



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA TÈCNICA
SUPERIOR ENGINYERS
INDUSTRIALS VALÈNCIA

TREBALL FINAL DE GRAU EN ENGINYERIA EN TECNOLOGIES INDUSTRIALS

DESENVOLUPAMENT D'UNA INTERFÍCIE AMIGABLE BASADA EN LABVIEW PER AL CONTROL I SUPERVISIÓ DE ROBOTS PARA LLELS DE REHABILITACIÓ

AUTOR: MIGUEL CUCÓ BOILS

TUTORA: MARINA VALLÉS MIQUEL

Curs Acadèmic: 2017-18

AGRAÏMENTS

A la meua tutora Marina Vallés

A Ángel Valera

A Jose Miguel Fabrich

A la meua família, sobretot a la meua mare

I als meus companys i companyes del grau

RESUMEN

Con la creciente demanda de automatización de procesos por parte de la industria i la consecuente bajada de precios en los apartados de control y robótica, la medicina ha dado un paso adelante para aliarse con la tecnología y conseguir unos tratamientos mucho más precioso y personalizados, manteniendo los costes anteriores o incluso reduciéndolos.

Con esta premisa, el proyecto se desarrolla alrededor de un robot paralelo de rehabilitación de tobillo de tres grados de libertad. No obstante, los profesionales que producen este tipo de tecnologías y los profesionales que las utilizan no tienen los mismos conocimientos, y se hace necesario el desarrollo de una interfaz de usuario que permita, a los futuros clientes, la utilización de esta tecnología de una forma más sencilla, sin tener que saber de programación o del funcionamiento interno del propio robot.

Con el objetivo de especificar más, el robot en el que se ha centrado el proyecto utiliza el lenguaje de programación ROS, para permitir una programación más flexible, con el middleware OrocOS, para proporcionar un control sólido del sistema en tiempo real.

Con todos estos factores en mente, el objetivo de este trabajo final de grado es cubrir esta necesidad, desarrollar una interfaz lo más sencilla posible sin perjudicar en exceso las posibilidades del robot. Para programarla se ha recurrido a LabVIEW, por ser una herramienta enfocada al desarrollo de interfaces que, además, cuenta con una librería gratuita que permite la comunicación con ROS.

Palabras Clave: interfaz de usuario, LabVIEW, ROS, robot paralelo.

RESUM

Amb la creixent demanda d'automatització de processos per part de la indústria i la consegüent baixada de preus en els apartats de control i robòtica, la medicina ha pegat un pas endavant per tal d'aliar-se amb la tecnologia i aconseguir uns tractaments molt més precisos i personalitzats, mantenint els costos anteriors o inclòs reduint-los.

Amb aquesta premissa, el projecte es desenvolupa al voltant d'un robot paral·lel de rehabilitació de turmell de tres graus de llibertat. No obstant, els professionals que desenvolupen aquests tipus de tecnologies i els professionals que les utilitzen no tenen els mateixos coneixements, i es fa necessari el desenvolupament d'una interfície d'usuari que permeti, als futurs clients, la utilització d'aquesta tecnologia d'una manera més senzilla, sense haver de saber de programació o del funcionament intern del propi robot.

Per tal d'especificar més, el robot en el que s'ha enfocat el projecte utilitza el llenguatge de programació ROS, per permetre una programació més flexible, amb el middleware Orocòs, per tal de proporcionar un control sòlid del sistema en temps real.

Amb tots aquests factors en ment, l'objectiu d'aquest treball final de grau és cobrir aquesta necessitat, desenvolupant una interfície el més senzilla possible sense perjudicar en excés les possibilitats del robot. Per tal de programar-la s'ha recorregut a LabVIEW, per ser una ferramenta enfocada al desenvolupament d'interfícies que, a més a més, consta d'una llibreria gratuïta que permet la comunicació amb ROS.

Paraules clau: interfície d'usuari, LabVIEW, ROS, robot paral·lel.

ABSTRACT

The industries' growing demand for the automation processes and the resulting lowering prices in control and robotics sections have helped medicine to make a step forward: working together with technology and achieving more accurate and personalized treatments, keeping the previous costs or even reducing them.

Under that premise, the project is developed around a parallel robot for ankle rehabilitation of three degrees of freedom. However, the professionals who develop these type of technologies and the professionals who use them do not have the same knowledge. Thus, it becomes necessary to create a user interface that allows future clients to use this technology in a simpler way, without necessarily knowing about programming or the internal working of the robot.

To be more specific, the robot in which the project is focused uses ROS as the programming language, to allow a more flexible programming, with the middleware Orocos, to offer a tighter control in real time.

Considering all these factors, the objective of this Final Undergraduate Project is covering this need through developing an interface as simple as possible, without being excessively prejudicial for the robot possibilities. LabVIEW is the tool used to program this interface, on account of being focused on developing interfaces and, in addition, offering a free library that allows the communication with ROS as well.

Keywords: user interface, LabVIEW, ROS, parallel robot.

ÍNDEX

DOCUMENTS CONTINGUTS AL TFG

- Memòria
- Pressupost

ÍNDEX DE LA MEMÒRIA

1. OBJECTIU DEL PROJECTE	19
2. ABAST DEL PROJECTE	19
3. JUSTIFICACIÓ DEL PROJECTE	19
4. DESENVOLUPAMENT TEÒRIC.....	20
4.1. INTERFÍCIE GRÀFICA D'USUARI	20
4.1.1. Importància de les interfícies d'usuari.....	20
4.2. QUÈ ÉS UN ROBOT?.....	20
4.3. BREU HISTÒRIA DELS ROBOTS.....	21
4.4. TIPUS DE ROBOTS.....	23
4.5. BREU HISTÒRIA DELS ROBOTS PARAL·LELS.....	25
4.6. ROS.....	26
4.6.1. Perquè ROS.....	26
4.6.2. Conceptes de ROS	27
4.6.3. Funcionament general de ROS.....	28
4.7. LABVIEW	29
4.7.1. Perquè LabVIEW	29
4.7.2. Funcionament general de LabVIEW	29
4.8. COMUNICACIÓ DE LABVIEW AMB ROS.....	30
4.9. DESCRIPCIÓ DEL ROBOT	31
4.9.1. Descripció física del robot.....	32
4.9.2. Software del robot	32

4.9.3. Controlador emprat als experiments	33
4.9.4. Exercicis realitzables al robot	36
4.10. REHABILITACIÓ ROBÒTICA	37
5. DESENVOLUPAMENT PRÀCTIC	39
5.1. DESCRIPCIÓ I JUSTIFICACIÓ	39
5.1.1. Entrada d'usuaris.....	39
5.1.2. Pàgina principal de control.....	41
5.1.3. Recuperar sessions.....	43
5.1.4. Control de l'enginyer.....	45
5.2. MANUAL D'USUARI	46
5.2.1. Entrada d'usuaris.....	46
5.2.2. Pàgina principal de control.....	46
5.2.3. Recuperar sessions.....	48
5.2.4. Control de l'enginyer.....	49
5.3. REPRODUCCIÓ DE LA INTERFÍCIE	50
5.3.1. Entrada d'usuaris.....	50
5.3.2. Pàgina principal de control.....	53
5.3.3. Recuperar sessions.....	72
5.3.4. Control de l'enginyer.....	75
5.3.5. Retocs finals	78
5.4. WEZARP	80
6. CONCLUSIONS	83
BIBLIOGRAFIA.....	85
ANNEX 1	87
ANNEX 2	88

ÍNDIX DEL PRESSUPOST

1. DEFINICIÓ DE LES UNITATS QUE INTERVENEN.....	101
1.1. Mà d'obra.....	101
1.2. Material.....	101
2. QUADRES DE PREUS UNITARIS.....	102
2.1. Mà d'obra.....	102
2.2. Material.....	102
3. PRESSUPOSTS PARCIAIS.....	103
3.1. Material que s'empra en tot el projecte.....	103
3.2. Desenvolupament del projecte.....	104
3.3. Proves.....	104
3.4. Redacció.....	105
4. PRESSUPOST TOTAL D'EXECUCIÓ.....	105
5. PRESSUPOST TOTAL D'EXECUCIÓ PER CONTRACTA.....	106
6. PRESSUPOST BASE DE LICITACIÓ.....	106

ÍNDIX D'IMATGES

Il·lustració 1. Unimate.....	22
Il·lustració 2. Shakey.....	22
Il·lustració 3. Stanford Arm.....	22
Il·lustració 4. T3.....	22
Il·lustració 5. Puma.....	23
Il·lustració 6. Segons la transmissió A)rodes, B)erugues, C)potes.....	24
Il·lustració 7. Robot serial.....	24
Il·lustració 8. Robot paral·lel.....	25
Il·lustració 9. Vista superior del robot paral·lel de pintura de Pollard.....	25
Il·lustració 10. Vista lateral del robot paral·lel de pintura de Pollard.....	26
Il·lustració 11. Baxter.....	27
Il·lustració 12. Husky A200.....	27
Il·lustració 13. Nao.....	27
Il·lustració 14. Esquema del funcionament de ROS.....	28

Il·lustració 15. ROS topic init	30
Il·lustració 16. ROS topic close	30
Il·lustració 17. ROS topic read	30
Il·lustració 18. ROS topic write	30
Il·lustració 19. Float 32 building	31
Il·lustració 20. Float 32 parsing	31
Il·lustració 21. Bota per al peu i sensor de força del robot objecte d'estudi	31
Il·lustració 22. El robot objecte d'estudi	32
Il·lustració 23. Possibles moviments del peu	36
Il·lustració 24. Pestanyes del programa	39
Il·lustració 25. Pantalla "entrada d'usuari"	39
Il·lustració 26. Indicador de l'usuari actual	40
Il·lustració 27. Pantalla "pàgina principal de control"	41
Il·lustració 28. Parell límit de seguretat	41
Il·lustració 29. Referència de força.....	42
Il·lustració 30. Indicador de l'angle del peu	43
Il·lustració 31. Pantalla "recuperar sessions"	43
Il·lustració 32. Indicador del tipus d'exercici.....	44
Il·lustració 33. Gràfic polar	44
Il·lustració 34. Llegenda del gràfic dels parells.....	45
Il·lustració 35. Pantalla "control de l'enginyer"	45
Il·lustració 36. Accés d'usuari	46
Il·lustració 37. Zona per gravar exercicis.....	47
Il·lustració 38. Zona per executar exercicis	47
Il·lustració 39. Models 3D dels peus.....	47
Il·lustració 40. Carregar un exercici guardat	48
Il·lustració 41. Gràfic polar amb valors	49
Il·lustració 42. Control de la plataforma del robot.....	49
Il·lustració 43. Estructura principal del programa.....	50
Il·lustració 44. Registre d'usuari	51
Il·lustració 45. Afegir els usuaris als grups	51
Il·lustració 46. Event structure	51

Il·lustració 47. Codi per refrescar l'usuari actual	52
Il·lustració 48. Codi per accedir els usuaris	52
Il·lustració 49. Codi per tancar els usuaris	53
Il·lustració 50. VI lectura de força	56
Il·lustració 51. Codi per llegir la força.....	56
Il·lustració 52. Codi per passar la força als indicadors	57
Il·lustració 53. VI lectura de posició	57
Il·lustració 54. Topic de posició	57
Il·lustració 55. Codi de transformació dels valors a graus.....	58
Il·lustració 56. VI tancament dels VI paral·lels	58
Il·lustració 57. Codi complet tancament dels VI paral·lels	58
Il·lustració 58. VI indicador del final d'un procés.....	59
Il·lustració 59. Codi condicional per al "case structure"	59
Il·lustració 60. Codi per publicar el final del procés	60
Il·lustració 61. VI per actualitzar els gràfics de força i parells	60
Il·lustració 62. Codi complet gràfics força i parells	61
Il·lustració 63. VI per actualitzar el gràfic d'altura	61
Il·lustració 64. VI per mostrar els models 3D	62
Il·lustració 65. Codi per enviar valors als models 3D	62
Il·lustració 66. Codi per centrar els models 3D i orientar-los.....	63
Il·lustració 67. VI per gravar un exercici.....	63
Il·lustració 68. Codi per generar una matriu amb una fila de 0	64
Il·lustració 69. Codi per desactivar botons i habilitar els gràfics.....	64
Il·lustració 70. VI per esperar la posició estable del robot.....	64
Il·lustració 71. Codi per esperar la posició estable del robot.....	65
Il·lustració 72. Codi per guardar els valors a una matriu	65
Il·lustració 73. Codi per habilitar botons, reactivar botons i guardar dades.....	66
Il·lustració 74. VI per executar un exercici	66
Il·lustració 75. Codi per generar una matriu amb el tipus d'exercici a la primera fila	67
Il·lustració 76. Codi per desactivar botons.....	67
Il·lustració 77. Codi per obrir un arxiu i publicar el primer valor	67
Il·lustració 78. Codi per condicionar el bucle "while"	68

Il·lustració 79. Codi per publicar valors de l'arxiu	68
Il·lustració 80. Codi per guardar valors	69
Il·lustració 81. VI d'elecció del tipus d'exercici	69
Il·lustració 82. Codi per elegir el tipus d'exercici.....	70
Il·lustració 83. VI límit de seguretat	70
Il·lustració 84. Codi complet per publicar el límit	71
Il·lustració 85. VI referència de força	71
Il·lustració 86. Codi per al condicional del "case structure"	71
Il·lustració 87. Codi per calcular la referència i publicar-la	72
Il·lustració 88. Codi per obrir un arxiu i mostrar el tipus d'exercici	73
Il·lustració 89. Codi per modificar l'indicador de l'instant de temps	73
Il·lustració 90. Codi per mostrar els gràfics	74
Il·lustració 91. Codi per mostrar els gràfics polars	75
Il·lustració 92. Codi del primer requadre del control	75
Il·lustració 93. VI per calcular i publicar les posicions	76
Il·lustració 94. Codi per publicar la posició inicial i esperar a l'estabilitat del robot	76
Il·lustració 95. Codi per aplicar increments i publicar la nova posició	77
Il·lustració 96. Codi per actualitzar l'escala de l'eix d'abscises dels gràfics	78
Il·lustració 97. Codi per al botó "finalitzar VI paral·lels"	78
Il·lustració 98. Codi per inicialitzar valors	79
Il·lustració 99. Codi per inicialitzar l'escala d'abscises	79
Il·lustració 100. Codi per establir els colors dels indicadors d'angle.....	79
Il·lustració 101. Codi per mostrar un text	80
Il·lustració 102. Procés per carregar el WEZARP_SRV.....	80
Il·lustració 103. Procés per afegir el WEZARP_SRV.....	81
Il·lustració 104. Configurar clients.....	81
Il·lustració 105. Aplicació mòbil de Wezarp	82
Il·lustració 106. Esquema dels mòduls del controlador	87
Il·lustració 107. Codi complet lectura de força	88
Il·lustració 108. Codi complet lectura de posició	89
Il·lustració 109. Codi complet per indicar el final d'un procés.....	90
Il·lustració 110. Codi complet per refrescar el gràfic d'altura.....	91

Il·lustració 111. Codi complet models 3D (1)	92
Il·lustració 112. Codi complet models 3D (2)	93
Il·lustració 113. Codi complet per gravar exercicis	94
Il·lustració 114. Codi complet per executar exercicis (1)	95
Il·lustració 115. Codi complet per executar exercicis (2)	96
Il·lustració 116. Codi complet per executar els canvis pel tipus d'exercici	97
Il·lustració 117. Codi complet per generar la referència de força	98

ÍNDIX DE TAULES DE LA MEMÒRIA

Taula 1. Exemples de robots que funcionen amb ROS	27
Taula 2. Moviments del turmell	36
Taula 3. Forces màximes aplicables al turmell	37
Taula 4. Llistat de controladors, indicadors, etc, que s'han de generar	53

ÍNDIX DE TAULES DEL PRESSUPOST

Taula 1. Quadre descriptiu del material	101
Taula 2. Quadre de preus unitaris de la mà d'obra	102
Taula 3. Quadre de preus unitaris del material	102
Taula 4. Quadre de costos del material	103
Taula 5. Quadre de mà d'obra del desenvolupament	104
Taula 6. Quadre de mà d'obra de les proves	104
Taula 7. Quadre de mà d'obra de la redacció	105
Taula 8. Quadre del pressupost total d'execució	105
Taula 9. Quadre del pressupost total d'execució per contracta	106
Taula 10. Quadre del pressupost base de licitació	106

MEMÒRIA

1. OBJECTIU DEL PROJECTE

L'objectiu del projecte és el desenvolupament d'una interfície d'usuari funcional, amb el programa LabVIEW, per un robot paral·lel de tres graus de llibertat que funciona amb ROS. Aquest està destinat a la rehabilitació de turmell, per tant la interfície ha de permetre la utilització del robot per part dels professionals mèdics. A més a més, deu funcionar d'una manera suficientment intuïtiva per al metge, per tal d'evitar errors en els moviments gravats, ja que en una rehabilitació els moviments deuen ser precisos per no danyar més la zona a rehabilitar, i proporcionar avantatges respecte a la rehabilitació manual.

2. ABAST DEL PROJECTE

Aquest projecte abraça des d'esbrinar el funcionament de la llibreria que adapta LabVIEW a la forma de treballar de ROS, passant per adaptar la interfície a les limitacions de comunicació existents, i preparar aquesta per tal que un professional no vinculat al món de la robòtica pugui utilitzar-lo (sense suposar una gran dificultat ni limitar les possibilitats del robot), fins a aconseguir que les dades recaptades es representen de manera útil o inclòs permetre la utilització del programa a través de dispositius mòbils.

3. JUSTIFICACIÓ DEL PROJECTE

Dins de la indústria la robòtica avança, i la medicina intenta aplicar aquesta tecnologia al seu sector per tal de millorar els temps de rehabilitació, els resultats de les operacions o inclòs les condicions de treball dels seus professionals. Però aquests professionals mèdics no tenen l'obligació de conèixer el funcionament intern de tota la tecnologia que utilitzen, així que es fa necessari, hui per hui, el desenvolupament d'interfícies d'usuari.

Aquestes suposen una aproximació dels robots a sectors en els quals no es coneix el funcionament d'aquests, és a dir, permeten utilitzar les funcionalitats d'aquests d'una manera molt més intuïtiva, per exemple prement un botó o movent un "joystick".

4. DESENVOLUPAMENT TEÒRIC

4.1. INTERFÍCIE GRÀFICA D'USUARI

La interfície d'usuari es el medi a través del qual l'usuari pot comunicar-se amb una computadora o màquina. Són tots aquells menús, botons, controls i finestres, d'entre altres coses, que s'utilitzen per interactuar amb l'ordinador, és a dir, aquesta és el medi traductor de l'idioma de la persona al llenguatge màquina. L'objectiu és permetre el funcionament i control més efectiu de la màquina. També ha de contenir tots els canals de comunicació entre l'usuari i l'equip, i deu ser fàcil d'entendre i d'utilitzar.

Com es tracta d'una interfície gràfica, aquesta ha de ser manipulable mitjançant icones gràfiques i indicadors visuals, al contrari que les interfícies basades en text.

4.1.1. Importància de les interfícies d'usuari

Les interfícies són allò primer que es troben els usuaris quan van a utilitzar un programa informàtic o un robot, o siga, és la "cara" d'aquests amb l'usuari. Estan pensades per simplificar el funcionament de les màquines, permetent als usuaris no familiaritzats amb la programació la possibilitat d'utilitzar-les.

Al ser la "cara" i simplificar la seua utilització, esdevé de gran importància el fet de dissenyar-la correctament, ja que suposa la diferència entre un programa o màquina d'èxit i una que no ho és. Encara que un programa siga molt complet, si a l'hora d'utilitzar-lo és molt complicat i, a més a més, és difícil d'aprendre el seu funcionament, el públic potencial s'acabarà decidint per un altre que posseeixi una interfície més amigable, facilitant així la seua feina i accelerant-la. I per altra banda, remarcar la importància de què les connexions entre la interfície i el propi programa o robot estiguen ben efectuades, ja que de no ser així aquest no actuaria correctament segons el que se li està demanant.

Resumint, una interfície permet als usuaris la realització d'una tasca, per això deu estar dissenyada ajustant-se a la manera de treballar d'aquests, sent senzilla i intuïtiva a la vegada.

4.2. QUÈ ÉS UN ROBOT?

Segons l'Associació d'Indústries Robòtiques (sigles en anglès RIA, Robotic Industries Association) un robot industrial és *un manipulador multifuncional reprogramable, capaç de moure matèries, peces, ferramentes, o dispositius especials, segons trajectòries variables, programades per a realitzar tasques diverses.*

Segons l'Organització Internacional d'Estàndards (sigles en anglès ISO, International Organization for Standardization) un robot industrial és *un manipulador multifuncional*

reprogramable i controlador automàtic que pot ser programat en tres o més eixos. Aquest robot pot ser fixe o mòbil per a la seua utilització en aplicacions industrials d'automatització.

Però quin és l'origen d'aquesta paraula? La paraula robot va ser utilitzada per primera vegada en un llibre de literatura del 1920 anomenat "Rossum's Universal Robots" el qual va ser escrit pel txec Karel Čapek. Aquesta paraula significava servei, esclau, feina forçada i servia per anomenar les màquines treballadores.

4.3. BREU HISTÒRIA DELS ROBOTS

Cal remuntar-se a les primeres dinasties xineses i a la Grècia Antigua. Els dispositius que desenvolupaven eren màquines que realitzaven accions similars a les humanes o animals i que no estaven alimentades per energia elèctrica.

Més tard, allà l'Edat Mitjana, en Europa i el Mitjà Orient, es popularitzaren alguns autòmats, com a part del mecanisme dels rellotges, en els quals hi havia criatures mecàniques semblants a les persones.

Ja cap al segle XVIII amb la Revolució Industrial i el creixement de la importància de les matemàtiques i l'enginyeria, començà el desenvolupament de la robòtica com es coneix avui en dia, ja que a finals d'aquest i a principis del següent segle es van construir les bases de la informàtica.

Aquest avanç juntament amb la divisió del treball en tasques més xicotetes, el control per realimentació i les ferramentes més especialitzades foren el desencadenant de l'automatització de les fàbriques amb l'objectiu d'incrementar la productivitat i la precisió de les seues línies de producció. Tot açò es va estendre durant el segle XIX, preparant les que serien les grans evolucions en la robòtica, que van tindre lloc al segle XX.

El 1946 G. C. Devol va desenvolupar un dispositiu per registrar senyals elèctriques mitjançant camps magnètics i reproduir-les per accionar una màquina mecànica.

Durant els anys 50 es van desenvolupar manipuladors de control remot per part del laboratori ARGONNE, per tractar material radioactiu. A continuació, a mitjans de la dècada G. C. Devol realitzà dissenys per a la transferència programada de peces i, a finals d'aquesta, Planet Corporation va introduir el primer robot comercial, basat en el control per interruptors de final de carrera.

Començant els 60, s'introduí en el mercat l'Unimate (il·lustració 1), que estava basat en la transferència programada de Devol, el qual a finals de la dècada s'instal·là a una fàbrica de General Motors. A partir del 1965, es va començar a desenvolupar robots amb diferents característiques orientades a un propòsit, com per exemple un robot de pintura per polvorització de Trallfa o Shakey (il·lustració 2), un robot mòbil del Stanford Research Institute (SRI) amb diversos sensors com càmera de visió i sensors tàctils.



Il·lustració 1. Unimate

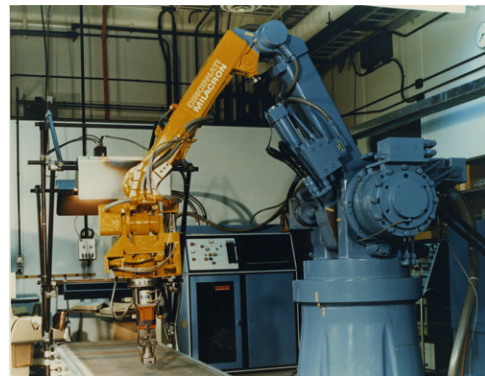


Il·lustració 2. Shakey

Passant a la primera part dels 70, la universitat de Stanford desenvolupà el Stanford Arm (il·lustració 3), un xicotet braç robòtic amb accionament elèctric. Seguidament, el 1973, el SRI va treballar en el primer llenguatge de programació de robots del tipus de computadora i un any després Cincinnati Milacron va introduir el T3 (il·lustració 4), un robot amb control per computadora.

