possible ampliació del treball realitzar la implementació amb les proves corresponents.

Per últim, també podria ser nous treballs intentar ajustar els controls del pont grua o de la impressora, per tant de tindre un funcionament encara millor.

7.- Bibliografia

- <https://www.makeblock.com/project/me-stepper-driver>, de MakeBlock
- https://www.makeblock.es/productos/motor_paso_a_paso_42/, de MakeBlock
- User Guide and Specifications NI myRIO-1900, de National Instruments
- <https://www.hta3d.com/es/blog/tipos-de-controladores-que-se-usan-frecuentemente-en-las-impresoras-3d>
- <http://deimpresoras3d.com/stepper-drivers-de-impresoras-3d/>
- https://en.wikipedia.org/wiki/Field-programmable_gate_array
- <https://vhdl.es/fpga/>
- <https://hardzone.es/reportajes/que-es/fpga-caracteristicas-utilidad/>
- <https://planetachatbot.com/qu%C3%A9-es-una-fpga-y-por-qu%C3%A9-jugar%C3%A1n-un-papel-clave-en-el-futuro-e76667dbce3e>
- <https://es.scribd.com/document/258185893/Ventajas-y-Desventajas-de-Los-Plc>
- <http://gigatecno.blogspot.com/2013/02/ventajas-y-desventajas-de-un-plc.html>
- <https://davidrojasticsplc.wordpress.com/2009/01/14/ventajas-y-desventajas/>

DOCUMENT II: PRESSUPOST

Índex del pressupost

1.- Preus materials i mà d'obra.....	pàg. 39
2.- Preus descompostos.....	pàg. 40
3.- Preus unitaris.....	pàg. 42
4.- Preu base de licitació.....	pàg. 43

1.- Preus materials i mà d'obra

A continuació es mostra una llista amb el cost dels materials, hardware i software necessaris per a realitzar aquest treball:

	Preu
Controlador myRIO-1900	602 €
Driver MakeBlock Me Stepper	17,99 €
Motor pas a pas MakeBlock 42BYG	24,95 €
Cable de 1,5 mm ² de diàmetre	0,13 €/m
Cable amb pinça de cocodril	0,65 €
Llicència de 1 any Matlab	800 €
Llicència de 1 any de Simulink	1200 €
Llicència de 1 any de Instrument Control Toolbox	400 €
Llicència 1 any de LabView	909 €
Mòdul de LabView Real-Time	811,25 €
Mòdul de LabView FPGA	811,25 €

El preu per hora d'un graduat en Enginyeria en Tecnologies Industrials es de 30 €/h, incloent la cotització a la seguretat social.

2.- Preus descompostos

Nº	Descripció de les unitats d'obra	Mesura	Preu	Import
01	Muntatge dels motors de la impressora 3D			
u.	Controlador myRIO-1900	1	602,00 €	602 €
u.	Driver MakeBlock Me Stepper	3	17,99 €	53,97 €
u.	Motor pas a pas MakeBlock 42BYG	3	24,95 €	74,85 €
u.	Cable amb pinça de cocodrill	2	0,65 €	1,30 €
m	Cable de 1,5 mm2 de diàmetre	1	0,13 €	0,26 €
h	Graduat en Eng. en Tecnologies Industrials	40	30,00 €	1.200,00 €
%	Costos indirectes	2	1.932,37 €	38,65 €
%	Gastes complementaris	3	1.971,02 €	59,13 €
	TOTAL			2.030,15 €
02	Programació del control dels motors a LabView			
h	Graduat en Eng. en Tecnologies Industrials	40	30,00 €	1.200,00 €
%	Costos indirectes	2	1.200,00 €	240,00 €
%	Gastes complementaris	3	1.440,00 €	43,20 €
	TOTAL			1.483,20 €
03	Comunicació de dades mitjançant les funcions TCP/IP			
h	Graduat en Eng. en Tecnologies Industrials	45	30,00 €	1.350,00 €
%	Costos indirectes	2	1.350,00 €	27,00 €
%	Gastes complementaris	3	1.377,00 €	41,31 €
	TOTAL			1.418,31 €
04	Anàlisi de simulador del pont grua			
h	Graduat en Eng. en Tecnologies Industrials	35	30,00 €	1.050,00 €
%	Costos indirectes	2	1.050,00 €	21,00 €
%	Gastes complementaris	3	1.071,00 €	32,13 €
	TOTAL			1.103,13 €
05	Programació del control per punt objectiu			
h	Graduat en Eng. en Tecnologies Industrials	35	30,00 €	1.050,00 €
%	Costos indirectes	2	1.050,00 €	21,00 €
%	Gastes complementaris	3	1.071,00 €	32,13 €
	TOTAL			1.103,13 €