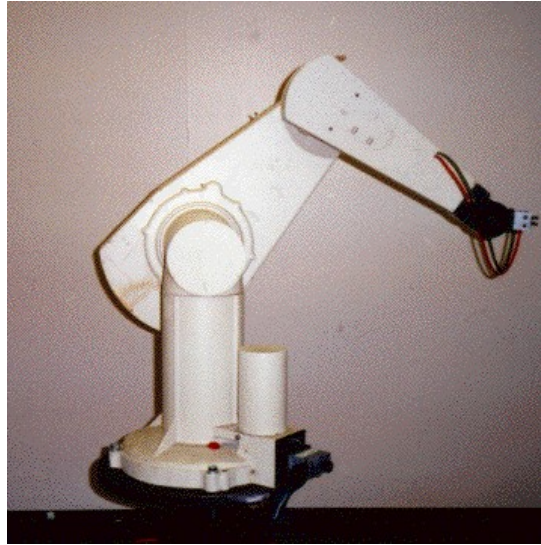


Il·lustració 3. Stanford Arm



Il·lustració 4. T3

En quant a la segona part d'aquesta dècada, és remarcable l'aplicació del microprocessador, que millorava considerablement les característiques del robot. Açò juntament amb la revitalització de l'automobilisme va dur a centrar els esforços en els robots de muntatge. El 1975 s'utilitzà el Sigma d'Olivetti amb eixe objectiu, seguit pel desenvolupament del PUMA (il·lustració 5) per part d'UNIMATION tres anys després i d'un robot tipus SCARA (sigles en anglès, Selective Compliance Arm for Robotic Assambly) per part de la universitat de Yamarashi (Japó).



Il·lustració 5. Puma

Els 80 suposaren una major flexibilitat en la utilització de robots, amb aquest objectiu Westinghouse Corp. va emetre un informe d'investigació sobre un Sistema de Muntatge Programable Adaptable (sigles en anglès APAS) que consistia en una línia de muntatge automatitzada i flexible. A part d'açò, el 1984 es desenvoluparen programes de robots utilitzant gràfics interactius en una computadora personal que després es carregaven al robot, la qual cosa facilitava la programació dels robots i per tant la seua utilització.

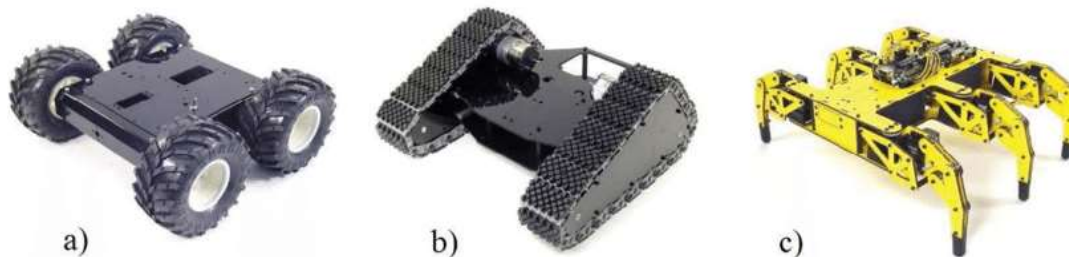
A partir d'aquesta dècada el fort impuls en la investigació per part de les empreses fabricants de robots i d'universitats de tot el món sobre informàtica aplicada i desenvolupament de sensors cada vegada més complexos, potenciaren la presència d'aquests tant en la indústria com en el dia a dia de la gent, facilitant les tasques a realitzar i obrint la possibilitat a noves feines que abans no es podien concebre. Aquests avantatges van fer créixer l'interès i van provocar que hi hagueren contínues millores en la seua aplicació fins al dia de avui, on és inconcebible la vida sense aquest tipus de màquines.

4.4. TIPUS DE ROBOTS

Hui per hui el món dels robots és molt extens, i hi ha una gran varietat en la funcionalitat que aquests poden exercir. Intentant ressaltar dues característiques, segons les quals es pugen classificar els robots, es troben la capacitat de mobilitat i la capacitat de manipulació.

- Robots mòbils: són màquines automàtiques capaces de moure's en un ambient determinat. Sorgeixen de la necessitat d'estendre el camp d'aplicació de la robòtica, restringit inicialment a l'abast d'una estructura mecànica ancorada. Disposen de mecanismes per a la transmissió de moviment, el que permet el seu desplaçament. Dins d'aquests, dues grans classificacions que es poden emprar són: segons el mecanisme de

transmissió que utilitzen (rodes, erugues, potes, hèlixs, etc) (il·lustració 6) i segons el medi en el qual es mouen (aeris, terrestres, marins, submarins, etc).



Il·lustració 6. Segons la transmissió A)rodes, B)erugues, C)potes

- **Robots manipuladors:** es caracteritzen per tindre posició fixa (ancorats) o moviment limitat a través de rails. Generalment disposen de tres o més eixos de posicionament per a l'execució de diferents tipus de treballs. Aquests solen utilitzar-se en entorns de treball totalment estructurats, és a dir, les condicions de contorn solen estar baix control, el que simplifica les tasques a realitzar. Dins d'aquests es poden classificar segons l'estructura mecànica que disposen.
 - **Serials:** presenten una configuració d'eslavons connectats en forma seqüencial, des de la base fins a l'eslavó final. Cadascun d'aquests està unit a l'anterior mitjançant articulacions on s'ubica un actuator. Els robots serials usualment tenen una forma antropomòrfica seguint el funcionament d'un braç humà i el principal avantatge que presenten és l'ampli espai de treball que poden abraçar en relació a les dimensions del propi robot. Algunes de les aplicacions a la indústria actual són muntatge, soldadura, fresat, tall per raig d'aigua, aplicació de pintura sobre superfícies, etc.



Il·lustració 7. Robot serial

- **Paral·lels:** mecanisme de cadena cinemàtica tancada en el qual una plataforma mòbil es troba unida a una base mitjançant diferents cadenes cinemàtiques independents. Aquesta configuració atorga als robots paral·lels certs avantatges en quant a rigidesa,

la qual cosa es tradueix en una major precisió, velocitat i acceleració degut a l'arquitectura del propi sistema, i en la relació càrrega-potència alta, ja que els actuadors del efector final estan connectats directament a la base. Algunes de les aplicacions més esteses en la indústria d'aquests tipus de robots són els simuladors de vol, transport d'objectes, cirurgia mèdica, posicionament d'antenes parabòliques, etc.

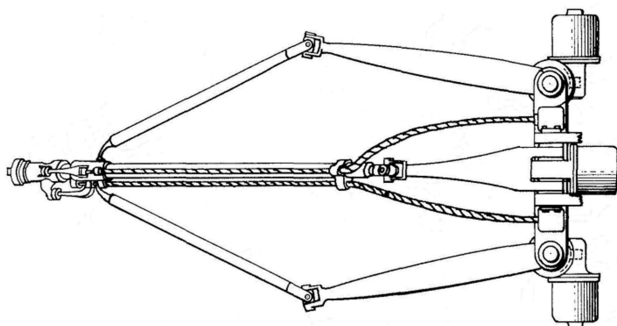


Il·lustració 8. Robot paral·lel

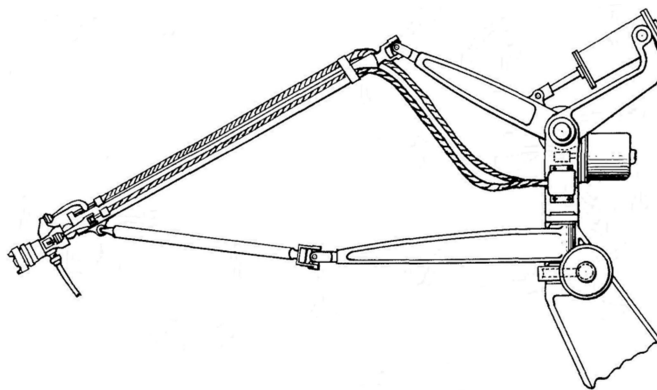
4.5. BREU HISTÒRIA DELS ROBOTS PARAL·LELS

Els treballs teòrics relacionats amb aquest tipus de robots es remunten a abans del concepte "robot". Encara que no va ser fins al segle XX que James E. Gwinnett pensà en construir una plataforma de moviment per a la indústria de l'entreteniment, i encara que se li va atorgar la patent en 1928, mai es va dur a terme la seua construcció.

Una dècada més tard, l'americà Willard LV Pollard dissenyà el primer robot paral·lel industrial. Aquest tenia cinc graus de llibertat i estava pensat per a operacions de pintura amb spray. Tenia tres braços amb dos esllavons i els actuadors es trobaven a la base (il·lustracions 9 i 10).



Il·lustració 9. Vista superior del robot paral·lel de pintura de Pollard



Il·lustració 10. Vista lateral del robot paral·lel de pintura de Pollard

No fou fins el 1947 quan Eric Gough dugué a terme el disseny d'una plataforma de tipus octaedre hexàpode amb costats de longitud variable, la finalitat de la qual era simular el procés d'aterratge d'un avió. Aquesta s'utilitzà posteriorment per a comprovació de pneumàtics baix càrregues aplicades en diferents eixos, és una de les màquines més reconegudes en la robòtica paral·lela.

Ja en els anys 80, el professor Raymond Clavel aparegué amb la idea d'utilitzar paral·lelograms per construir una estructura paral·lela robusta amb tres graus de llibertat de translació i un de rotació. El citat robot s'anomenà Robot Delta (una de les posteriors versions a la il·lustració 8), i a finals de segle es convertí en un dels més adequats dissenys per a robots paral·lels.

4.6. ROS

Robot Operating System (ROS) és un framework open source per a l'escriptura de software de sistemes robòtics. Es compon d'una col·lecció de ferramentes, llibreries i serveis amb l'objectiu de simplificar la creació d'aplicacions robòtiques complexes i robustes.

4.6.1. Perquè ROS




Crear aplicacions robòtiques complexes i robustes pot ser realment complicat, ja que hi ha problemes que semblen senzills o trivials per als humans, però són difícils de programar i d'implementar en un robot. Aquest framework s'ha creat amb l'objectiu de reunir els principals avantatges de les plataformes existents anteriorment i eliminar així part de la dificultat existent. A més a més, proporciona una arquitectura distribuïda, per tant no han d'estar els nodes en el mateix sistema.

Al ser un projecte col·laboratiu, s'ha facilitat l'intercanvi d'informació des del principi entre professionals de la robòtica, empreses participants i aficionats a aquest món, cosa que ha dut a la creació d'una comunitat mundialment estesa que treballa pel desenvolupament d'aquest projecte. Açò també assegura que siga una plataforma molt flexible i adaptable a les diferents necessitats.

Actualment ROS permet de forma oficial l'ús de certs llenguatges de programació com C++, Python i Lisp, encara que s'està treballant per ampliar aquest catàleg. ROS està orientat a ser utilitzat en un sistema basat en Unix, per tant sistemes operatius com Linux o Ubuntu estan oficialment coberts, i altres com macOS o Windows estan en fase experimental.

Per últim remarcar que aquesta plataforma, al ser ja una alternativa de pes i haver-se desenvolupat per gent amb diferents coneixements i robots molt variats, té suport natiu per una gran quantitat de hardware com pot ser:

Taula 1. Exemples de robots que funcionen amb ROS

Baxter de Rethink Robotics, Inc.	Husky A200 de Clearpath Robotics	Nao del laboratori de robots humanoides de la universitat de Freiburg
 <p>Il·lustració 11. Baxter</p>	 <p>Il·lustració 12. Husky A200</p>	 <p>Il·lustració 13. Nao</p>

4.6.2. Conceptes de ROS

- **Mestre:** conté el registre de noms dels nodes i consulta la resta de la xarxa. Sense aquest la resta de nodes no es trobarien entre ells i no podrien intercanviar missatges.
- **Nodes:** cadascun dels subsistemes que componen el sistema del robot. Per exemple, en una impressora 3D un node controlaria la temperatura a la que es fon el plàstic, un altre controlaria el moviment del motor en una dimensió, un altre el sensor de final de carrera, etc. Per tant els nodes poden subscriure's o publicar en els diferents topics que es generen, controlant així tot el sistema robòtic d'una manera flexible i ampliable. Cada node ha de tindre un nom propi i diferenciat per evitar problemes d'identificació.
- **Missatges:** els nodes es comuniquen mitjançant la utilització de missatges. Cadascun d'aquests és una estructura de dades que s'ha d'interpretar posteriorment.
- **Topics:** són canals de comunicació per transmetre missatges amb unes característiques comunes. En ROS, els topics poden ser transmesos sense una comunicació directa, és a dir, és un sistema desacoblat. Al publicar o subscriure's a un topic, ja es saben les

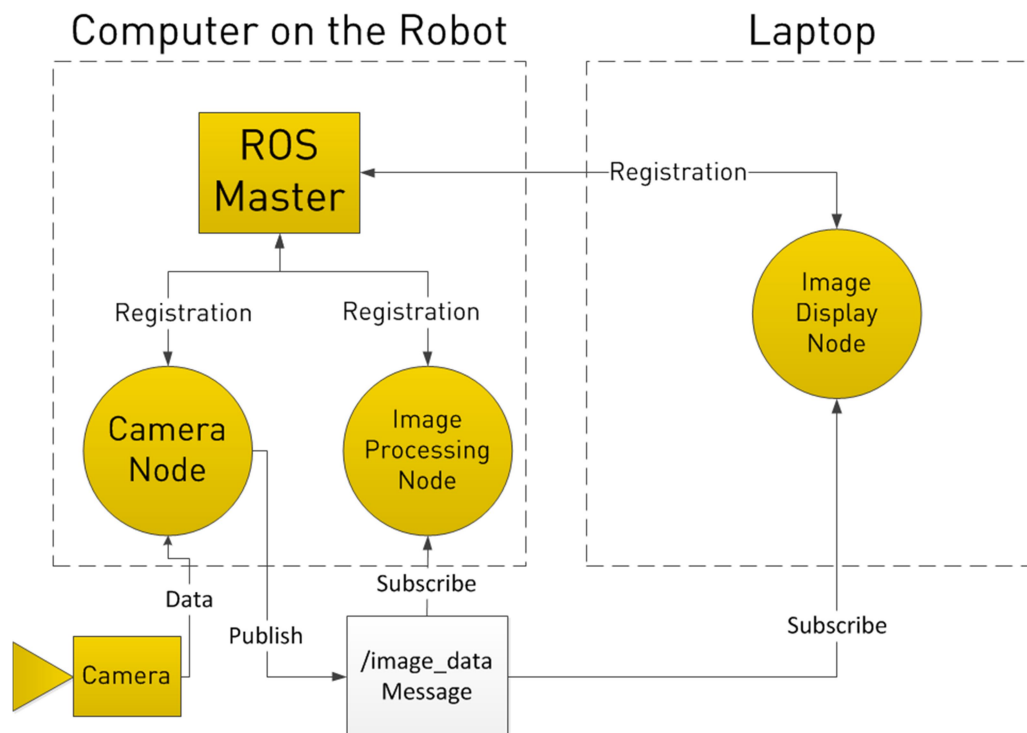
característiques del missatge que han d'enviar o que van a rebre. Poden existir molts publicadors per a un mateix topic i cada node pot publicar i subscriure's a molts topics. Les connexions entre diferents sistemes poden fer-se via TCP/IP o UDP.

- **Serveis:** el model de comunicació anteriorment plantejat és molt flexible, però en determinades situacions no és suficient. En els casos de transport de dades en un únic sentit, s'han creat unes interaccions petició i resposta que es realitza a través dels serveis. Un node proporciona un servei amb un nom i un altre node, que serà el client d'aquest servei, envia una petició per tal de rebre la resposta. D'aquesta manera queda configurada la connexió directa mitjançant serveis.
- **Bosses:** serveixen per guardar i tornar a llegir els missatges. Són l'historial dels missatges que s'han publicat durant tota l'execució.

4.6.3. Funcionament general de ROS

ROS es compon dels elements abans anomenats, dins d'aquests el mestre estaria jeràrquicament per dalt dels nodes generals. El mestre, com ja s'ha comentat abans, té el registre dels nodes, per tant, per a generar-los, s'han d'enregistrar. Primerament es decideix en quants nodes es vol dividir el sistema robòtic i es registren amb un nom únic en la xarxa. Una vegada registrats s'ha d'establir amb quins topics volen tindre relació i si és de tipus publicador o subscriptor, en ambdós casos s'ha de tindre clar quin tipus de missatges s'adeqüen al topic per tal de funcionar correctament.

Amb tot açò es poden establir els serveis que es creguen necessaris, i ja finalment fer funcionar el sistema tenint en compte la importància de que les connexions entre els diferents subsistemes que s'han de comunicar siguen les correctes.



Il·lustració 14. Esquema del funcionament de ROS

4.7. LABVIEW

Software d'enginyeria de National Instruments que utilitza llenguatge de programació "G". Aquest és un llenguatge de programació gràfic desenvolupat en la dècada dels 80 que utilitza un model de flux de dades i està pensat per al ràpid desenvolupament d'aplicacions d'adquisició de dades, automatització industrial i control d'instrumentació.

4.7.1. Perquè LabVIEW

Els principals avantatges de LabVIEW són, primerament, l'àmplia compatibilitat que presenta per a la connexió amb instrumentació industrial, ja que National Instruments segueix expandint les possibilitats d'aquest programa publicant noves llibreries per tal d'aconseguir-ho. Per altra banda, al ser un llenguatge de programació gràfic s'aproxima més a la forma de pensar de l'enginyer, fent així que siga més còmode i ràpid desenvolupar els programes requerits a la indústria, on el temps és crucial. A més a més, el programa recompila el codi amb cada acció que succeeix, el que ens duu a una ràpida detecció d'errors, eliminant així el temps de revisió del conjunt del codi. Per últim, remarcar que ja té incloses certes estructures recurrents al món de l'enginyeria com poden ser PIDs, gràfics, transformades de Fourier, etc.

Tot açò ha dut a aquest programa a ser una ferma alternativa per a la indústria actual, fent que cada vegada més empreses aposten per utilitzar-lo i més universitats per ensenyar-lo a les seues classes.

4.7.2. Funcionament general de LabVIEW

Primerament s'ha de generar un Instrument Virtual, què és com s'anomenen els programes, o un projecte, on dins s'agrupen els instruments virtuals relacionats. Una vegada generat, es disposen de dues pantalles, la del codi gràfic i la de l'usuari. Simplement s'ha d'anar arrastrant els diferents elements de la paleta de funcions en una de les dues pantalles (hi ha elements que poden anar en una o en l'altra i d'altres que si els arrastrem en una, es generen automàticament en l'altra) i anar interconnectant-los de manera gràfica en la pantalla del codi. Per saber com connectar-los de manera correcta es disposa d'un menú d'ajuda on s'explica quin tipus de dades van en les entrades i quin tipus es generen a les eixides, per evitar així haver de buscar-ho a un altre lloc.

Com el programa va compilant regularment el codi, quan s'intenta executar es sap immediatament si s'ha produït algun error en la programació o si està correcte, ja que si hi ha algun error no deixa executar-lo.

Una altra ajuda que es té és aquella que permet la visualització del flux d'informació en el moment en el qual el programa s'està executant, permetent així comprovar si s'ha plasmat el que es tenia en ment.

4.8. COMUNICACIÓ DE LABVIEW AMB ROS

Aquesta comunicació ha sigut possible a través de la llibreria “ROS for LabVIEW” disponible a la pàgina web <https://github.com/tuftsBaxter/ROS-for-LabVIEW-Software>. Aquesta va ser desenvolupada per la Tufts Univeristy i ha anat actualitzant-se amb els anys.

Els instruments virtuals que més s'han utilitzat durant aquest projecte són:

- ROS topic init: permet crear un node i establir el topic amb el qual es va a treballar (nom, tipus de dades d'aquest, freqüència de publicació...).



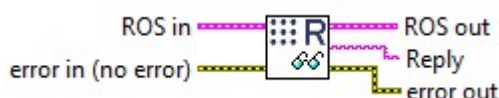
Il·lustració 15. ROS topic init

- ROS topic close: és molt important que una vegada es crea un node i s'ha acabat d'utilitzar, es tanque correctament.



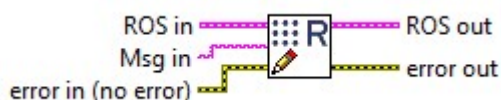
Il·lustració 16. ROS topic close

- ROS topic read: serveix per tal de llegir el topic indicat al “ROS topic init”. Posteriorment a aquest s'han d'acoblar els instruments virtuals de “message parsing” per tal de traduir el missatge.



Il·lustració 17. ROS topic read

- ROS topic write: es fa servir per escriure al topic indicat al “ROS topic init”. Abans d'aquest node s'han de traduir correctament els missatges amb “message Building”.



Il·lustració 18. ROS topic write

- Message building: té l'objectiu de traduir les dades que es volen passar del llenguatge de LabVIEW al llenguatge de programació de ROS.



Il·lustració 19. Float 32 building

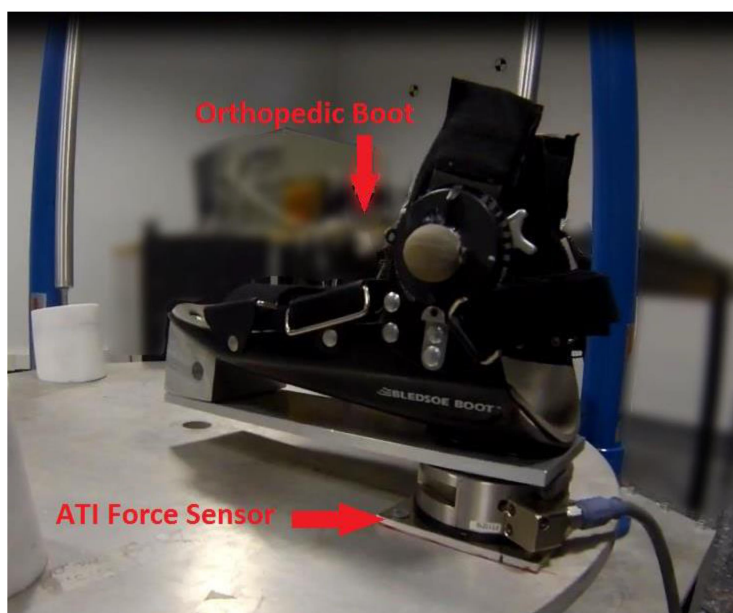
- Message parsing: fa la funció inversa al "message Building", o siga, tradueix els missatges del llenguatge de ROS al de LabVIEW.



Il·lustració 20. Float 32 parsing

4.9. DESCRIPCIÓ DEL ROBOT

Es tracta d'un robot paral·lel amb tres graus de llibertat (dos de rotació per tal de dur a terme els exercicis principals del turmell, que són dorsiflexió/plantarflexió i eversió/inversió, i un de translació en la direcció vertical). És un robot del tipus plataforma, el que permet realitzar exercicis en els quals el pes del pacient es suportat per aquest. El mecanisme es compon de tres cadenes cinemàtiques obertes amb la plataforma anteriorment anomenada i també presenta un sensor de força, el que permet avaluar com està interactuant el pacient amb la rehabilitació (il·lustració 21). Açò fa possible implementar activitats on la referència varia entre la posició i la força.



Il·lustració 21. Bota per al peu i sensor de força del robot objecte d'estudi

4.9.1. Descripció física del robot

El sistema el formen tres braços connectats a la plataforma mòbil mitjançant una esfera per braç. Cadascun d'aquests va unit amb una junta de revolució, en l'altre extrem, a una barra d'acoblament, la qual és perpendicular a la plataforma. Els braços en la part inferior, on estan les esferes, estan distribuïts formant un triangle equilàter sobre la plataforma (il·lustració 22).

Parlant de la part electrònica, els motors, disposats a la part baixa de les barres d'acoblament, són de corrent directa sense raspalls (en anglès "brushless DC") i estan equipats amb amplificadors de potència.

I per últim, es disposa d'una bota ortopèdica i d'un sensor de força sobre la plataforma de la màquina, ja comentats abans.



Il·lustració 22. El robot objecte d'estudi

4.9.2. Software del robot

L'arquitectura del software de la màquina es basa en el middleware Orocos (sigles en anglès, Open RObot COntrol Software). Per poder entendre-ho, un middleware és un software intermedi que assisteix a les aplicacions per interactuar o comunicar-se amb altres aplicacions o sistemes operatius. Dins dels tipus de middlewares, Orocos està orientat a components, i té dues característiques fonamentals: és configurable i reconfigurable. Aquesta segona es pot realitzar durant l'execució, el que aporta una gran flexibilitat per satisfer les necessitats de les aplicacions. A més a més, aquest està implementat íntegrament en C++ i permet la creació de mòduls que funcionen en temps real.

Degut a l'estructura i al disseny modular, encara que al principi pot ser complicat de programar, la feina del programador es veu reduïda ja que si un node funciona correctament en un esquema en particular, funcionarà en qualsevol altre sistema. Açò permet una reducció dels errors de programació i implementació de cadascuna de les parts del software.

Orocos és un bon marc de control de moviment disponible hui per hui, però presenta certes restriccions quan s'intenta arribar a algun objectiu més que el control en sí mateix.

Junt a l'anterior està ROS (sigles en anglès, Robot Operating System), que fou dissenyat per tal d'unir diferents ferramentes de programació de robots i obtindre així una solució més pròxima a la ideal. Els paquets d'aquest contenen llibreries, executables o scripts i cada paquet pot agrupar diferents nodes per tal de fer funcionar la xarxa.

Tenint en consideració els avantatges que cadascun ens aporta, al robot s'ha emprat ROS, ja que posseeix moltes ferramentes i funcionalitats útils en el desenvolupament d'aplicacions robòtiques, i Orocos, per què permet un control sòlid del sistema en temps real. Per tant s'ha utilitzat una combinació dels dos per aconseguir el correcte funcionament del mateix.

4.9.3. Controlador emprat als experiments

Per a dur a terme el control del robot paral·lel s'ha emprat un controlador per dinàmica inversa amb control per trajectòria que a més inclou el mòdul de força.

El model dinàmic emprat relaciona la localització del robot (definida per les variables de les articulacions), la força de parell aplicada en les articulacions i els paràmetres dimensionals del robot (com poden ser la longitud, massa i inèrcia des elements). Endemés és primordial per a la simulació del robot i per al disseny i avaluació de l'estructura mecànica del robot i del control dinàmic del robot.

A remarcar que en el projecte de desenvolupament del robot, encara que s'han utilitzat diversos paquets d'Orocos, un dels que més ha intervingut ha sigut el "Real time toolkit" (RTT) ja que permet, entre altres coses, capturar i graficar el flux de dades, reassignar prioritats i modificar l'algoritme de control en execució i interactuar amb altres dispositius directament amb una interfície gràfica d'usuari. I la "Orocos component library" (OCL) per tal de crear els components en Orocos, modificar-los i configurar-los segons les preferències dels usuaris.

La implementació d'aquest controlador està feta de manera modular amb Orocos. Els mòduls emprats per a implementar el controlador han estat:

- moduloPcl833: aquest s'encarrega de marcar el temps per al període de mostreig, donat que quan ell s'executa, provoca que també ho facen la resta. A més, s'encarrega de llegir els valors dels encoders del robot real. Aquest valor després es transforma en metres i es passa als mòduls que corresponen.
- moduloGenRefMod: mòdul encarregat de generar la referència de posició per a cadascuna de les iteracions. La referència d'aquesta es divideix en tres etapes: Spline inicial – Moviment central – Spline final. A la primera etapa, es du al robot des d'una posició de repòs fins al primer valor de referència del moviment central, mitjançant una trajectòria en

forma de spline amb 9 segons de temps. De la mateixa forma, la tercera etapa s'encarrega de fer el contrari, és a dir, dur el robot des de l'últim valor de referència fins a una posició segura de repòs. Al moviment central el que es feia, abans del canvis introduïts amb aquest projecte, era obtenir els valors de Gamma, Beta, Zeta de referència per a cada iteració, els quals es llegien d'un fitxer. I després feia la cinemàtica directa per tal d'obtenir els valors de referència de posició per a les articulacions del robot. A més, envia una senyal al control de força, si s'empra aquest.

- actuador: s'encarrega d'enviar un valor de tensió als motors. S'instancia tres vegades per a cadascuna de les articulacions.
- cine3dof: fa el càlcul de la cinemàtica.
- para_Gv: calcula els termes gravitacionals.
- pd2: implementa un controlador PD.
- sumInerGvC: s'encarrega de sumar el que li vinga per la entrada i retornar-ho per la seua eixida.
- derivada: calcula la derivada d'un vector de tres components. S'ha instanciat quatre vegades com a "deriv_real" (derivada de la posició real), "deriv_des" (derivada de la posició desitjada o referència), "acel" (derivada de la velocitat desitjada), "deriv_GBZ" (derivada de la referència en la forma Gamma, Beta, Zeta)
- moduloFuerza: llegeix els valors provinents del sensor de força i els envia a altres mòduls en forma d'un vector de sis components on els tres primers corresponen a la força (en Newtons) i els altres tres als moments (en Newtons*metre).
- moduloGenerFuerzaGBZ: genera la referència de força a aplicar quan s'empra control de força.
- moduloJacobiano: s'encarrega d'implementar el Jacobià de força.
- moduloSumaRefes: suma la referència de posició amb la integral del Jacobià.
- cine3dofdirecta: calcula els valors de Gamma, Beta i Zeta corresponents a la posició de les tres articulacions del robot.

Cadascun d'aquests mòduls es comunica amb l'altre o altres mitjançant ports d'entrada o d'eixida. Aquests es defineixen mitjançant el codi per a implementar el mòdul concret, i la connexió entre ells es crea mitjançant el deployer que s'encarrega de crear les aplicacions (per a crear les aplicacions es fa mitjançant un script .ops). L'esquema de les connexions es troba disponible a l'annex 1.

A les últimes versions de ROS, aquest incorpora el metapaquet "rtt_ros_integration" que permet crear i emprar components d'Orocos dins de ROS. Aquest incorpora a la seua vegada el paquet "rtt_roscomm" el qual permet publicar i subscriure's a topics de ROS.

Per tal de poder fer açò últim, cal que el mòdul d'Orocos que vol publicar o subscriure's a un topic de ROS cree un port, o bé d'entrada o d'eixida (depenent de que vulga subscriure's o publicar del topic), on el tipus de dades que va a rebre o emetre siga dels tipus inclosos a "std_msgs", és a dir, els missatges de ROS estàndard. A més d'açò, quan es creen les aplicacions amb el deployer, cal transmetre les dades des del port d'Orocos, el qual publica o rep missatges estàndard, a un topic de ROS. Al fitxer .ops corresponent a l'aplicació s'ha d'afegir:

```
stream("ComponentOrocos.NomPort", ros.topic("/nom_topic"))
```

Per a desenvolupar l'aplicació que es planteja a aquest projecte, s'han hagut de fer les següents modificacions als mòduls esmentats abans:

- Al mòdul "moduloFuerza" s'ha afegit un port que publica missatges estàndard de ROS, concretament els sis valors que proporciona el sensor de força, per a poder després, del fitxer .ops, transmetre'ls a un topic de ROS al qual es subscriu la interfície.
- Per a poder gravar els exercicis des de LabVIEW, el controlador ha de passar-li a través d'un topic les distintes posicions que al llarg de l'execució va prenent el robot. Habitualment, per al control del robot s'empren les posicions de cadascuna de les articulacions, però a l'hora de fer que el robot segueixi una determinada trajectòria, aquesta es dona com una sèrie de punts amb els angles Gamma, Beta i l'altura Zeta. Així, com que es vol que l'exercici gravat es pugui reproduir posteriorment quan el metge ho sol·licite, el port d'eixida del mòdul "moduloPcl833" corresponent a les posicions reals de les articulacions del robot es passen pel port d'entrada al mòdul "cine3dofdirecta", el qual, com s'ha dit abans calcularà els valors de Gamma, Beta i Zeta corresponents a les posicions de les articulacions. A aquest mòdul s'ha fet el mateix que el que s'ha esmentat per al cas del "moduloFuerza", però en aquest cas per tal de publicar els valors corresponents a un topic de ROS de manera que l'aplicació de LabVIEW es pugui subscriure.
- Per al cas de voler reproduir algun exercici prèviament gravat, cal que la interfície publiqui en un topic de ROS les posicions Gamma, Beta i Zeta per a que l'aplicació de control del robot agafi aquests valors com a referència. Per a aconseguir açò, s'ha hagut de modificar el mòdul "moduloGenRefMod" donat que era aquest l'encarregat de generar els valors de referència. Així, en lloc d'agafar els valors de referència d'un fitxer ha d'agafar-los del que li arriba pel topic que publica l'aplicació de LabVIEW. El que passa és que els valors aquests únicament corresponen a la part del moviment central que s'ha explicat abans, per tant, la interfície ha d'esperar a enviar els valors fins que el robot haja fet la primera etapa de spline. Per a aconseguir açò, s'ha afegit un port a aquest mòdul que envia un valor binari indicant si ha finalitzat o no la primera etapa de moviment. Les dades d'aquest port es retransmeten al corresponent topic com abans. Una vegada finalitza el primer moviment, el mòdul empra com a valors Gama, Beta i Zeta de referència els valors rebuts per un port nou d'entrada del mòdul al qual li retransmet les dades el topic corresponent de l'aplicació de LabVIEW. A més, s'ha afegit un port nou també d'entrada per a que la interfície informi al controlador, i concretament a aquest mòdul, de que finalitza el segon moviment i que per tant, ja no li van a arribar més valors de referència i ha de fer la spline final per a anar a la posició de repòs.

4.9.4. Exercicis realitzables al robot

Els moviments realitzables pel turmell són els indicats en la il·lustració 23, encara que el robot del qual es disposa en el laboratori està pensat per realitzar els exercicis determinats pels eixos x i y de la il·lustració nomenada, és a dir, plantarflexió i dorsiflexió per part de l'eix y i inversió i eversió per part del x.



Il·lustració 23. Possibles moviments del peu

Una part importat d'aquest projecte és saber quins són els límits de moviment del turmell per tal d'adequar correctament la interfície a realitzar. En la taula 2 es pot veure quins són els valors que s'aplicaran en el disseny del programa informàtic, ja que, recolzant-se en diferents estudis, els dissenyadors del robot decidiren establir aquests límits. Remarcar que ja que no totes les persones arriben a aquests límits d'actuació, l'exercici a realitzar s'ha d'adequar al pacient a tractar per part del professional que utilitza el robot.

Taula 2. Moviments del turmell

Tipus de moviment		Moviment màxim del turmell	Moviment màxim del robot
Eix Y	Dorsiflexió	20.3° - 29.8°	50°
	Plantarflexió	37.6° - 45.8°	50°
Eix X	Eversió	15.4° - 25.9°	50°
	Inversió	22° - 36°	50°

A continuació, en la taula 3, es poden veure els límits en quant a la magnitud de par que es deuria aplicar a un turmell humà i les que pot arribar a enregistrar el sensor del robot a tractar.

Taula 3. Forces màximes aplicables al turmell

Tipus de moviment		Màxim par passiu aplicable a un turmell	Límits del sensor
Eix Y	Dorsiflexió	- 33.1±- 16.5 N·m	± 270 N·m
	Plantarflexió	40.1 ± 9.2 N·m	± 270 N·m
Eix X	Eversió	- 48.1 ± 12.2 N·m	± 270 N·m
	Inversió	34.1 ± 14.5 N·m	± 270 N·m

Centrant-nos en els exercicis que es poden aplicar en aquest sistema, amb l'objectiu de rehabilitar turmells lesionats, trobem els següents tipus d'exercicis:

- Exercicis passius: en aquests tipus d'exercicis es programa al robot per seguir un camí indicat per l'especialista, en base a uns sensors de posició (amb un error de posició al voltant de 0.5 mm) dels quals disposa el robot a tractar.
- Exercicis actiu-resistius: són exercicis actius en els quals el pacient ha de superar una resistència aplicada pel robot. Centrant la descripció en el robot que ens ocupa, la plataforma descriu un moviment i el pacient ha de mantenir-la tot el que puga en posició horitzontal, aplicant parells oposats als del robot.
- Exercicis actiu-assistits: estan pensats per ser exercicis actius com els anteriors, però amb la diferència que en aquests el pacient no deu ser capaç d'anar en contra del moviment. Solen requerir d'ajuda externa per a realitzar-se correctament, ja que es solen realitzar en les primeres fases de la rehabilitació. Aquest tipus d'exercici no serà suportat en la interfície generada.

4.10. REHABILITACIÓ ROBÒTICA

Actualment l'enginyeria robòtica està participant activament en el camp de la medicina per tal de millorar i facilitar la tasca dels professionals mèdics, ja siguin metges, infermers, fisioterapeutes, dentistes, etc. Açò ens duu al punt de plantejar-nos, en la situació actual, quins són els avantatges i inconvenients de la rehabilitació mitjançant robots front a la rehabilitació manual.

Primerament es van a remarcar els avantatges que suposa la rehabilitació robòtica. Entre aquests és important l'aspecte que un robot permet generar repeticions més exactes dels moviments i permet rebre informació quantitativa sobre com s'està duent a terme la rehabilitació, açò permet acotar més fermament l'exercici i un posterior tractament de les dades per part del professional mèdic, per tal de veure si ha de variar o continuar amb la teràpia marcada. Com a conseqüència de què s'està monitoritzant en tot moment els exercicis dels pacients, passat un temps, es pot disposar d'una base de dades de rehabilitacions

realitzades i poder comparar així la de diferents persones per a veure si tenen una progressió semblant, podent preveure quin serà el següent pas en la recuperació del pacient actual. Per últim es podria puntualitzar que el robot no es cansa rehabilitant persones, la qual cosa ens duu a què el professional tinga la possibilitat de rehabilitar a un nombre superior de pacients sense desgast físic.

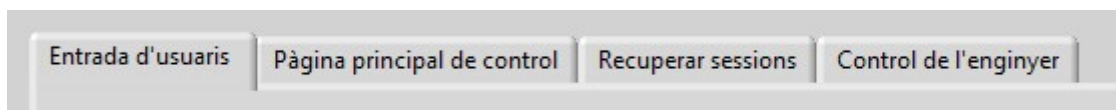
En la part dels desavantatges podríem trobar l'alt cost del producte de rehabilitació, en comparació amb els productes que es solen utilitzar en la rehabilitació convencional, i la negativa d'alguns pacients a rebre certs tipus de tractaments, és a dir, el component psicològic del pacient amb una mala experiència anterior amb altres robots de rehabilitació. Encara que també es podria remarcar que al realitzar una màquina l'exercici, i estar acotat el possible moviment açò produeix que l'exercici repetitiu siga sempre igual, la qual cosa no es totalment adaptable a les condicions usuals de moviment d'eixa part del cos.

5. DESENVOLUPAMENT PRÀCTIC

L'explicació de la interfície d'usuari que s'ha decidit realitzar amb l'objectiu de cobrir les necessitats de control, manipulació del robot i obtenció de dades dels diferents usuaris que puguen interactuar amb aquest, es va a dividir en tres parts, una que consistirà en la descripció de la interfície i la justificació de cadascun dels elements que s'han utilitzat, una segona que servirà de manual d'usuari, per tal d'explicar la correcta utilització del programa i les possibilitats de què disposa, i l'última per tal d'explicar el codi creat per al funcionament de la interfície.

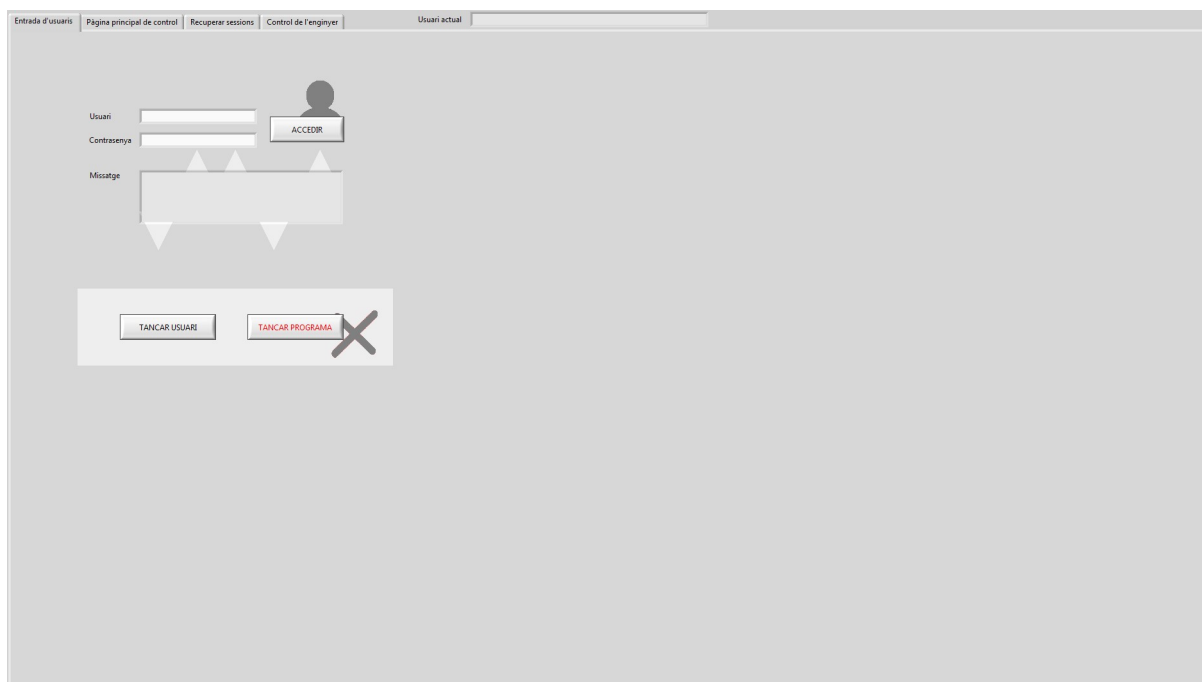
5.1. DESCRIPCIÓ I JUSTIFICACIÓ

Primerament remarcar que es va a separar en tants apartats com pantalles es disposen (il·lustració 24), per tal d'establir un ordre en la descripció i facilitar la comprensió del programa.



Il·lustració 24. Pestanyes del programa

5.1.1. Entrada d'usuaris



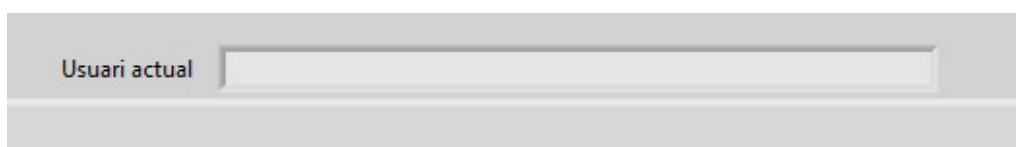
Il·lustració 25. Pantalla "entrada d'usuaris"

En aquesta pantalla, a la part superior, es disposa de dos requadres on poder escriure l'usuari i la contrasenya, els quals hauran sigut afegits al registre per part de l'administrador. Per predeterminat s'han creat dos usuaris, un que serà per al metge i un altre per a l'enginyer, encara que s'haurien de crear un usuari per a cada persona que haja d'utilitzar el programa. El primer tindrà accés a totes les pantalles excepte a la de "Control de l'enginyer", i el segon a totes. A més a més els enginyers tenen propietats d'administrador, el que els permet crear nous usuaris i eliminar els ja existents.

El registre d'usuaris s'ha creat per tal d'evitar que una persona que no conega el correcte funcionament del robot pugui fer un ús irresponsable d'aquest. I s'ha dotat de diferents característiques als usuaris perquè s'entén que durant les possibles revisions, l'enginyer ha de poder moure el robot a qualsevol posició de manera remota, a diferència del metge que ha d'executar un exercici guardat o guardar-ne un.

A mà dreta dels requadres abans esmentats, es disposa d'un botó que permet la introducció de l'usuari i la contrasenya indicada per tal de comprovar si són correctes ambdós camps o si hi ha algun error. Aquest possible error s'indicaria en el caixetí anomenat missatge que es troba un poc més cap a baix, al igual que indicaria si s'ha accedit correctament a l'usuari indicat. Aquest caixetí és merament informatiu, i no permet l'escriptura sobre ell.

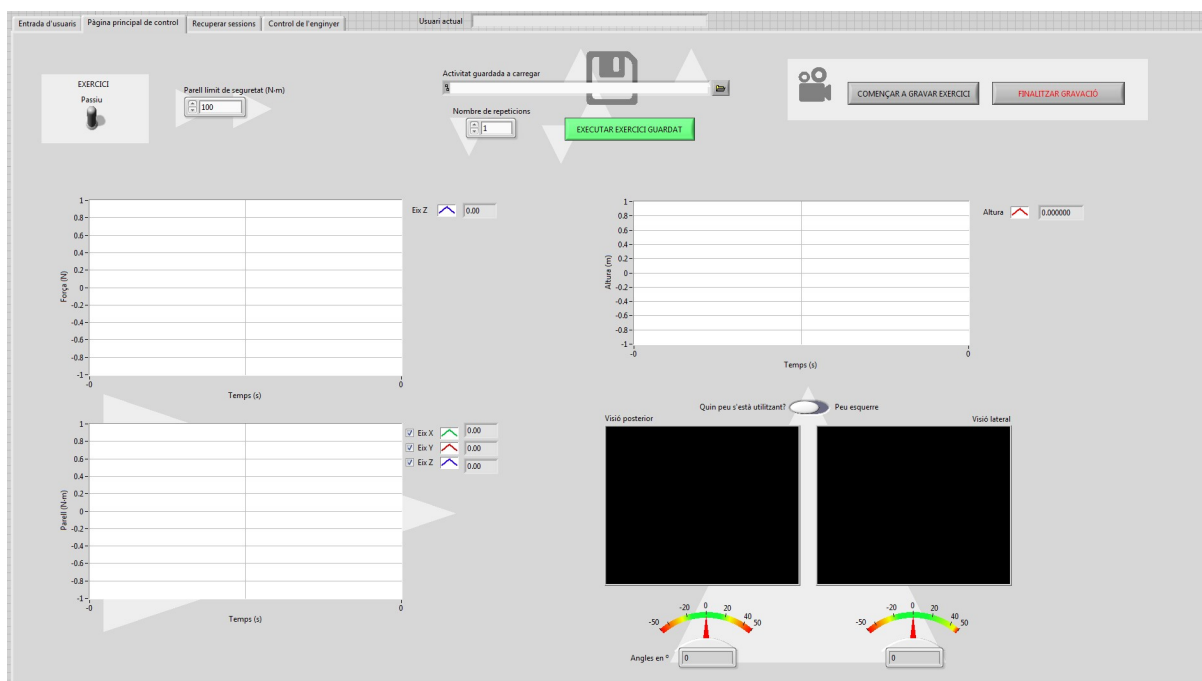
Encara que no es troba en la mateixa pantalla com a tal, sinó que serà visible a totes les pantalles, a mà dreta de les pestanyes d'elecció podem veure un altre requadre en el qual s'indicarà, durant tota l'execució del programa quin usuari està connectat (il·lustració 26). Aquest s'ha ubicat d'aqueixa manera per tal que es pugui saber d'una sola mirada quin usuari està enregistrat, i per tant, quines característiques del programa té habilitades o no, evitant així que s'haja de canviar de pantalla per saber-ho.



Il·lustració 26. Indicador de l'usuari actual

Per últim, en la part mitja de la pantalla, en un requadre de color més clar, es poden trobar dos botons. El de l'esquerra té l'objectiu de permetre el tancament de l'usuari actual sense haver de parar el funcionament del programa, ja que així es facilita l'intercanvi entre els professionals que requereixen el robot. I el de la dreta l'objectiu de tancar el programa, per tal d'evitar que s'haja de tancar de manera forçosa com a únic recurs.

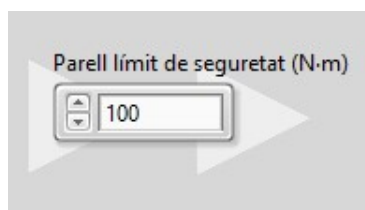
5.1.2. Pàgina principal de control



Il·lustració 27. Pantalla “pàgina principal de control”

Per a la descripció d'aquesta es començarà per la part superior. A l'esquerra del tot es troba un interruptor que permet l'elecció del tipus d'exercici que es va a realitzar, aquest activa o desactiva la referència de força característica dels exercicis resistents i també serveix per saber quin tipus d'exercici es tracta en la posterior recuperació de les sessions.

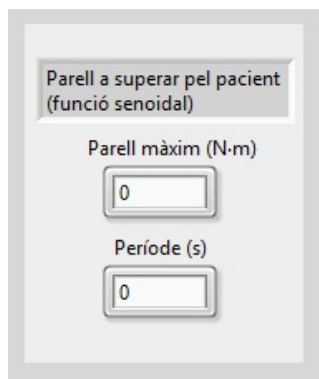
A continuació d'aquest, anant cap a la dreta, es pot veure un controlador del par límit de seguretat (il·lustració 28), si aquest és sobrepasat durant un exercici el robot pararia d'executar-lo. Remarcar que aquest és modificable de manera dinàmica durant l'execució dels diferents exercicis, per tal de permetre-li al metge una major flexibilitat en el control de la seguretat del pacient.