Desenvolupament i implementació de controladors dinàmics per a un robot cartesià sobre un controlador encastrat i FPGA

Nº	Descripció de les unitats d'obra	Mesura	Preu	Import
06	Programació del control per referència			
h	Graduat en Eng. en Tecnologies Industrials	45	30,00 €	1.350,00 €
%	Costos indirectes	2	1.350,00 €	27,00 €
%	Gastes complementaris	3	1.377,00 €	41,31 €
	TOTAL			1.418,31 €
07	Llicències de LabView i Matlab/Simulink			
u.	Llicència de 1 any de Matlab	1	800,00 €	800,00 €
u.	Llicència de 1 any de Simulink	1	1.200,00 €	1.200,00 €
u.	Llicència de 1 any de Instrument Control Toolbox	1	400,00 €	400,00 €
u.	Llicència de 1 any de LabView	1	909,00 €	909,00 €
u.	Mòdul de LabView Real-Time	1	811,25 €	811,25 €
u.	Mòdul de LabView FPGA	1	811,25 €	811,25 €
%	Costos indirectes	2	4.931,50 €	98,63 €
%	Gastes complementaris	3	5.030,13 €	150,90 €
	TOTAL			5.181,03 €
08	Redacció dels documents			
h	Graduat en Eng. en Tecnologies Industrials	60	30,00 €	1.800,00 €
%	Costos indirectes	2	1.800,00 €	36,00 €
%	Gastes complementaris	3	1.836,00 €	55,08 €
	TOTAL			1.891,08 €

3.- Preus Unitaris

Nº	Descripció de les unitats d'obra	Preu unitari
01	Muntatge dels motos de la impressora 3D	2.030,15 €
02	Programació del control dels motors en LabView	1.483,20 €
03	Comunicació de les dades mitjançant funcions TCP/IP	1.418,31 €
04	Anàlisi del simulador del pont grua	1.103,13 €
05	Programació del control per punt objectiu	1.103,13 €
06	Programació del control per referència	1.418,31 €
07	Llicències de LabView i Matlab\Simulink	5.181,03 €
08	Redacció dels documents	1.891,08 €

4.- Pressupost base de licitació

UNITATS D'OBRA:

Muntatge dels motos de la impressora 3D	2.030,15 €
Programació del control dels motors en LabView	1.483,20 €
Comunicació de les dades mitjançant funcions TCP/IP	1.418,31 €
Anàlisi del simulador del pont grua	1.103,13 €
Programació del control per punt objectiu	1.103,13 €
Programació del control per referència	1.418,31 €
Llicències de LabView i Matlab\Simulink	5.181,03 €
Redacció dels documents	1.891,08 €
Pressupost d'execució material	15.628,34 €
Gastes generals (13%)	2.031,68 €
Benefici Industrial (6%)	937,70 €
Pressupost d'execució per contracta	18.597,72 €
IVA (21%)	3.905,52 €
Pressupost base de licitació	22.503,24 €

El pressupost total ascendeix a la quantitat de:

VINT-I-DOS MIL CINC-CENTS TRES EUROS AMB VINT-I-QUATRE CÈNTIMS