Il·lustració 28. Parell límit de seguretat

A la dreta del par límit de seguretat hi ha uns elements que són visibles, com s'ha comentat anteriorment, solament si es troba en un exercici resistiu. Es tracta de la referència de força (il·lustració 29), la qual s'ha decidit que siga una funció sinusoidal a la qual s'ha d'aportar l'amplitud i el període. És sinusoidal per tal que varie en el temps i no haja de fer el

pacient contínuament la mateixa força, sinó que faça treballar el turmell d'una forma més exigent, més semblant al món real.



Il·lustració 29. Referència de força

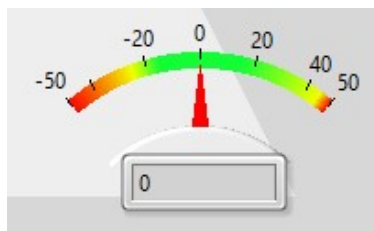
Encara en la part superior, però ja en el centre de la pantalla hi ha una zona per tal de carregar activitats guardades i executar-les. Aquesta compta amb un controlador per introduir la ubicació de l'arxiu i una icona per tal de buscar-lo, un controlador numèric per tal d'introduir el nombre de vegades que es vulga repetir l'exercici, així alleujar la càrrega del metge quan ha de gravar els exercicis, i per últim un botó per tal d'executar l'exercici, el qual, quan està executant-se, serveix per tal de tallar-lo a la meitat si es requerira.

Passant a la dreta del tot, s'hi pot trobar, en un requadre més clar, dos botons que es fan servir per començar i finalitzar la gravació d'un exercici per part del metge, per després fer-lo utilitzar amb el pacient. S'ha pensat aquesta manera de procedir amb la rehabilitació perquè els professionals de la medicina no solen estar acostumats a treballar amb robots, solen fer les rehabilitacions manualment, i aquesta suposa la forma més propera a aquest concepte. D'aquesta manera simplement han de fer la rehabilitació que creuen pertinent una vegada, gravar-la, i posteriorment reproduir-la, la qual cosa facilita la utilització del robot i la comprensió del seu funcionament per part d'aquests.

Puntualitzar què tots els botons d'aquesta pantalla són tots del tipus de comportament "switch" per permetre la seua utilització en els instruments virtuals que s'explicaran en la part de programació. Açò és manipulable a "clic dret sobre el botó/Properties/Operation/Button behaviour" i és estrictament necessari per poder utilitzar referències d'aquests.

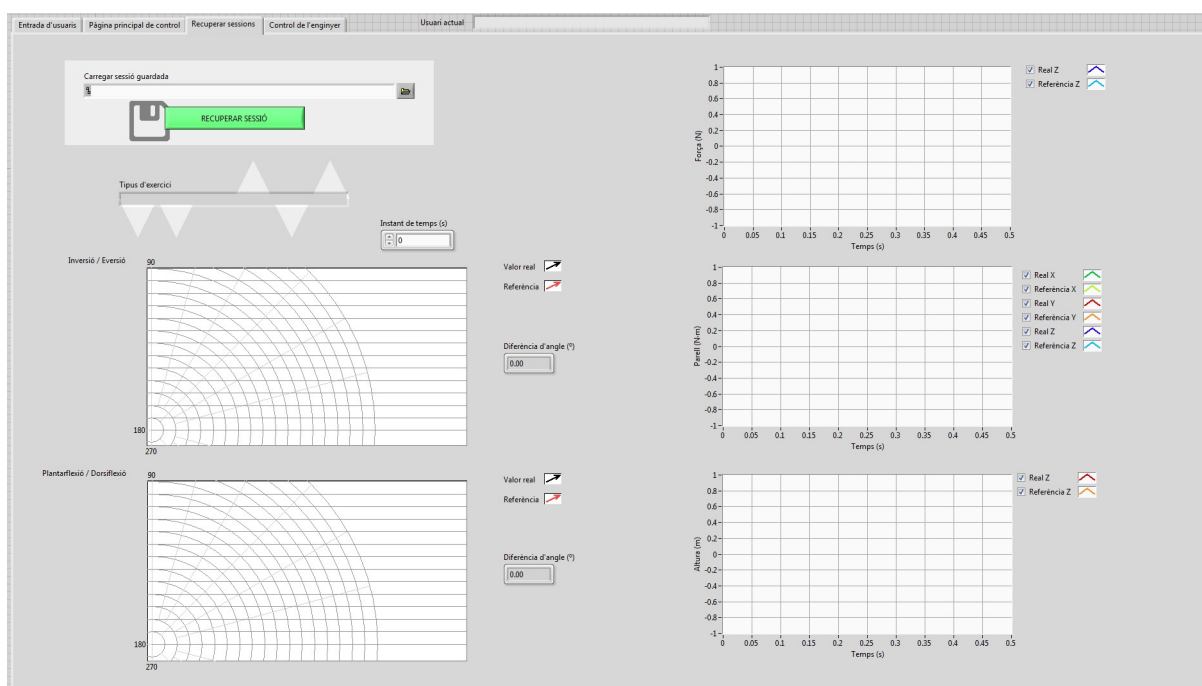
La major part de la pantalla l'ocupen els gràfics, ja siguen els gràfics de força o de pars que trobem a l'esquerra, o el gràfic d'angles o d'altura a la dreta. Els dos primers gràfics i el d'altura són el mateix tipus de gràfics, als quals a part de la llegenda s'ha afegit un indicador que mostra l'últim valor, ja que en qüestions com la força és molt important saber el valor que s'està aplicant en el moment. El gràfic d'angles consisteix de dos models 3D (peu dret i peu esquerre) per tal de mostrar la inclinació del peu, i com és una manera prou intuïtiva de veure, però no es sap fàcilment el valor exacte de l'angle, s'ha disposat un indicador numèric a la part baixa d'aquests. A part d'aquests dos elements, també es disposa d'un indicador d'agulla que està graduat per colors segons és una zona d'inclinació del peu segura o no, açò depèn del

pacient que estiga en la rehabilitació, però s'ha dispostat per tal que siga un indicador d'ajuda (il·lustració 30).



Il·lustració 30. Indicador de l'angle del peu

5.1.3. Recuperar sessions



Il·lustració 31. Pantalla "recuperar sessions"

A la part superior de la pestanya es disposa, en una àrea de color més clara, d'un requadre on poder introduir la ubicació d'un fitxer generat prèviament en una sessió de rehabilitació o en la gravació d'un exercici. Aquesta ubicació es pot introduir manualment o buscant l'arxiu a l'ordinador, polsant el botó amb la icona de la carpeta que es troba a la dreta del requadre. Més cap a baix es troba el botó de recuperar sessió per tal de carregar l'arxiu indicat i poder visualitzar la informació.

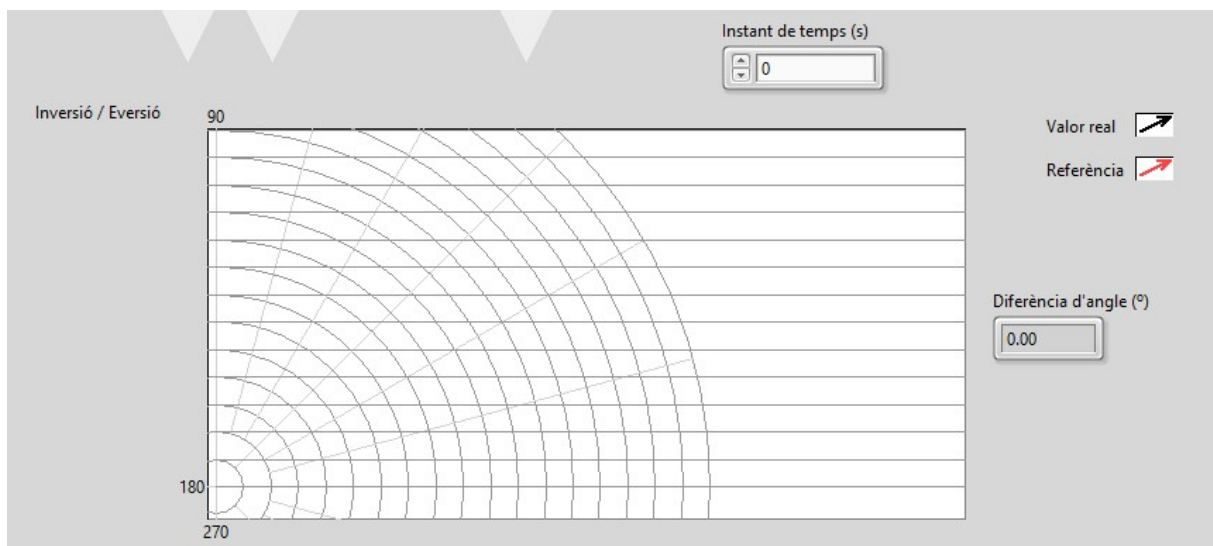
Ja fora d'aquesta zona clara, podem veure un indicador anomenat tipus d'exercici (il·lustració 32) que permetrà visualitzar si es tracta d'un exercici gravat, resistiu o passiu. Açò s'ha dispostat per tal de reduir la possibilitat d'error del metge, ja que al carregar els arxius podria pensar que acaba de carregar un exercici passiu d'un pacient, sent realment un exercici resistiu, i interpretar de manera errònia les dades de les quals disposa.



Il·lustració 32. Indicador del tipus d'exercici

A la zona mitjana i baixa de la part esquerra es disposen dos gràfics polars (il·lustració 33). Aquests permeten veure en cada instant de les sessions guardades quina inclinació tenia el peu del pacient, i quina era la referència que s'havia aplicat. L'instant que es vol visualitzar es pot elegir mitjançant un controlador situat dalt dels gràfics polars.

A part d'aquest controlador, també hi ha dos indicadors, un per a cada gràfic, que permeten veure la diferència d'angle que s'ha produït entre la referència i l'angle real, per tal d'intentar-ho corregir en futures sessions.



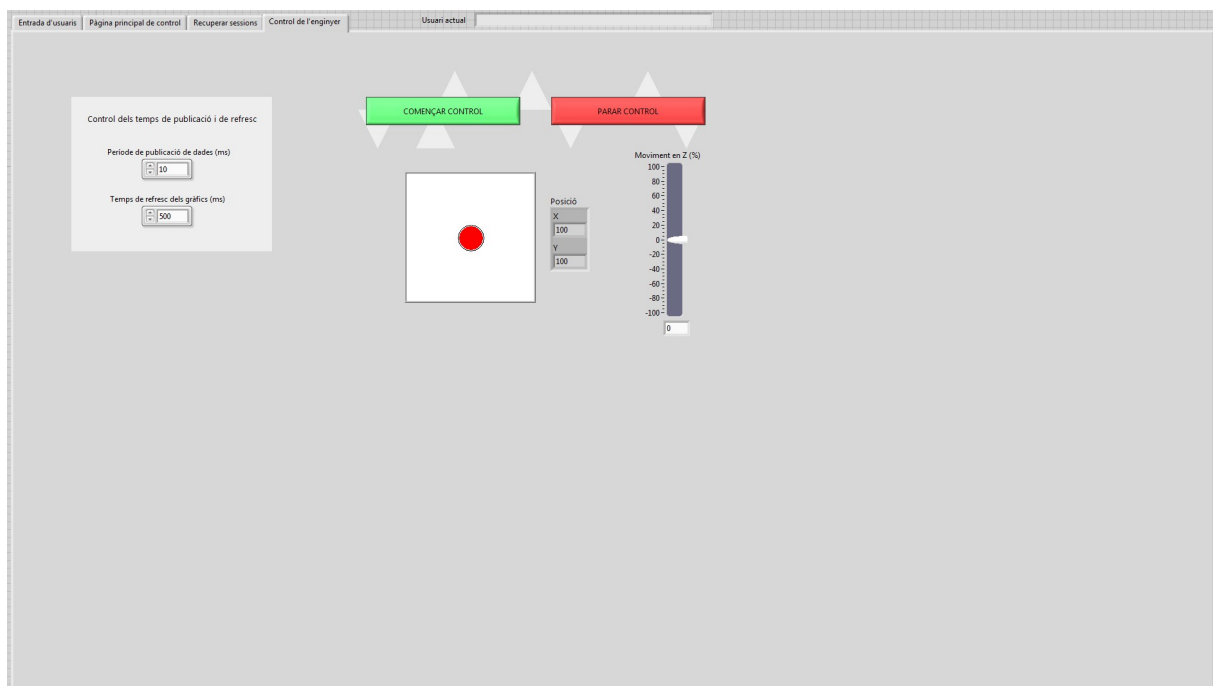
Il·lustració 33. Gràfic polar

Ja a la part dreta de la pantalla es troben tres gràfics: un de força en l'eix Z, un de parells i un de l'altura del peu del pacient. Els tres estan pensats per a què utilitzen la mateixa escala de temps, disposant així de totes les dades d'un mateix instant d'una ullada. També s'han disposat de caselles per a cadascuna de les funcions per tal de mostrar-les o amagar-les (il·lustració 34), amb l'objectiu d'utilitzar el mateix gràfic per a tots el parells i les seues referències (6 funcions) i que no resulte incòmode d'interpretar per part del professional.



Il·lustració 34. Llegenda del gràfic dels parells

5.1.4. Control de l'enginyer



Il·lustració 35. Pantalla "control de l'enginyer"

Es tracta d'una pantalla molt senzilla la qual serveix, per una banda per poder variar el temps de publicació de dades al robot i el temps de refresc de les dades als gràfics. I per l'altra banda, per poder manipular la plataforma de manera remota, per saber si tots els motors, guies, boles, elements de fixació, etc, funcionen correctament. Açò s'ha aconseguit amb un emulador d'un "joystick" per a la inclinació de la plataforma i un control del tipus "slide" per a l'altura d'aquesta. El control amb aquests elements s'activa i desactiva amb els botons visibles a la part superior de la pantalla.

5.2. MANUAL D'USUARI

5.2.1. Entrada d'usuaris

A l'hora de començar a treballar amb aquesta interfície, el primer que s'ha de fer és parlar amb l'enginyer perquè inscriba, en la llista d'usuaris, el nom i la contrasenya a utilitzar. S'haguera pogut ficar una secció per tal d'enregistrar usuaris, però s'ha considerat que no qualsevol ha de poder accedir al control del robot, sinó que l'enginyer, què és el responsable, ha de valorar qui deu de poder accedir i amb quin tipus de restriccions.

Una vegada en la llista, s'executa el programa, on apareixeran totes les pestanyes, però solament tindrà informació la d'entrada d'usuaris. En aquesta s'han d'escriure l'usuari i la contrasenya guardats i prémer el botó d'accedir (il·lustració 36), per tal que el programa comprove les dades aportades. Si el procés ha sigut exitós, al requadre de missatge indicarà que ho ha sigut i en cas contrari indicarà què és incorrecte. Quan ja s'haja accedit, és pot començar a controlar el robot en les altres pestanyes.



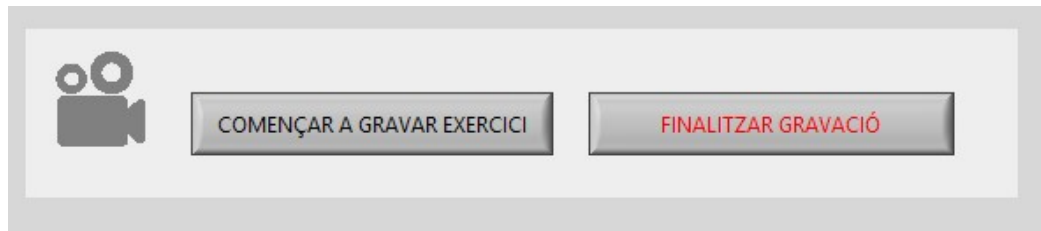
Il·lustració 36. Accés d'usuaris

En el cas que ja s'haguera fet ús del robot i es volguera canviar d'usuari o tancar el programa, s'ha de prémer el botó indicat per qualsevol d'ambdues coses, que es troben al requadre clar, a la part inferior de l'accés dels usuaris.

5.2.2. Pàgina principal de control

Els dos elements principals d'aquesta pestanya són l'àrea de gravar i la d'executar exercicis. Però abans de poder utilitzar-les s'han de fer unes eleccions. Primerament s'ha de triar quin tipus d'exercici es va a utilitzar amb l'interruptor. Posteriorment elegir el parell límit de seguretat, el qual està per predeterminat en 100 N·m què és el que hi havia fixat al robot del laboratori, i, si s'està en un exercici resistiu, fixar també les característiques de la referència de força a superar. Una vegada tot açò establert, es pot procedir al control.

Per tal de gravar un exercici nou (il·lustració 37), simplement el metge ha de polsar el botó de gravar exercici i anar junt al robot per fer els moviments desitjats, i quan finalitze la gravació polsar el botó corresponent. D'aquesta manera el programa guardarà l'altura i la inclinació de la plataforma, durant tot l'exercici, i anirà mostrant-ho en els gràfics, el que li permet al professional pegar-li una ullada per veure si s'ha realitzat com tenia previst. Una vegada finalitzada la gravació s'ha d'indicar el nom de l'arxiu i la ubicació per poder guardar les dades.



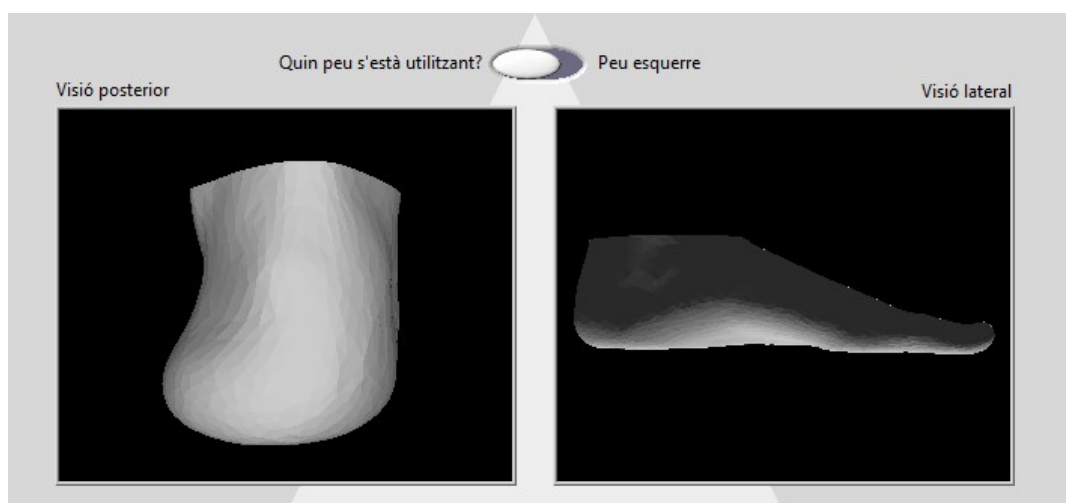
Il·lustració 37. Zona per gravar exercicis

Després d'haver gravat un exercici es procedeix a executar-lo (il·lustració 38). Solament s'ha d'introduir la ubicació d'aquest manualment o amb la icona de la carpeta, seleccionar les vegades que es vol repetir l'exercici i pulsar el botó d'executar. Mentre s'està executant l'exercici tots els gràfics s'estan actualitzant, per tant el metge pot saber el què està passant sense veure el robot en funcionament. Aquest exercici es pot parar a mitja execució tornant a pulsar el mateix botó, funcionant així com a botó de seguretat.



Il·lustració 38. Zona per executar exercicis

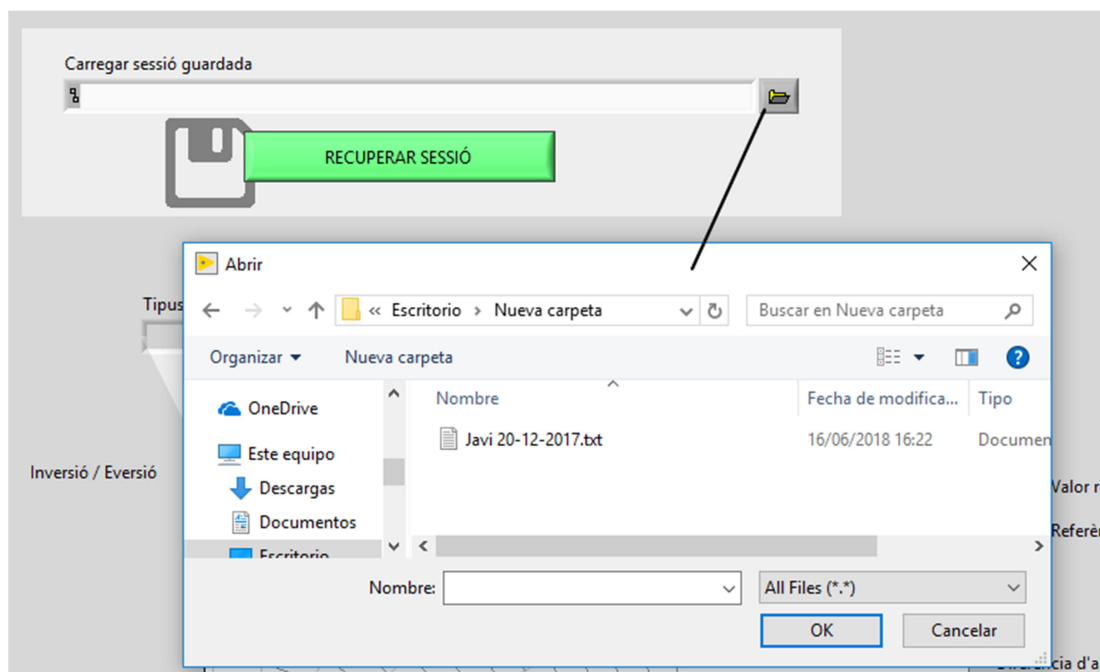
Encara que no s'ha comentat abans, durant l'execució d'un exercici es pot canviar la visió dels models 3D per si s'està realitzant l'exercici amb un peu o l'altre (il·lustració 39). Açò simplement facilita la visualització de les dades de l'angle per part del professional.



Il·lustració 39. Models 3D dels peus

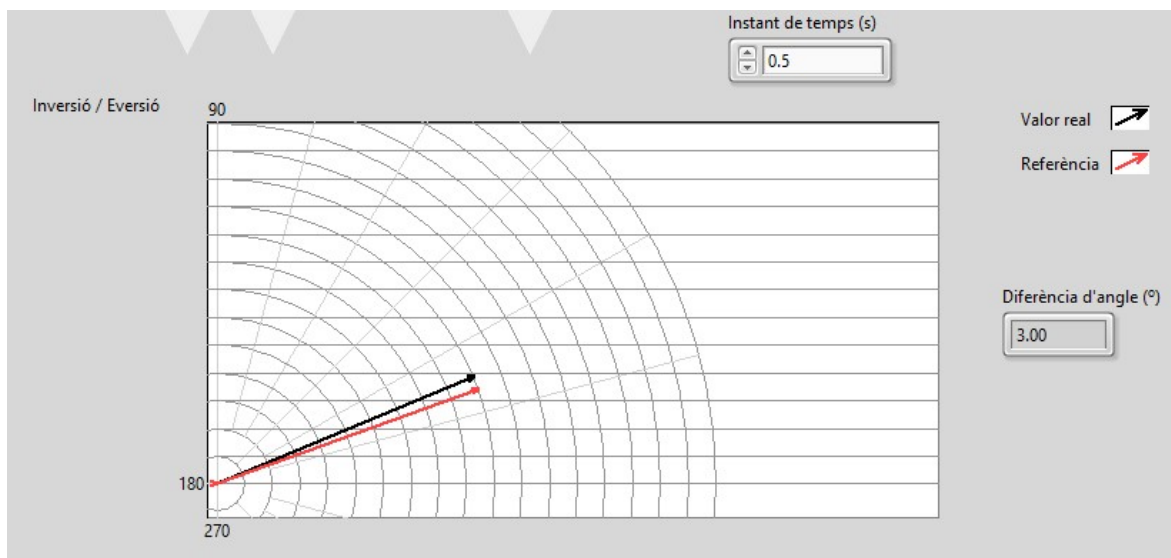
5.2.3. Recuperar sessions

La manera de procedir per tal de recuperar la informació d'una sessió gravada o realitzada anteriorment comença amb la introducció, a la casella de carregar sessió guardada, de la ubicació del fitxer on s'ha guardat aquesta, ja siga escrivint-ho a mà o polsant la icona de la carpeta i seleccionant el fitxer indicat (il·lustració 40). I una vegada introduïda la ruta es clica el botó de recuperar sessió.



Il·lustració 40. Carregar un exercici guardat

A continuació ja estaran tots els gràfics amb la informació recuperada, excepte els gràfics polars, per als quals s'haurà de seleccionar quin instant de la sessió es vol recuperar a través del control habilitat per a aquest objectiu. Per explicar açò un poc més: si un exercici consta de 2 minuts de duració, el gràfic polar pot recuperar solament les dades de una en una, per tant es poden visualitzar solament el angle real i la referència que hi havia en un moment determinat, per exemple als 0.5 segons (il·lustració 41).

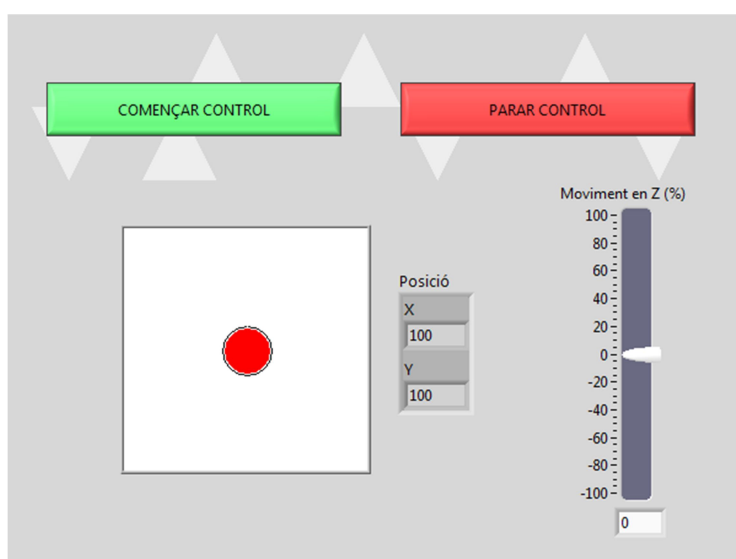


Il·lustració 41. Gràfic polar amb valors

Per finalitzar amb aquesta part, remarcar la presència, ja comentada abans, de les caselles per tal de fer visibles o invisibles les funcions dels gràfics d'altura del peu, força en l'eix Z i parells.

5.2.4. Control de l'enginyer

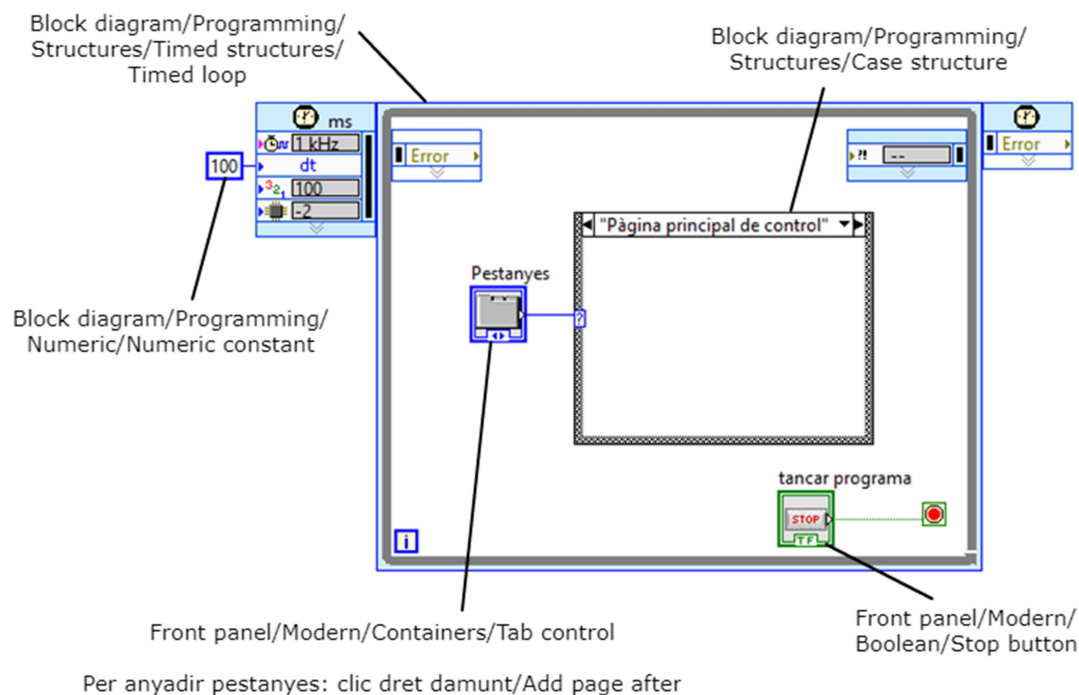
Els controls de temps es manipulen simplement canviant els valors disponibles als desitjats, i la part del "joystick" s'activa polsant el botó de començar control, el qual s'esperarà a què el robot faci el spline inicial i a partir d'este començarà a enviar les dades de referència de posició, tant si es varia aquest com si es modifica l'altura. Per tal d'acabar el control simplement s'ha de polsar el botó de finalitzar control.



Il·lustració 42. Control de la plataforma del robot

5.3. REPRODUCCIÓ DE LA INTERFÍCIE

Per tal de començar a programar la interfície s'ha de saber que l'estructura principal és un "timed loop" amb un temps de refresc de 100 ms i amb el tancament connectat al botó de tancar programa (il·lustració 43). Dins d'aquesta estructura es disposa un contenidor de pestanyes connectat a un "case structure", el qual permet que durant l'execució solament estiguen funcionant les parts del programa contingudes en la pestanya indicada.



Il·lustració 43. Estructura principal del programa

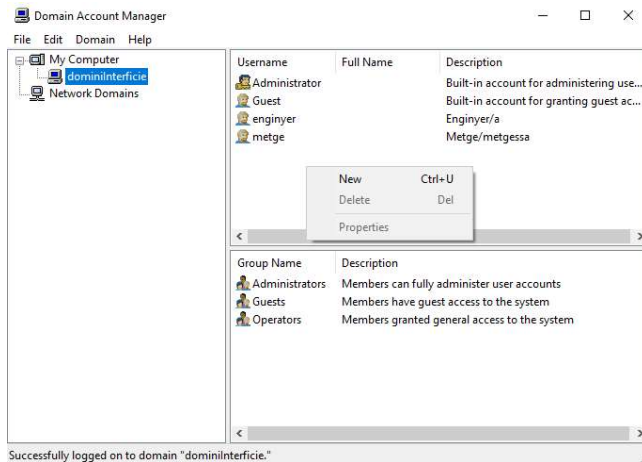
Amb açò ja es té l'estructura principal, i cadascuna de les explicacions següents estaran contingudes en la pestanya corresponent del "case structure".

5.3.1. Entrada d'usuaris

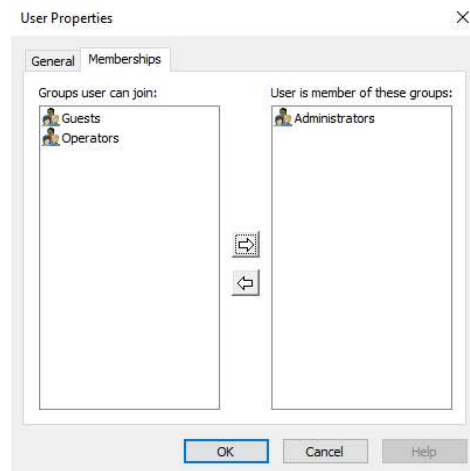
Per tal de crear els usuaris amb les contrasenyes associades s'ha d'accedir a "Tools/Security/Domain Account Manager" i una vegada ací es fa clic dret sobre "My computer" i s'accedeix a "New local domain". Aplegats a aquest punt, s'escriu el nom del domini, en el nostre cas "dominiInterficie", i s'estableixen les característiques de les longituds de les contrasenyes. Es clica "OK" i es guarda una clau per a l'administrador.

A continuació, en la finestra del "Domain Account Manager" hi ha dos usuaris i tres grups d'usuaris creats per part del programa. El que s'ha de fer és: clic dret en el caixetí dels usuaris i prémer "New" (il·lustració 44), ací es fica el nom ("User Name") i la contrasenya ("Change Password"). Canviant de pestanya a "Membership" es pot associar l'usuari a un grup

o a un altre segons les característiques de les que es vaja a dotar, fent clic sobre el grup i passant-lo, amb la fletxa de la pantalla, a "User is membre of these groups:" (il·lustració 45). Es polsa "OK" i ja es té l'usuari creat.



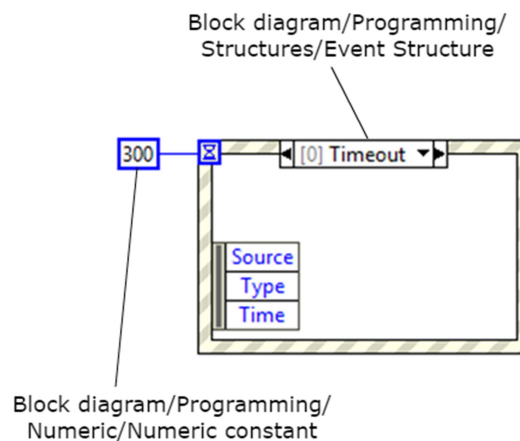
Il·lustració 44. Registre d'usuaris



Il·lustració 45. Afegir els usuaris als grups

Amb tots els usuaris creats ja es pot procedir a la programació de la interfície.

En "Front Panel" es creen dos "Modern/Boolean/OK button" (accedir i tancar usuari) i un "stop button" (tancar programa). Després, es genera una "event Structure" al diagrama de blocs, on, en el terminal de "temps de timeout", dalt a l'esquerra, es fica "300", per tal que, si no ha interromput ningun dels esdeveniments enregistrats en 300 ms s'execute el codi de "timeout" (il·lustració 46).

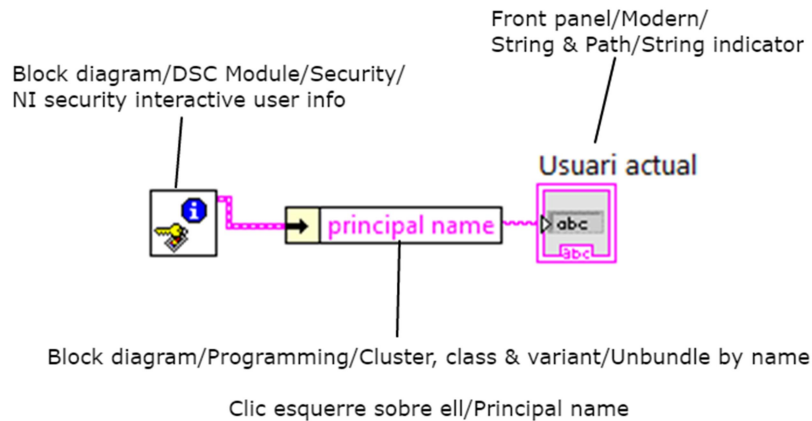


Il·lustració 46. Event structure

Per tal de crear els esdeveniments, es fa clic dret sobre la "event structure" i es polsa "Add event case...". Amb la finestra nova, en "event sources" es busca el botó d'accedir i en

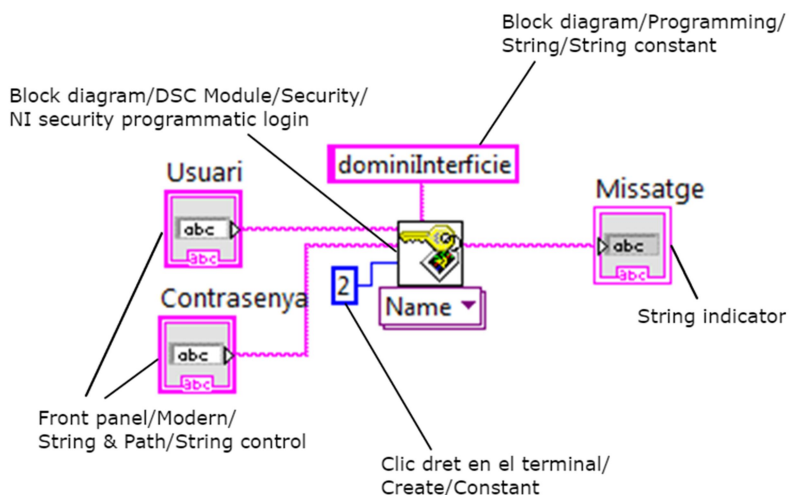
“events” es busca “Mouse/Mouse down”. Es finalitza amb “OK” i es repeteix el mateix per al botó de tancar usuari.

Dins de l'esdeveniment de “timeout” s'ha disposat una estructura que permet indicar quin és l'usuari actual, per tant si es supera el “temps de timeout” simplement es refresca la informació a l'indicador destinat a tal fi (il·lustració 47).



Il·lustració 47. Codi per refrescar l'usuari actual

En l'esdeveniment del botó d'accedir s'utilitza el “NI security programmatic login” (il·lustració 48), el qual desenvolupa la funció de comprovar si els camps introduïts a l'usuari i la contrasenya són correctes. L'única diferència existent en els camps esmentats és el fet que la contrasenya està modificada per tal que no mostre el que s'està escrivint, sinó asteriscs en el seu lloc. Aquesta característica és modificable amb “clic dret sobre el controlador/Properties/Appearance/Display style/Password”.



Il·lustració 48. Codi per accedir els usuaris

L'últim esdeveniment permet simplement el tancament de l'usuari existent, el què està suportat pel "NI security programmatic logout" (il·lustració 49), on no hi ha que connectar ninguna funció de LabVIEW.

Block diagram/DSC Module/Security/
NI security programmatic logout



Il·lustració 49. Codi per tancar els usuaris

Havent creat la part de la programació, es procedeix a restringir l'accés per als diferents usuaris. La restricció consisteix, per a cadascun dels indicadors, controls i gràfics que es troben a les diferents pestanyes "clic dret sobre ells/Properties/Security", i una vegada en aquest diàleg afegir cada usuari, o grups d'usuari, amb l'accessibilitat que es desitge per a l'element escollit. D'aquesta manera es pot controlar que solament un usuari pugui utilitzar certs elements o inclòs que es puguin veure però no controlar-los.

5.3.2. Pàgina principal de control













Amb l'objectiu d'entendre correctament l'organització d'aquesta secció de la programació, començar explicant que totes les funcionalitats utilitzades estan basades en instruments virtuals paral·lels, és a dir, que tots els instruments virtuals que s'expliquen a continuació estan organitzats per tal que funcionen al mateix temps durant l'execució. Per a aquest fet s'han d'arrastrar dins del "case structure" principal i connectar les referències pertinents.

Abans d'exposar els instruments paral·lels s'exposaran, en la taula 4, totes les dades que s'han de connectar a aquests (botons, gràfics, indicadors, etc). Totes aquestes han de connectar-se als instruments amb referències per què funcionen correctament.

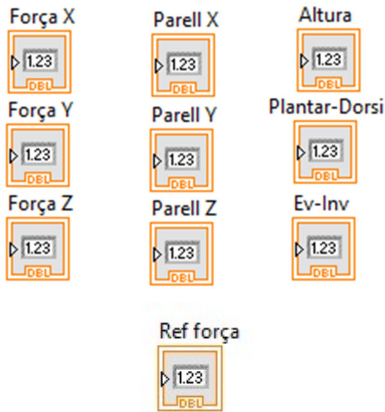

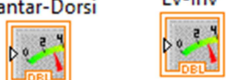
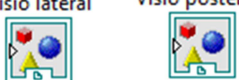



Taula 4. Llistat de controladors, indicadors, etc, que s'han de generar

Icona	Tipus de variable	Utilitat
<p>Elecció exercici</p>	Vertical toggle switch	Permet elegir entre els dos tipus d'exercicis
<p>Referència força?</p>	Round led	Arreplega si es necessita la referència de força o no (no visible)

Desenvolupament d'una interfície amigable basada en LabVIEW
per al control i supervisió de robots paral·lels de rehabilitació

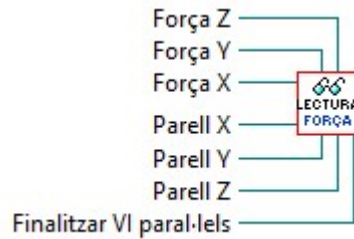
<p>Tipus d'exercici</p> 	Numeric indicator	Tipus d'exercici que s'està executant (no visible)
<p>Parell límit de seguretat (N·m)</p> 	Numeric indicator	Marca el parell màxim que es pot aplicar sobre el sensor (mesura de seguretat)
<p>Parell màxim (N·m) Període (s)</p>  	Numeric indicator	Paràmetres per a la referència de força dels exercicis resistius
<p>Text parell a superar</p> 	String indicator	Permet mostrar el text superior de la referència de força
<p>Reproduir gràfics?</p> 	Stop button	Permet saber si s'han de reproduir els gràfics (no visible)
<p>Finalitzar VI paral·lels</p> 	Stop button	Indica quan s'han de parar els instruments virtuals paral·lels, ja que si no es tancaren no es podria accedir a les altres pestanyes (no visible)
<p>Activitat guardada a carregar</p> 	File path	Indica la ubicació de l'exercici a carregar
<p>Nombre de repeticions</p> 	Numeric indicator	Marca les repeticions que s'han de realitzar de l'exercici
<p>Executar exercici</p> 	OK button	Comença a executar i para l'exercici indicat al "file path"
<p>Gravar exercici Finalitzar gravació</p>  	OK button Stop button	Per començar i finalitzar la gravació d'exercicis

Desenvolupament d'una interfície amigable basada en LabVIEW
per al control i supervisió de robots paral·lels de rehabilitació

 <p>Força X Força Y Força Z Parell X Parell Y Parell Z Altura Plantar-Dorsi Ev-Inv Ref força</p>	<p>Numeric indicator</p>	<p>Indicadors per tal de mostrar els valors llegits del robot i utilitzar-los posteriorment als gràfics (els únics visibles al programa són els de Plantar-Dorsi i Ev-Inv, en la part inferior del models 3D)</p>
 <p>Gràfic força Gràfic parells Gràfic altura</p>	<p>Waveform chart</p>	<p>Gràfics</p>
 <p>Plantar-Dorsi Ev-Inv</p>	<p>Meter</p>	<p>Indicador d'angles perillosos per al peu (zones amb colors)</p>
 <p>Visió lateral Visió posterior</p>	<p>3D picture</p>	<p>Mostra la inclinació del peu amb models 3D</p>
<p>Quin peu s'està utilitzant?</p> 	<p>Slide switch</p>	<p>Per tal d'eleger el peu a utilitzar en l'exercici i mostrar-lo així als models 3D</p>
<p>Temps de refresc dels gràfics (ms)</p>  <p>Període de publicació de dades (ms)</p> 	<p>Numeric indicator</p>	<p>Permeten regular la publicació al robot i el refresc dels gràfics (es troben a la pestanya de control de l'enginyer, sols aquest pot manipular-los)</p>

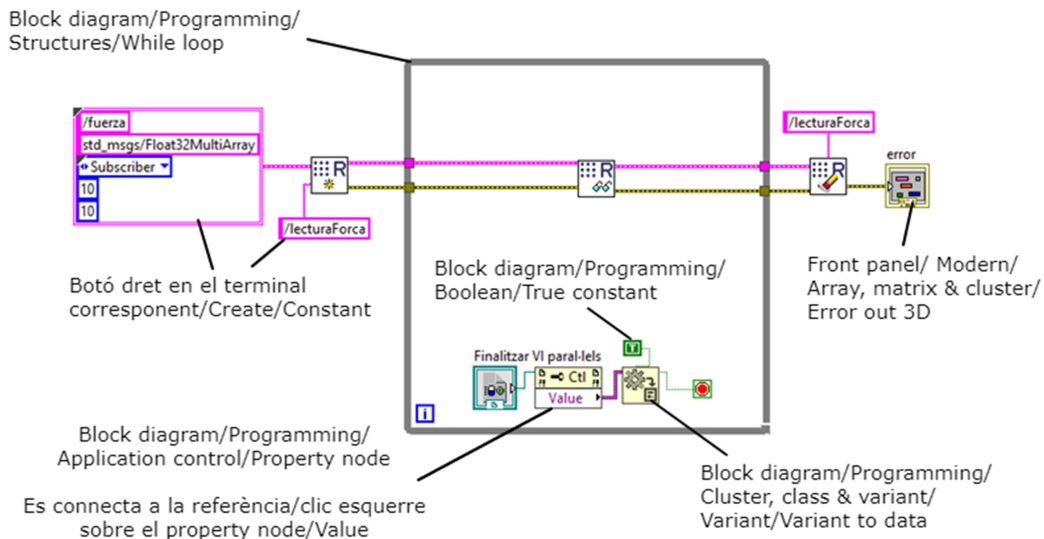
Una vegada creats tots aquest es pot procedir a generar els instruments virtuals (d'ara en avant VI). Els codis complets dels VI d'aquest apartat es troben al propi apartat o a l'annex 2.

- Lectura de força: com indica el nom, llegeix els paràmetres de força i parell del sensor present al robot.



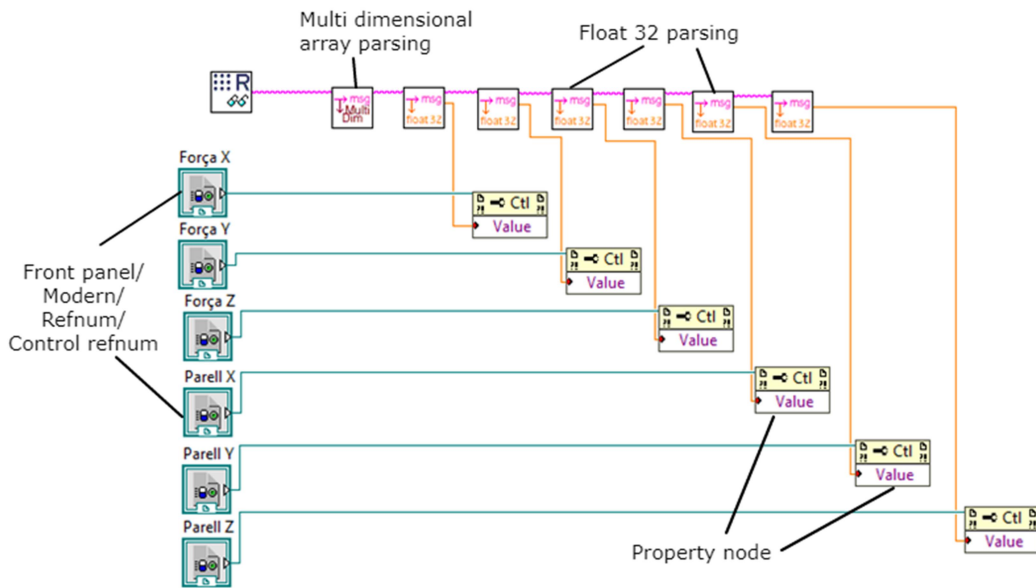
Il·lustració 50. VI lectura de força

Aquest instrument virtual està compost per un bucle “while”, el terminal de finalització del qual està controlat pel botó de “Finalitzar VI paral·lels”. Per tal de llegir el missatge primerament es crea el node i s'indiquen el topic i les característiques d'aquest a tractar amb el “ROS topic init”, posteriorment es passa per l'instrument de lectura i, quan es vulga aturar la lectura, es tanca el node abans creat (il·lustració 51). Aquest és el procediment a seguir sempre que es desitge llegir o escriure en algun topic.



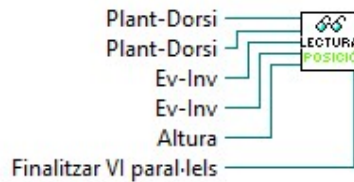
Il·lustració 51. Codi per llegir la força

Amb el missatge llegit solament queda connectar el terminal de “reply” del VI de lectura als instruments de traducció dels missatges i enviar els valors als indicadors de l'instrument virtual principal a través de les referències i els “property nodes” (il·lustració 52). Aquesta part va també a l'interior del bucle.



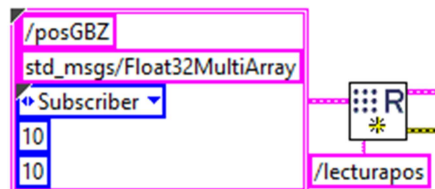
Il·lustració 52. Codi per passar la força als indicadors

- Lectura de posició: permet la lectura de l'altura i dels dos angles de la plataforma.



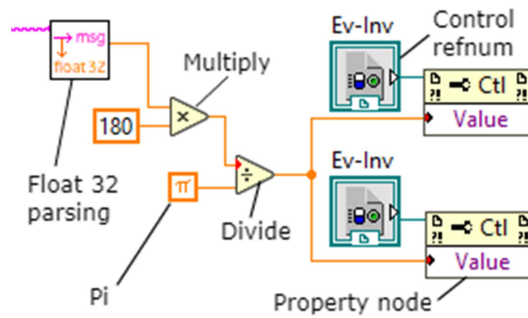
Il·lustració 53. VI lectura de posició

Es tracta de la mateixa estructura que l'instrument virtual de lectura de força però amb les especificacions del topic de posició (il·lustració 54).



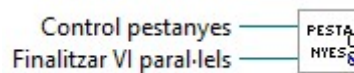
Il·lustració 54. Topic de posició

L'única diferència remarcable entre aquest i el de força resideix en la transformació que es fa de radiants a graus (il·lustració 55).



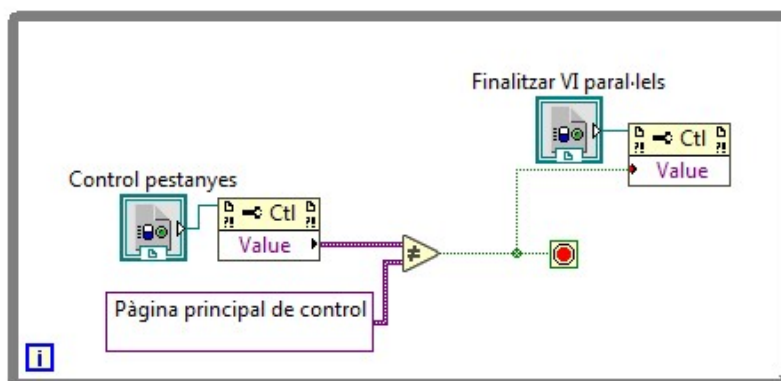
Il·lustració 55. Codi de transformació dels valors a graus

- Tancament dels instruments virtuals paral·lels: aquest és el més important de tots els instruments virtuals presents, ja que si aquest no tancara correctament tots els VI paral·lels, el programa es congelaria i no es podria continuar amb l'execució.



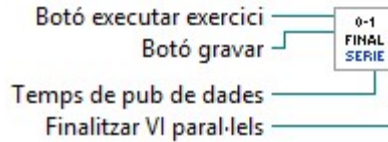
Il·lustració 56. VI tancament dels VI paral·lels

Està format per un bucle "while", dintre del qual ens trobem amb el control de pestanyes, que no és més que la referència del "tab control" anomenat "pestanyes", explicat al principi de la programació del codi. Aquest es compara amb una constant per tal de saber si s'ha canviat de pestanya, i si ho ha fet, activa el botó de "finalitzar VI paral·lels" per tal de parar tots aquests (il·lustració 57).



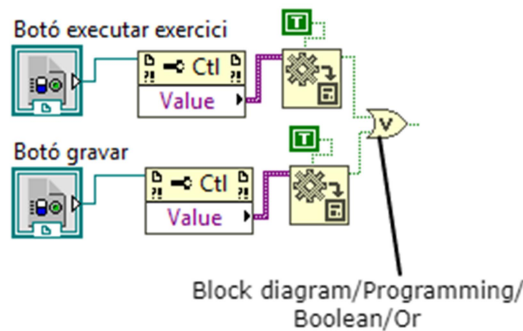
Il·lustració 57. Codi complet tancament dels VI paral·lels

- Indicador del final d'un exercici de gravació o d'execució al robot: quan s'està executant un exercici arriba un punt on s'acaben les dades i li s'ha d'indicar al robot açò que ha ocorregut.



Il·lustració 58. VI indicador del final d'un procés

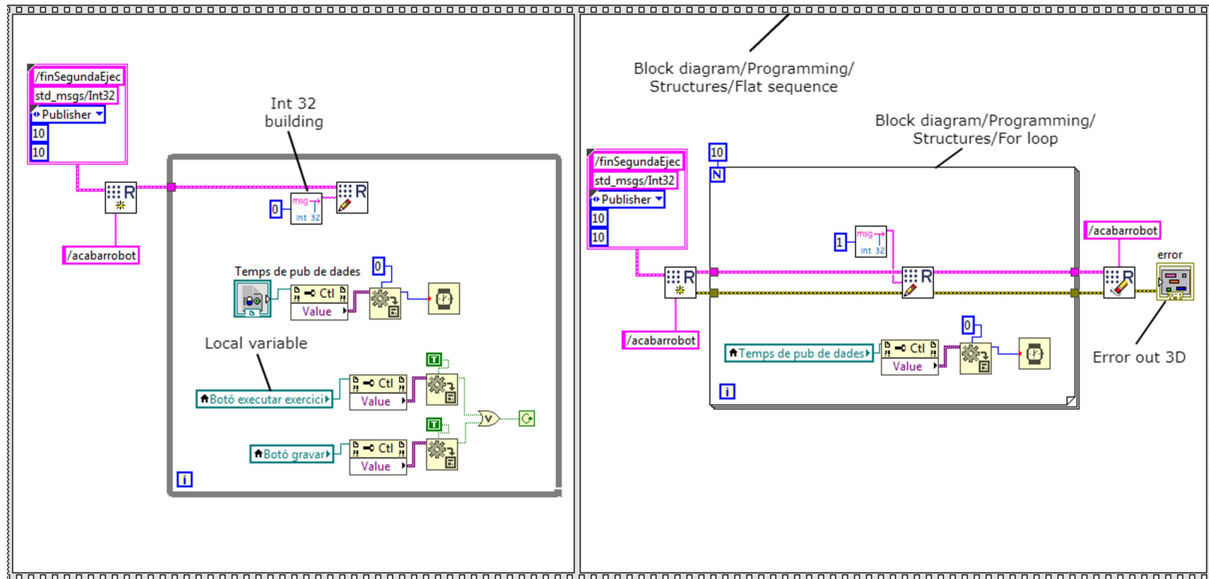
Aquest està estructurat amb un "timed loop" amb repeticions cada 500 ms i el tancament controlat per l'usual "finalitzar VI paral·lels". Dins d'aquest es disposa d'un "case structure" amb el condicional connectat a l'indicat a la il·lustració 59, és a dir, si s'executa el gravar o executar exercici, s'activaria el "true".



Il·lustració 59. Codi condicional per al "case structure"

Ja que en el cas de "false" no es reproduïx ningun codi, tot el següent estarà en el "true". Ací es troba una "flat sequence" el primer requadre de la qual simplement publicarà zeros amb la periodicitat de "temps de publicació de dades", fins que ambdós botons abans indicats passen a estar apagats. Amb açò es passarà al segon requadre, el qual publicarà deu uns (per tal d'assegurar-se que el robot rep el 1) al mateix topic i finalitzarà el "flat sequence" (il·lustració 60).

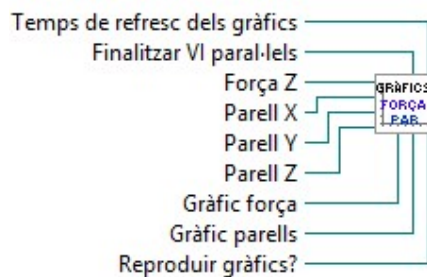
Desenvolupament d'una interfície amigable basada en LabVIEW per al control i supervisió de robots paral·lels de rehabilitació



Il·lustració 60. Codi per publicar el final del procés

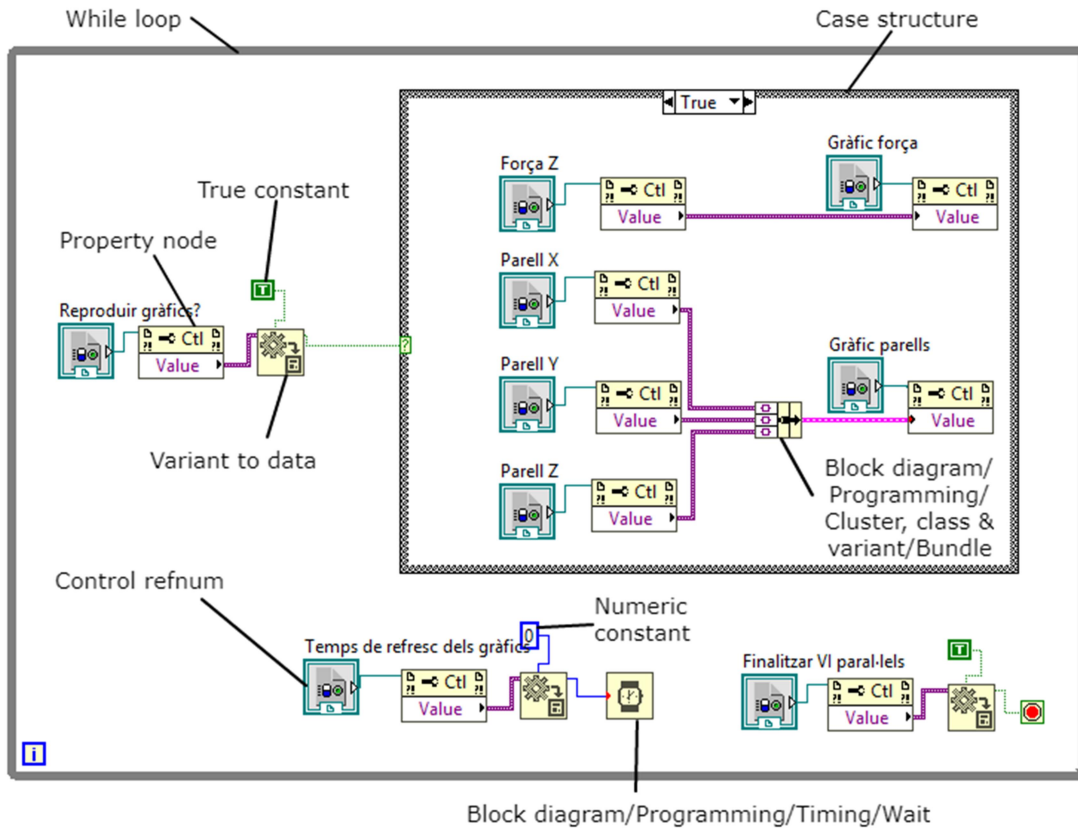
Per tant, quan es comença a gravar algun exercici o es comencen a executar exercicis comença a publicar zeros al topic fins que s'acabe de fer l'acció, quan publicarà els uns.

- Gràfics força i parells: agafa els valors llegits de força i parell i els dibuixa en el gràfic corresponent.



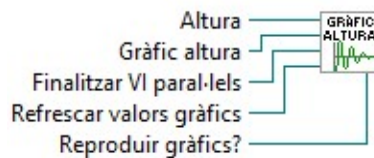
Il·lustració 61. VI per actualitzar els gràfics de força i parells

Està format per un bucle "while" la finalització del qual està controlada pel mateix botó que els anteriors instruments, i es repeteix a un freqüència marcada pel "temps de refresc dels gràfics". A més a més, disposa d'una "case structure" a l'interior que permet reproduir o no els gràfics segons se li indica (a la opció de "false" no hi ha codi, està buida). Per tal de dibuixar els valors als gràfics solament s'han de passar els valors al gràfic pertinent, com s'ha fet al "true" (il·lustració 62).



II·lustració 62. Codi complet gràfics força i parells

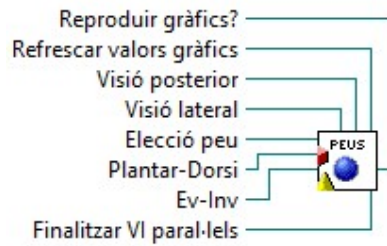
- Gràfic altura: permet la visualització del gràfic d'altura.



II·lustració 63. VI per actualitzar el gràfic d'altura

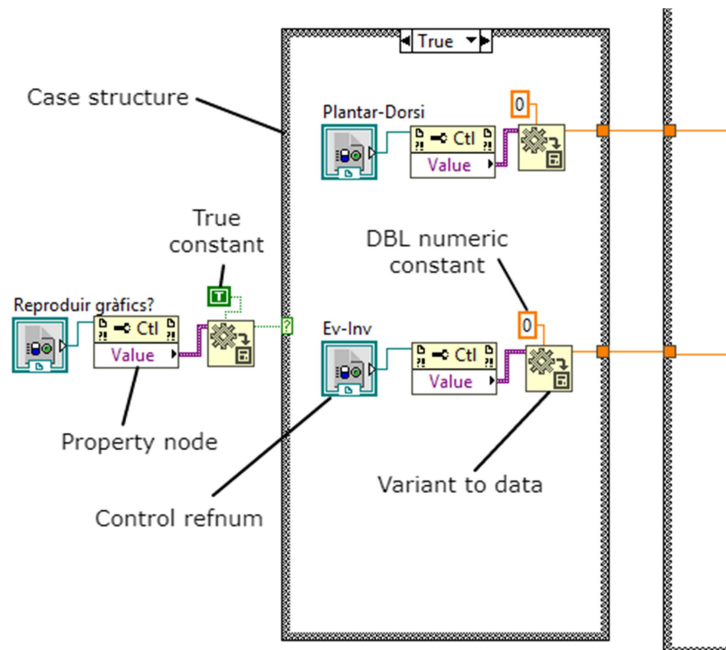
L'estructura és igual al instrument de "gràfics força i parell" però aplicada a l'altura de la plataforma i al gràfic corresponent a aquesta.

- Models 3D dels peus: disposant dels valors d'inclinació del peu del pacient, aquests es recreen en un model 3D mostrat per pantalla.



Il·lustració 64. VI per mostrar els models 3D

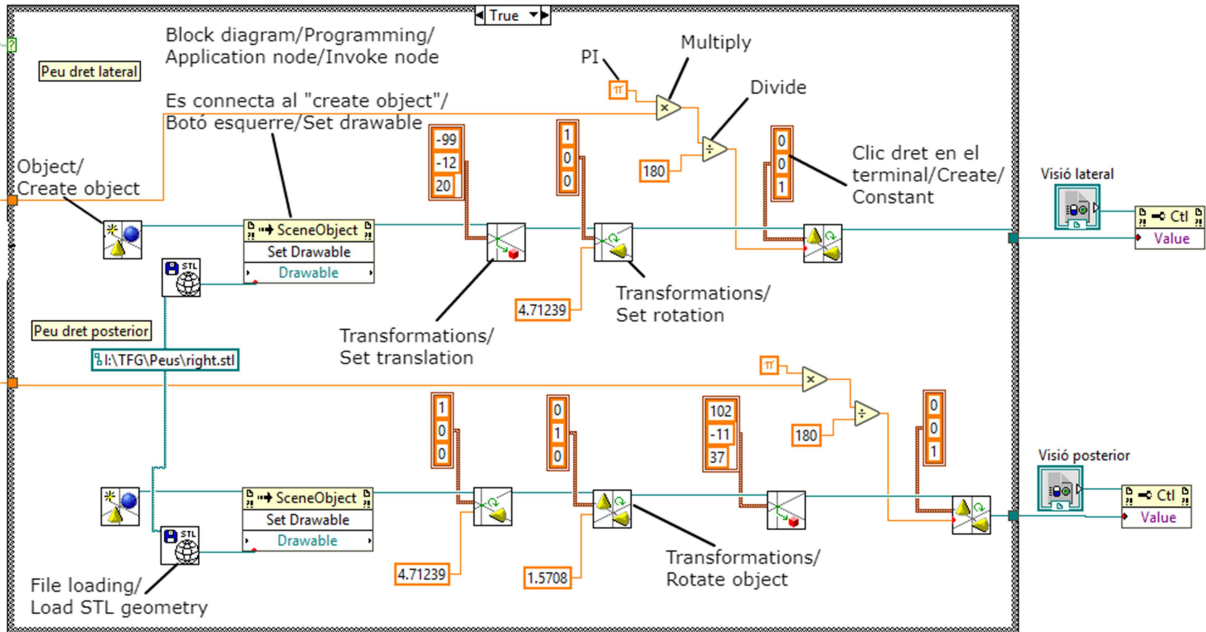
Disposa de la mateixa estructura principal que els altres gràfics amb el bucle “while”, el mateix botó per tal de finalitzar el bucle i el mateix temps de refresc de valors. Primerament el que es fa es arrebregar els valors de la lectura de posició (il·lustració 65). És important saber que aquest gràfic està sempre actiu, però quan no ha d'estar-ho simplement li arriba un 0 com a valor, és a dir, en el “false” arriba un zero als terminals taronja.



Il·lustració 65. Codi per enviar valors als models 3D

A continuació dels cables de color taronja ens trobem amb un altre “case structure” el qual depèn del botó de “Quin peu s’està utilitzant?”. Dins d’aquest es produeix l’apertura dels models 3D, es procedeix amb una sèrie de transformacions de posició que permeten que els models estiguen centrats a les finestres destinades per a tal fet, i en última instància es rota (transformació del tipus “rotate object”) per poder mostrar el valor rebut als indicadors (aquesta presenta una transformació de graus a radians, ja que es treballa amb aquests). En la il·lustració 66 tots els mòduls que

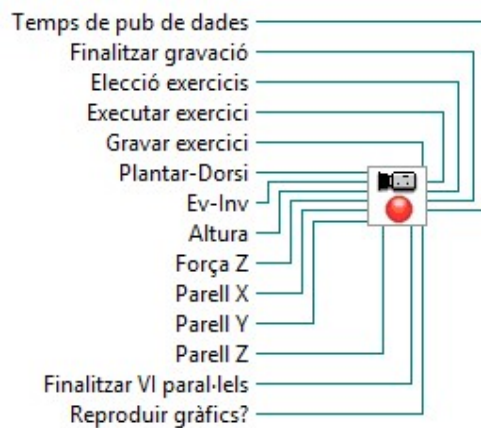
encara no havien sigut explicats s'obtenen de "Block diagram/Programming/Graphics & sound/3D Picture control".



Il·lustració 66. Codi per centrar els models 3D i orientar-los

En la situació de "false" trobem una estructura quasi exacta per al peu esquerre, però amb transformacions lleugerament diferents.

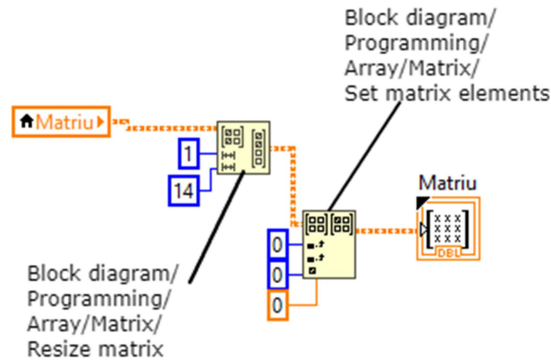
- Gravar exercici: cobreix la necessitat de gravar els exercicis per part del professional mèdic.



Il·lustració 67. VI per gravar un exercici

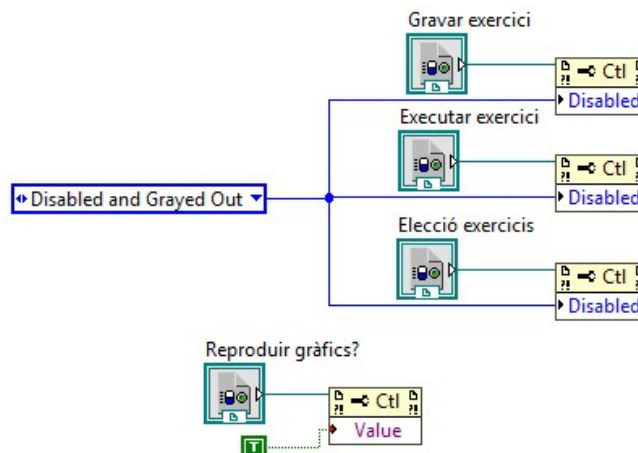
A la part més exterior de l'estructura ens trobem un "timed loop" amb un refresc de 500 ms condicionat pel "finalitzar VI paral·lels". Entrant cap endins trobem un "case structure", condicionat al botó de gravar, amb el condicional fals buit i el

vertader amb un "flat sequence" de tres requadres. En el primer d'aquests s'executen diferents esdeveniments per tal de poder passar al segon. Entre aquests trobem la generació d'una matriu de 14 columnes i 1 fila de zeros per tal de marcar que és un exercici gravat (il·lustració 68).



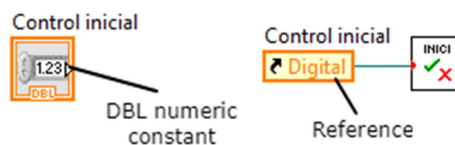
Il·lustració 68. Codi per generar una matriu amb una fila de 0

També es desactiven certs botons per evitar que el metge no pugui pulsar-los durant l'execució de la gravació i s'habilita el dibuixat dels gràfics (il·lustració 69).



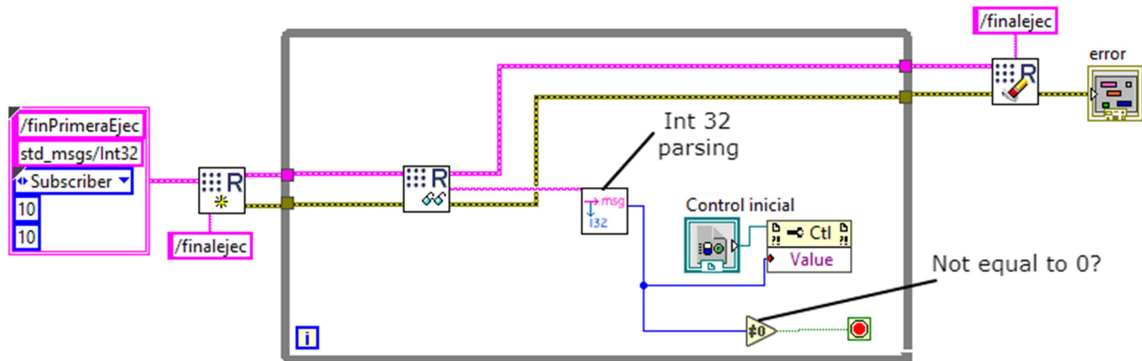
Il·lustració 69. Codi per desactivar botons i habilitar els gràfics

Per últim dins del primer requadre es troba un instrument en paral·lel que comprova quan el robot ha arribat al final del spline de pujada (il·lustració 70).



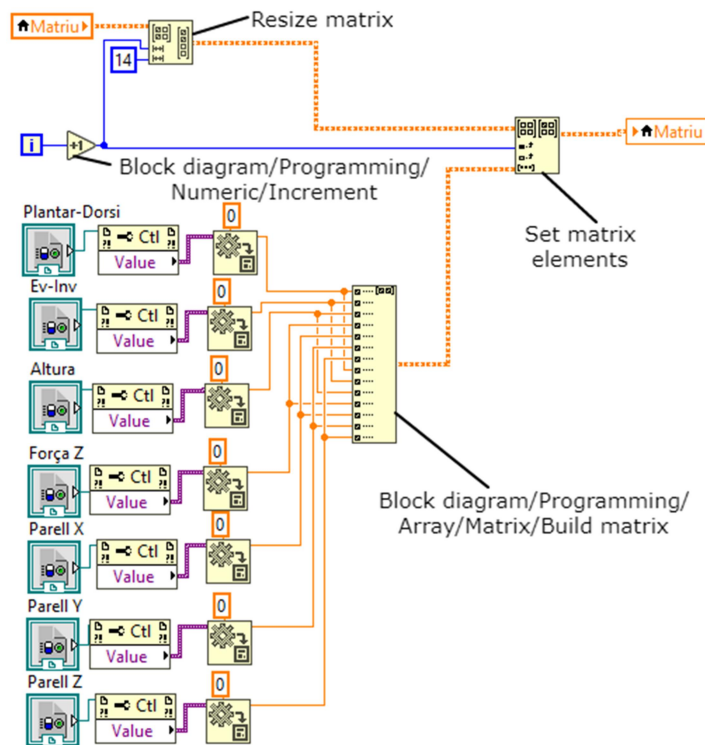
Il·lustració 70. VI per esperar la posició estable del robot

Dins d'aquest trobem la lectura d'un topic i la comparació de si és un 0 o no (il·lustració 71). Quan és diferent de 0 simplement finalitza el VI i li permet continuar al "flat sequence".



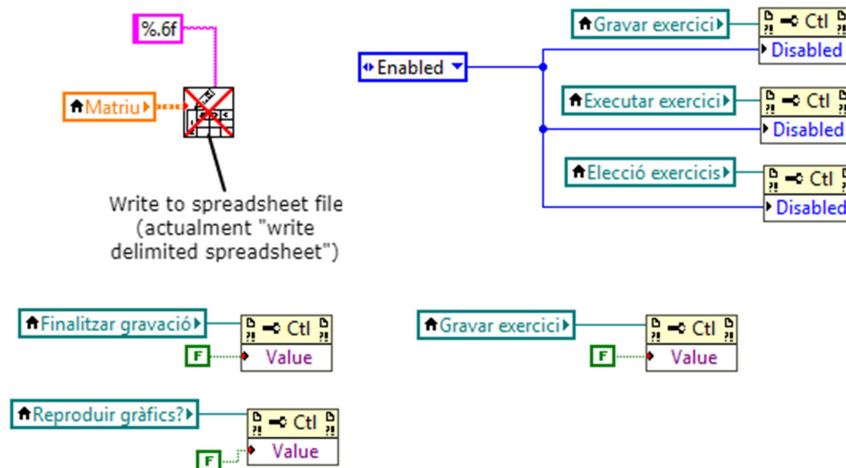
Il·lustració 71. Codi per esperar la posició estable del robot

Ja en el segon requadre es pot veure un bucle "while" marcat per a repetir-se cada "temps de publicació de dades" i amb el terminal de finalitzar unit al botó de finalitzar gravació. Dins d'aquest hi ha un redimensionat de la matriu anterior afegint una fila en cada iteració, en la què es posaran els nombres llegits als indicadors on aboquen els valors els VI de lectura. S'ha decidit copiar cada valor en 2 columnes per tal de permetre recuperar també les sessions gravades en la pestanya destinada a tal finalitat (il·lustració 72).



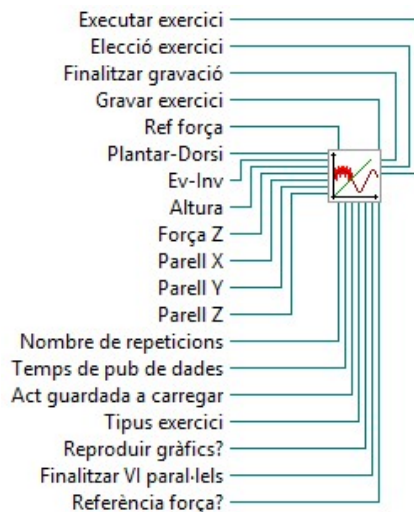
Il·lustració 72. Codi per guardar els valors a una matriu

I l'últim requadre del "flat sequence" s'ha utilitzat per guardar la matriu amb les dades a un fitxer, habilitar els botons abans desactivats i restablir el valor de fals als botons utilitzats per permetre la seua utilització posterior (il·lustració 73).



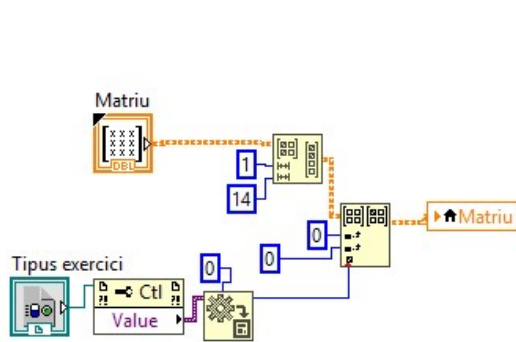
Il·lustració 73. Codi per habilitar botons, reactivar botons i guardar dades

- Executar exercici: permet posar en marxa la utilitat principal d'aquesta interfície, l'execució d'exercicis gravats.

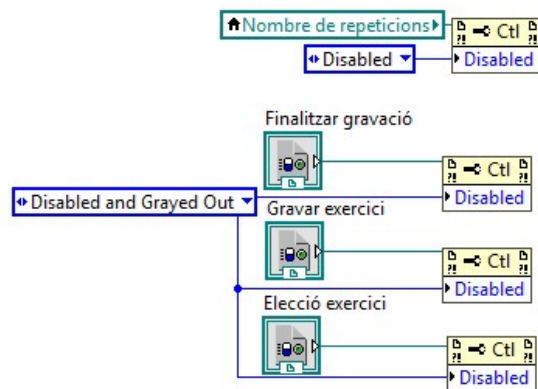


Il·lustració 74. VI per executar un exercici

En aquest es troba una estructura igual a l'anterior exceptuant el condicionant del "case structure" que ara és el botó d'executar exercici. Dins del primer requadre del "flat sequence" trobem el mateix de l'anterior instrument, exceptuant:

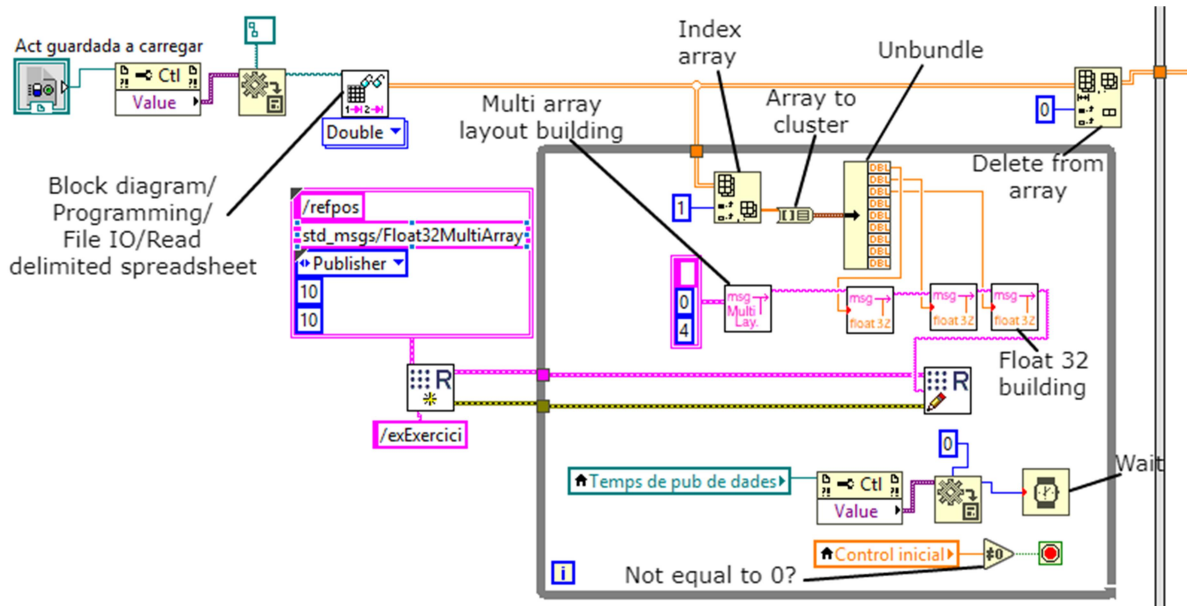


Il·lustració 75. Codi per generar una matriu amb el tipus d'exercici a la primera fila



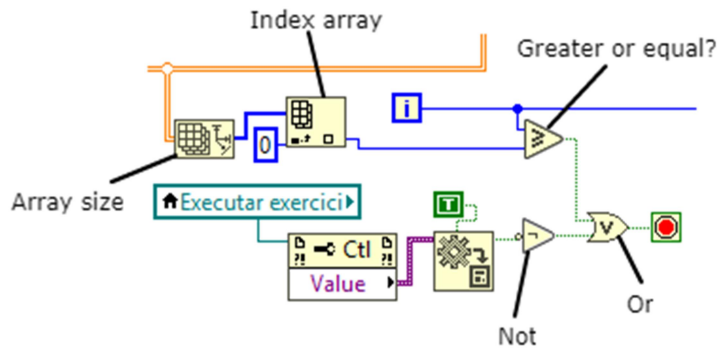
Il·lustració 76. Codi per desactivar botons

També s'ha d'afegir el codi de la il·lustració 77 al mateix requadre. Aquest permet primerament obrir l'arxiu indicat a la ubicació i passar-lo a una matriu, després se li extrau la segona fila per a dins del bucle "while", el qual publica contínuament aquest valor durant el spline inicial per mantenir el canal de comunicació obert. A partir d'ací s'elimina la primera fila de la matriu, la qual solament indica el tipus d'exercici que es tracta (en aquests casos indica que és un exercici gravat), i es passa al segon requadre de funcionament.



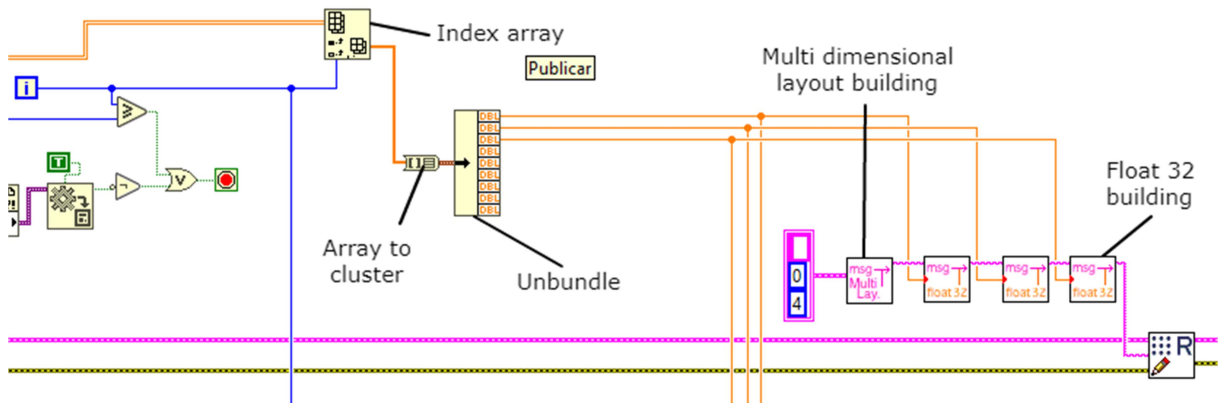
Il·lustració 77. Codi per obrir un arxiu i publicar el primer valor

En el segon requadre es té un bucle “for” per tal de repetir l'exercici el nombre de vegades indicades al controlador, i se li dona una “true constant” a la variable de referència de força per activar-la si fora un exercici resistiu. Dins del bucle abans mencionat es crea un bucle “while” condicionat pel nombre de files de l'arxiu i pel botó de parar execució de l'exercici (il·lustració 78), que és el d'executar quan està en “true”.



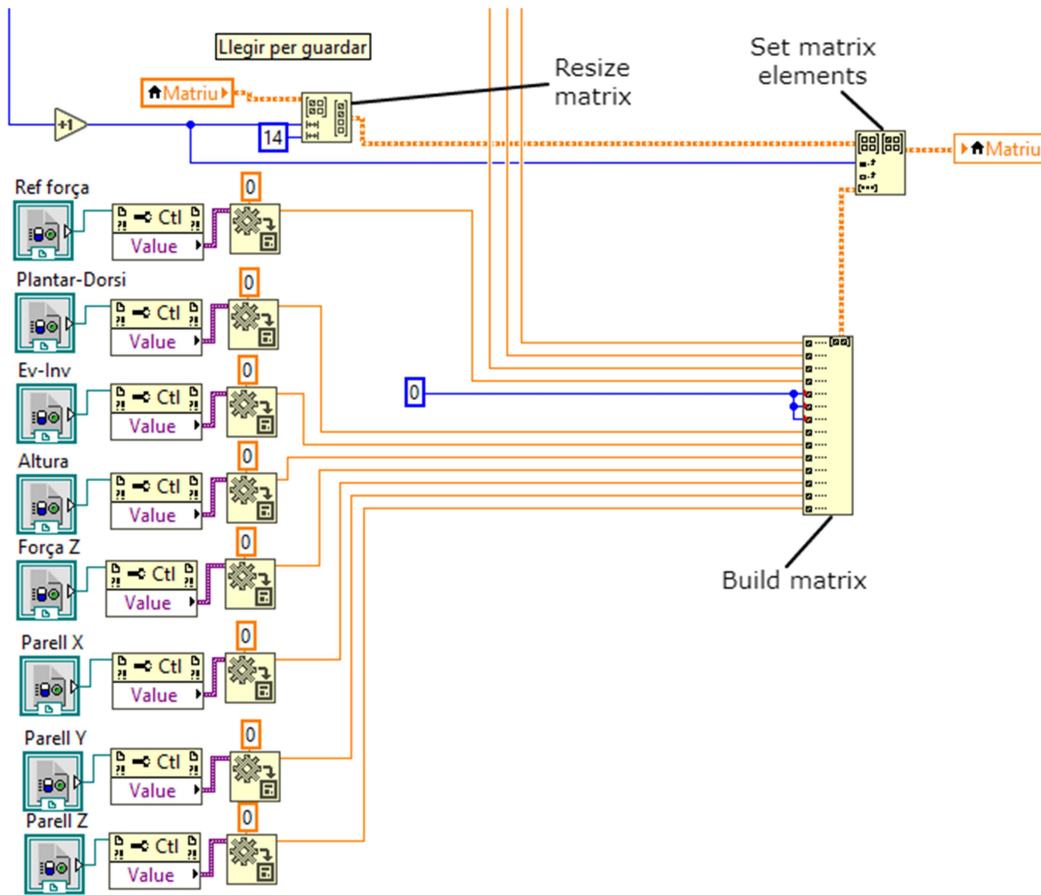
Il·lustració 78. Codi per condicionar el bucle “while”

Dins d'aquest “while” es troben dues funcionalitats diferenciades. La primera és la de publicació de la posició a assolir per part del robot, la qual es processa agafant la filera corresponent a la iteració actual i publicant els tres primers valors (valors dels angles i altura de la plataforma) (il·lustració 79).



Il·lustració 79. Codi per publicar valors de l'arxiu

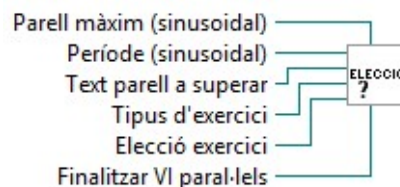
La segona es correspon amb la recaptació de dades dels sensors del robot, que es fa llegint en cada instant el valor de l'indicador corresponent i passant-lo a la matriu, juntament amb la referència d'aquests (que serà 0 si es tracta d'un exercici passiu) (il·lustració 80).



Il·lustració 80. Codi per guardar valors

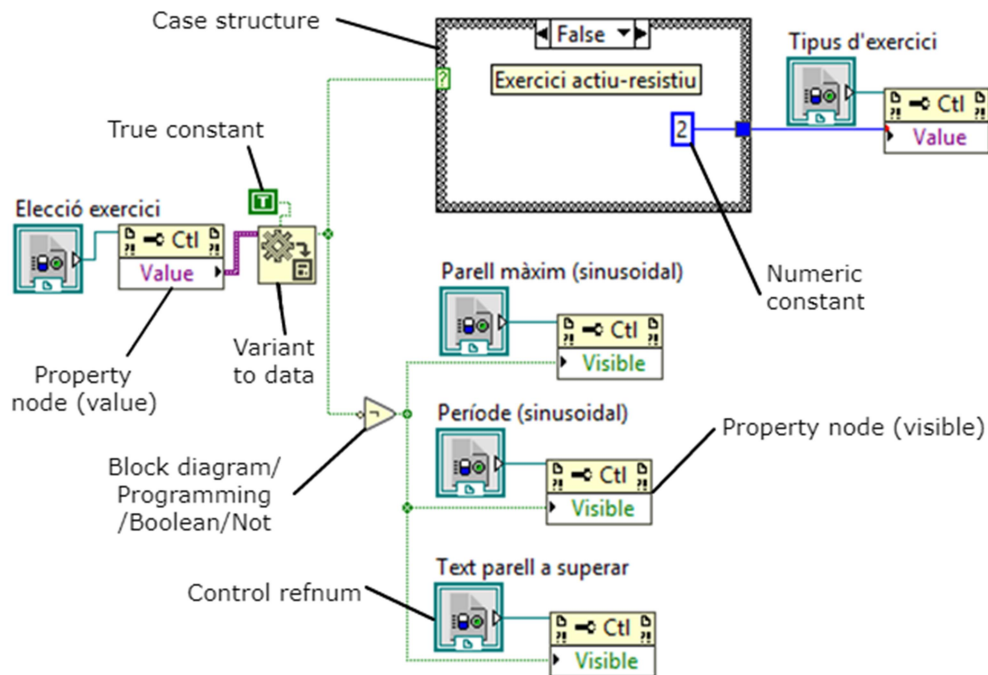
Passant al tercer i últim dels requadres, hi ha la reactivació dels botons desactivats a l'inici, la posada en fals dels botons utilitzats durant aquest procés d'execució i l'acció de guardar la matriu de dades.

- Canvis per l'elecció del tipus d'exercici: l'objectiu d'aquest és permetre el correcte funcionament de l'interruptor d'elecció d'exercici i aplicar els canvis convenients a la pantalla.



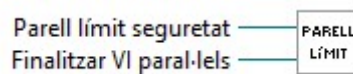
Il·lustració 81. VI d'elecció del tipus d'exercici

Tots els canvis depenen directament del valor de l'interruptor abans indicat. A la part superior del codi s'aplica el nombre indicatiu del tipus d'exercici (2 per als resistius i 1 per als passius) i a la part inferior es pot veure l'aplicació de visibilitat dels elements de la referència de força (il·lustració 82). Tot açò, com en el cas de tots els instruments virtuals paral·lels, va dins d'un bucle "while" amb el tancament d'aquest governat pel botó "finalitzar VI paral·lels".



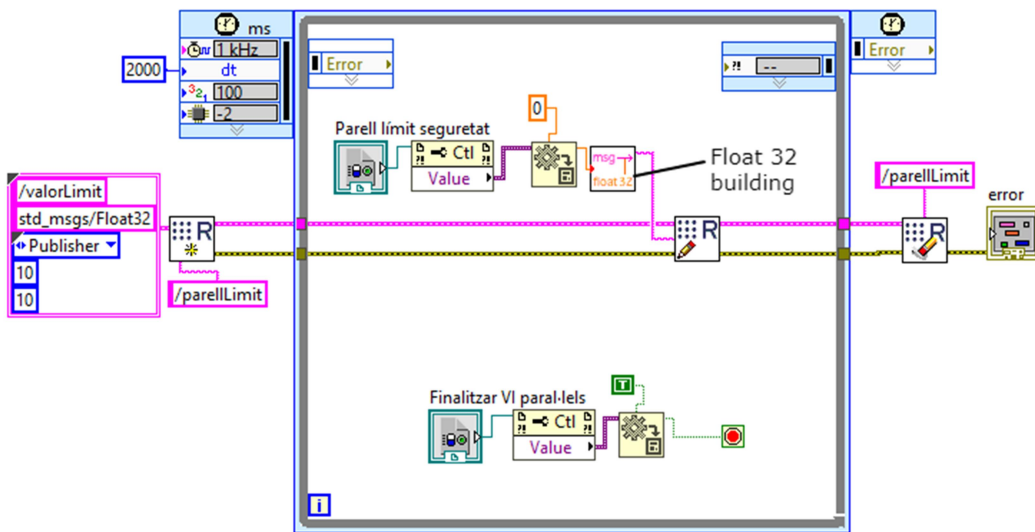
Il·lustració 82. Codi per elegir el tipus d'exercici

- Establiment del límit de seguretat: serveix com a seguretat per si en algun cas el pacient en algun moviment superara el límit marcat.



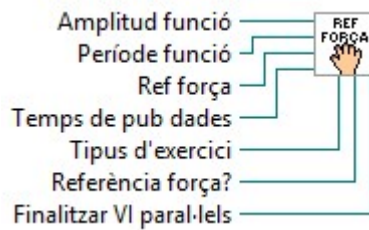
Il·lustració 83. VI límit de seguretat

Està compost per un "timed loop" marcat cada 2 segons per a la publicació del parell indicat a la pantalla principal (il·lustració 84). Així s'aconsegueix un control dinàmic d'aquest límit sense una sobre utilització dels recursos de comunicació de l'ordinador amb el robot.



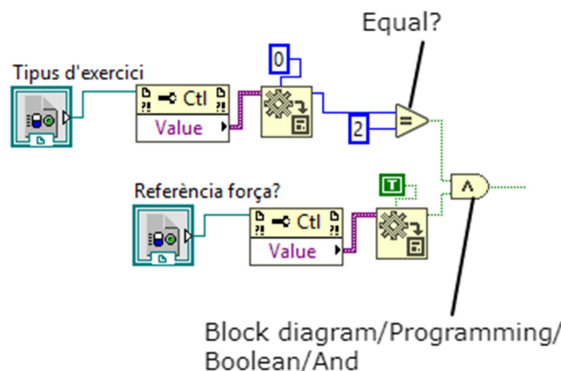
Il·lustració 84. Codi complet per publicar el límit

- Referència de força: genera la referència de força necessària per als exercicis resistius.



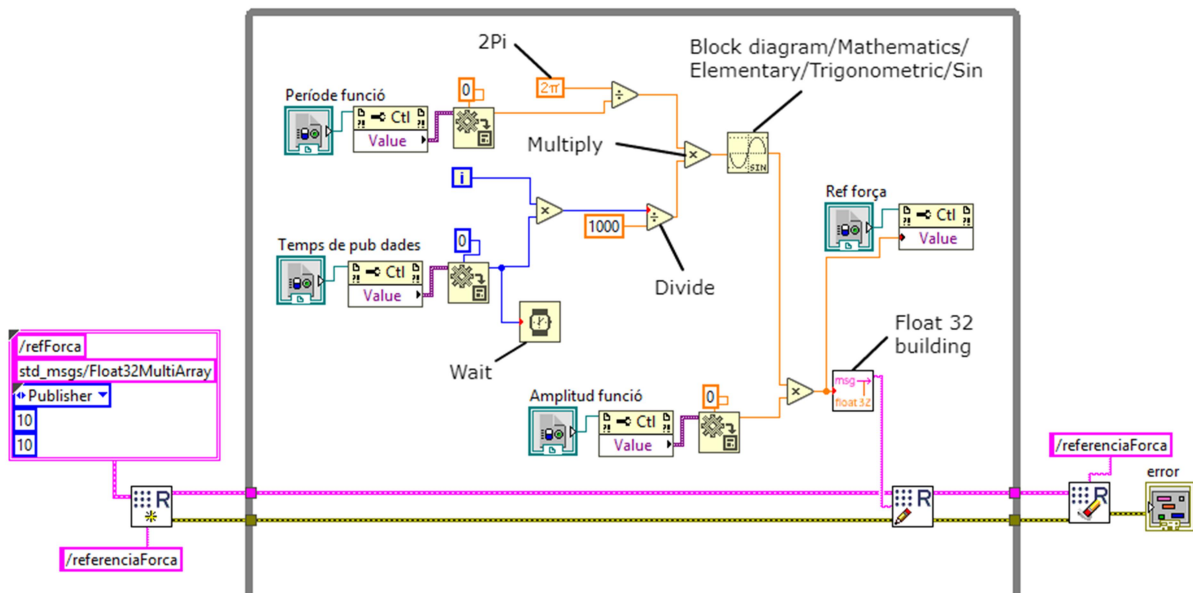
Il·lustració 85. VI referència de força

Consisteix d'un “timed loop” que es repeteix cada 500 ms i es tanca amb el “finalitzar VI paral·lels”. Dins d’aquest es disposa d’una estructura “case” amb el terminal de condicional controlant si és un exercici resistiu i si és el moment d’executar la referència (il·lustració 86).



Il·lustració 86. Codi per al condicional del “case structure”

Dins d'aquesta "case structure" al cas de "true" es troben uns càlculs per tal de generar una funció sinusoidal variant amb el temps i que vaja publicant el valor de la funció entre el màxim i el mínim cada moment marcat pel "temps de publicació de dades" (il·lustració 87). A més a més, aquest bucle "while" es tancarà quan deixi de ser un exercici resistiu o ja no estiga activa la situació de necessitar una referència. En el cas d'estar al "false" no es publicarà res, ja que aquesta referència és solament necessària als exercicis indicats anteriorment.



Il·lustració 87. Codi per calcular la referència i publicar-la

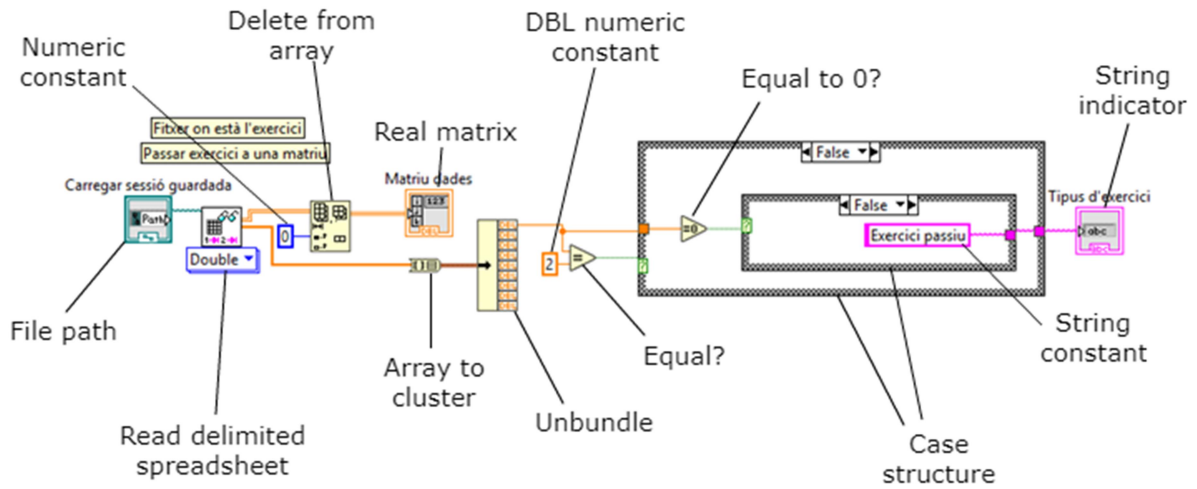
5.3.3. Recuperar sessions

Aquesta part de la programació està dividida en dues estructures principals: un "case structure" i un "event structure" (amb un "temps de timeout" de 100 ms). El primer amb l'objectiu d'atendre al botó de recuperar sessió i permetre carregar les dades i representar-les als gràfics XY, i el segon per a permetre la representació dels angles del peu, ja que depèn del controlador de l'instant a representar.

En el "case structure", que va connectat al botó de recuperar sessió, es disposa d'un fals que no conté ningun tipus de codi. En el vertader es disposa de tres parts clarament dividides.

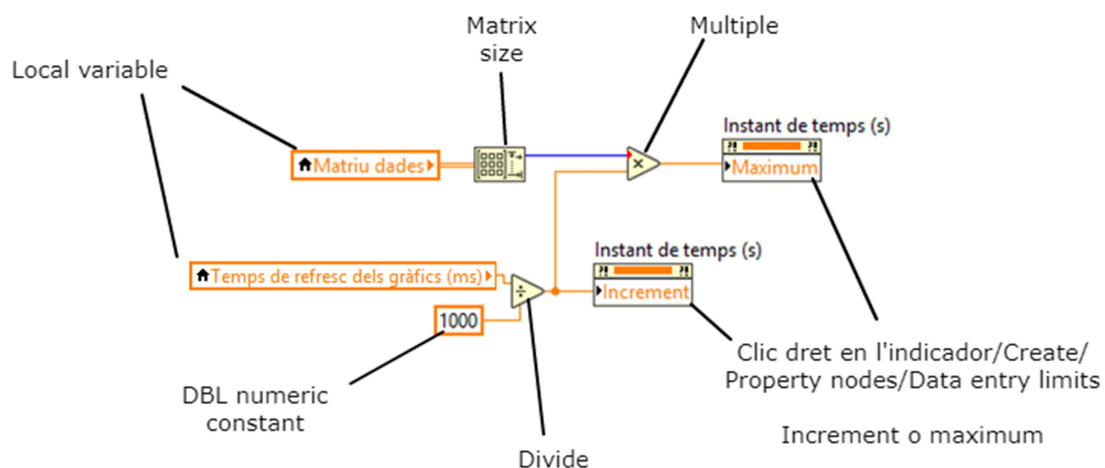
Primerament la part que serveix per tal d'obrir l'arxiu i indicar quin tipus d'exercici es tracta (il·lustració 88). Per obrir-lo es disposa d'un "read delimited spreadsheet" amb la seua corresponent ruta de l'arxiu. Una vegada obert s'ha decidit ficar-lo a una matriu per tal de poder fer ús d'aquesta dins de "l'event structure" posterior fàcilment. Per tal d'indicar el tipus es compara el valor de la posició (0,0) de l'arxiu de dades, i si és un 0 es tracta d'un exercici gravat, si és un 1 un exercici passiu i si és un 2 un resistiu. Açò també explica per què s'utilitza un "delete from array" per a la fila 0, ja que aquesta fila solament serveix per tal d'indicar el tipus, és a dir, no conté informació útil de la sessió a representar. La comparació per tal de

veure quin nombre és, es realitza amb els comparadors “equal?” i “equal to 0?”, i la decisió del text a mostrar amb dos “case structure” niats.



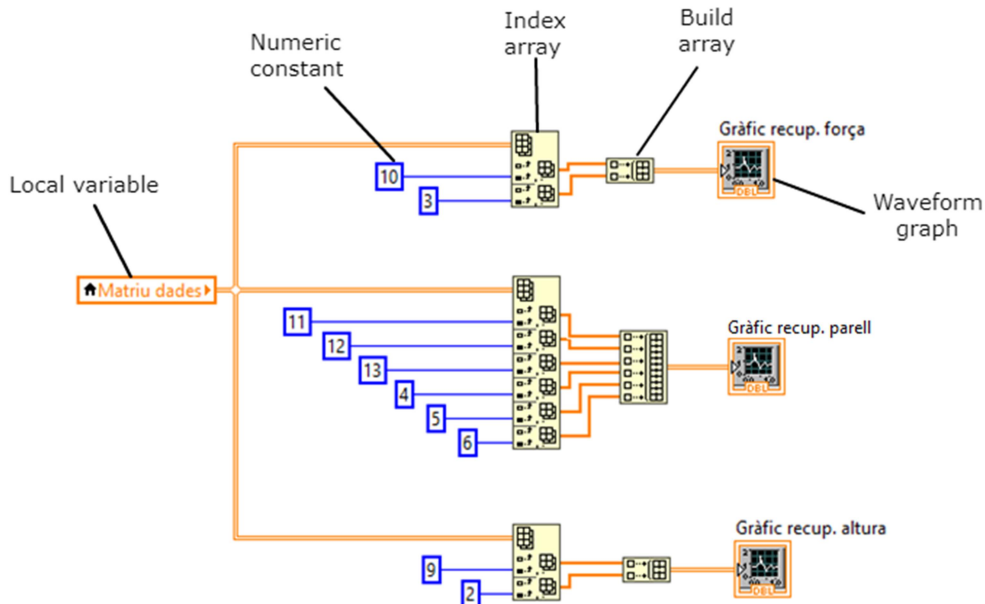
Il·lustració 88. Codi per obrir un arxiu i mostrar el tipus d'exercici

En segon lloc es troba la part per adequar el control de l' instant dels gràfics polars al temps d'execució de l'exercici i a l'interval en el què s'han agafat les dades (il·lustració 89). L'increment es calcula simplement llegint quin és el temps de refresc dels gràfics actual passant-lo a segons, ja què es considera què com solament pot variar-ho l'enginyer no deuria de canviar molt a sovint. I el màxim es calcula comprovant quantes files hi ha a l'arxiu i multiplicant-les per l'increment de temps que ha suposat cada filera.



Il·lustració 89. Codi per modificar l'indicador de l' instant de temps

I per últim mostrar els gràfics de força, parell i altura dels què es disposen, què no és més que elegir correctament les columnes on es guarden les dades corresponents, construir un "array" amb aquestes i passar les dades al gràfic corresponent (il·lustració 90).



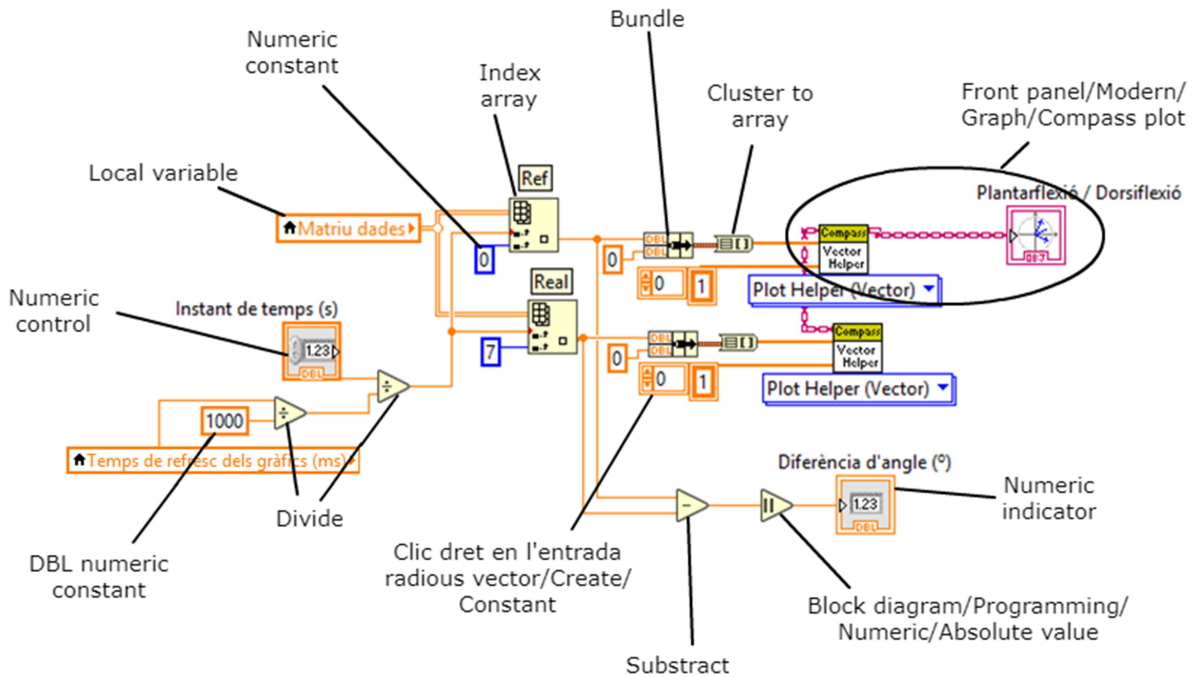
Il·lustració 90. Codi per mostrar els gràfics

Ja en "l'event structure" l'únic esdeveniment que s'ha de crear és el canvi de valor del control de l'instant dels gràfics polars, ja que cada vegada que varia s'han de mostrar els nous valors (il·lustració 91). Dins d'aquest s'ubica primerament l'elecció de la columna on es troben les dades reals i de referència de l'angle del peu (la qual cosa és fixa perquè sempre es guarda en el mateix ordre) i, com aquest tipus de gràfic solament representa un valor, s'elegeix la fila corresponent amb el controlador de l'instant abans citat (fent una reconversió per tal de traduir aquest instant en la fila corresponent).

Una vegada es té el valor desitjat, s'ajunta en un vector amb un 0, ja què els "compass plot" solament permeten la representació de vectors, ja siguin reals o imaginaris, però no permeten representar valors individuals. Posteriorment es transforma de "cluster a array", per tal de poder representar-lo, i s'introdueix en el "plot helper" paral·lelament a un "array" de constants que serveix per indicar la longitud de cadascun dels valors del vector.

Si es volen representar valors amb diferents colors, com és el cas, simplement s'uneixen els "plot helper" i es duen a un mateix gràfic polar.

Per acabar d'explicar aquesta part, els indicadors de la diferència d'angle solament agafen els valors actualment representats i obtenen la diferència entre ells, mostrant-ho per pantalla.



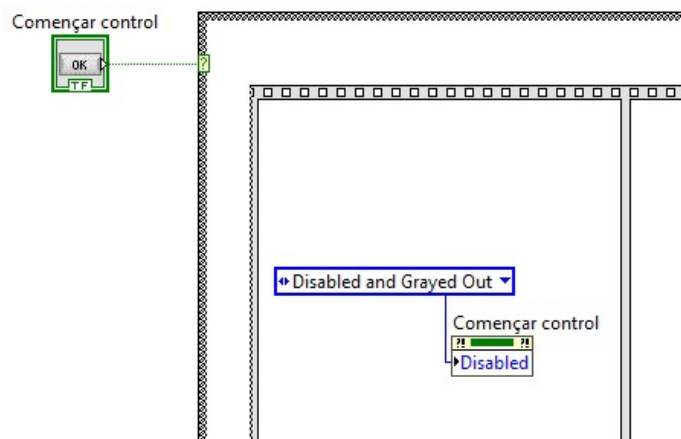
Il·lustració 91. Codi per mostrar els gràfics polars

Aquesta part és solament per a un dels dos gràfics, s'hauria de repetir el mateix canviant els "numeric constant" per les columnes on s'emmagatzemen les dades de l'altre angle.

5.3.4. Control de l'enginyer

Ací també hi ha dues estructures principals: un "case structure" i un "event structure".

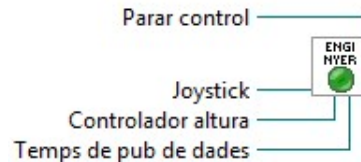
La primera va condicionada pel botó començar control, i ja que la part del "false" està buida, al "true" es troba un "flat sequence" amb tres requadres. El primer desactiva el botó de començar control i el fa translúcid (il·lustració 92).



Il·lustració 92. Codi del primer requadre del control

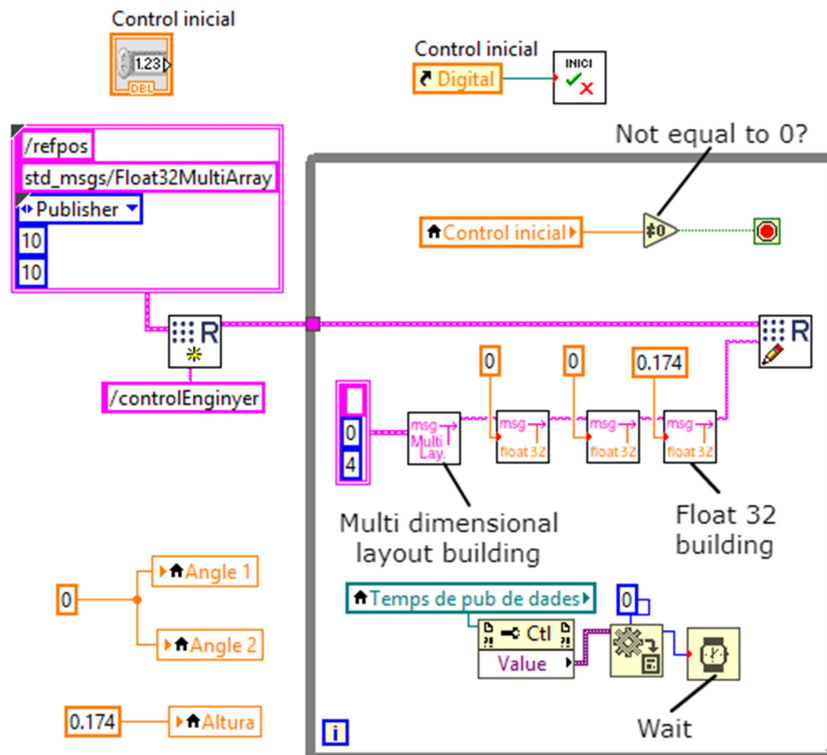
Al segon requadre trobem dos instruments paral·lels amb un bucle "while". El bucle conté el codi per a generar el "joystick", el qual es pot trobar a <https://forums.ni.com/t5/Example-Programs/Joystick-Emulator-Using-LabVIEW/ta-p/3496390> i es tanca amb el botó de parar control. Per altra banda, els instruments paral·lels són unes versions similars a uns altres instruments que ja s'han utilitzat, però així i tot es descriuran a continuació.

- Càlcul i publicació de posicions.



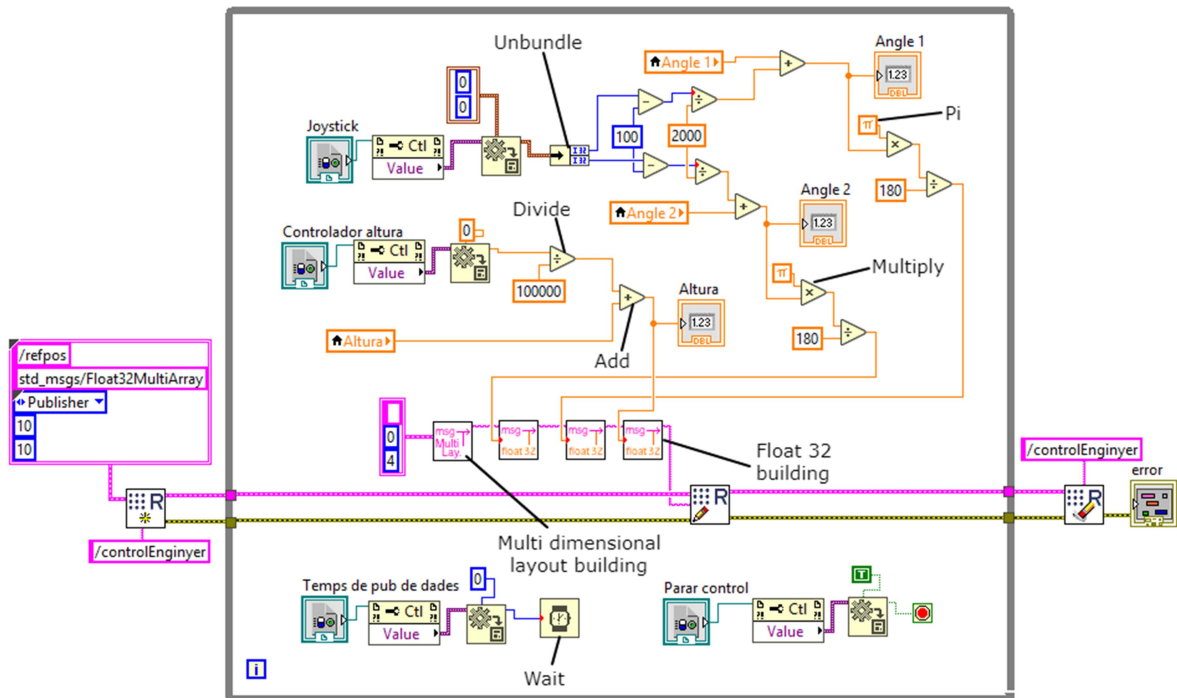
Il·lustració 93. VI per calcular i publicar les posicions

Consisteix d'un "flat sequence" amb dos requadres. El primer utilitza el VI de la il·lustració 70 per esperar-se a la primera execució. Mentre açò ocorre, es publica paral·lelament la posició final d'aquesta execució, per tal de mantindre el canal de comunicació obert (il·lustració 94).



Il·lustració 94. Codi per publicar la posició inicial i esperar a l'estabilitat del robot

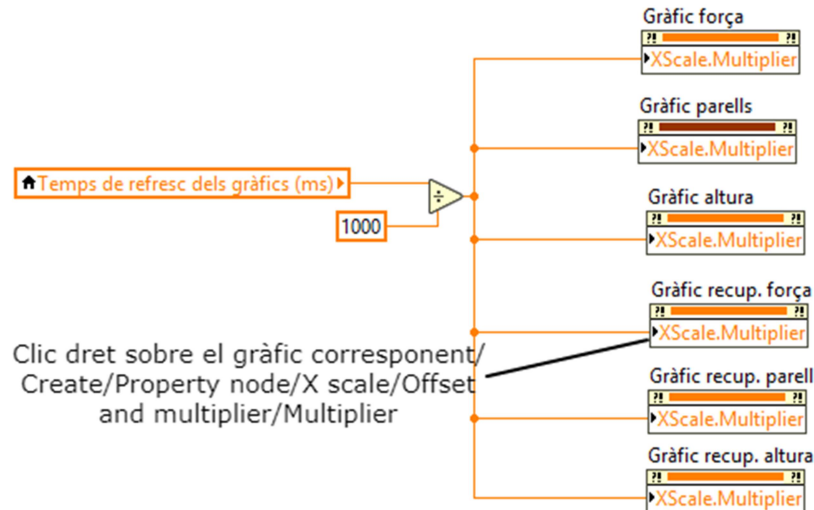
Una vegada es passa al segon requadre, allí es troba un bucle “while” que publica cada temps de publicació establert, i fins que es pare el control, la posició anterior més un increment de posició que depèn dels valors del “joystick” i del controlador d'altura. Per tal d'aconseguir açò s'han realitzat una sèrie de transformacions intermèdies sobre els valors que s'obtenen d'aquests controladors (il·lustració 95).



Il·lustració 95. Codi per aplicar increments i publicar la nova posició

- Indicador del final del control: es tracta de exactament la mateixa estructura mostrada a la il·lustració 60, amb l'única diferència que el condicional del final del bucle “while” està connectat al valor del botó “parar control”, per referència òbviament. Remarcar, per tant que aquesta estructura i l'anterior no contenen ningun “case structure” o “timed loop” a diferència del “Indicar el final d'un exercici de gravació o d'execució al robot”.

Passant a tractar la segona estructura, "l'event structure", té un "temps de timeout" de 100 ms i solament un esdeveniment, el canvi de valor del refresc dels valors dels gràfics. Dins d'aquest s'estableix l'escala de temps de tots els gràfics del programa (il·lustració 96).



Il·lustració 96. Codi per actualitzar l'escala de l'eix d'abscises dels gràfics

5.3.5. Retocs finals

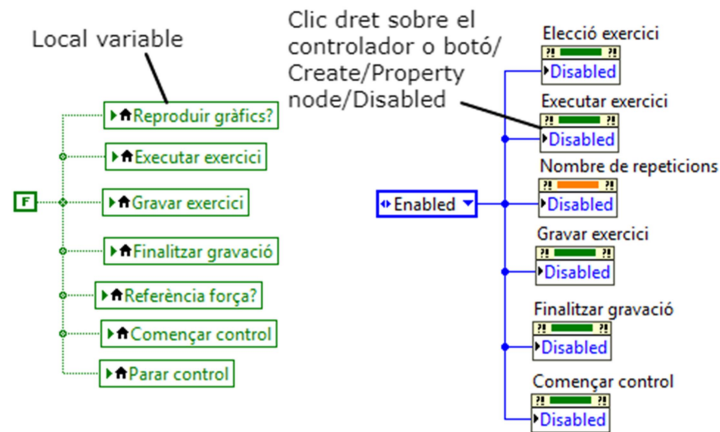
Ja havent realitzat tota aquesta feina, solament queden un parell de coses a establir per al correcte funcionament de l'aplicació. Primerament, a tots els casos del "case structure" de la il·lustració 43, excepte a la "pàgina principal de control", s'ha d'afegir un "local variable" del botó "finalitzar VI paral·lels" connectat a una "false constant".



Il·lustració 97. Codi per al botó "finalitzar VI paral·lels"

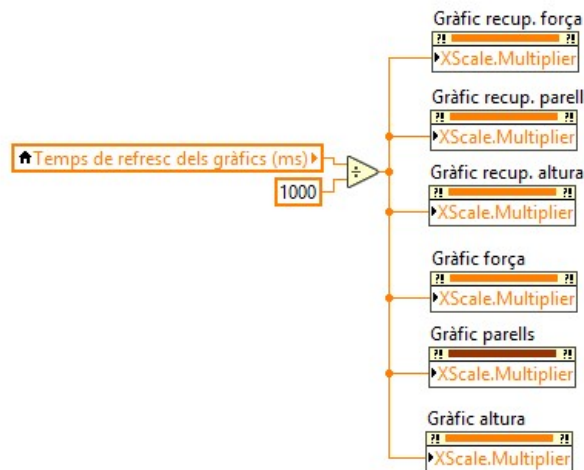
I a la part exterior del "timed loop" de la mateixa il·lustració, s'han d'establir uns valors inicials per tal de:

- Evitar errors del programa al tancament forçós d'aquest.



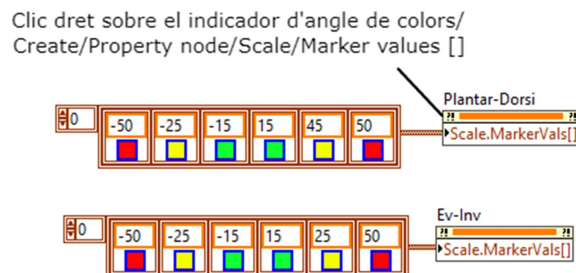
Il·lustració 98. Codi per inicialitzar valors

- Establir valors modificats sense estar en funcionament el programa.



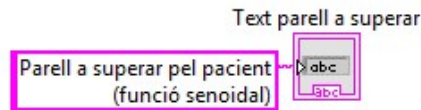
Il·lustració 99. Codi per inicialitzar l'escala d'abscises

- Modificar els color dels indicadors.



Il·lustració 100. Codi per establir els colors dels indicadors d'angle

- Indicar el text a mostrar.



Il·lustració 101. Codi per mostrar un text

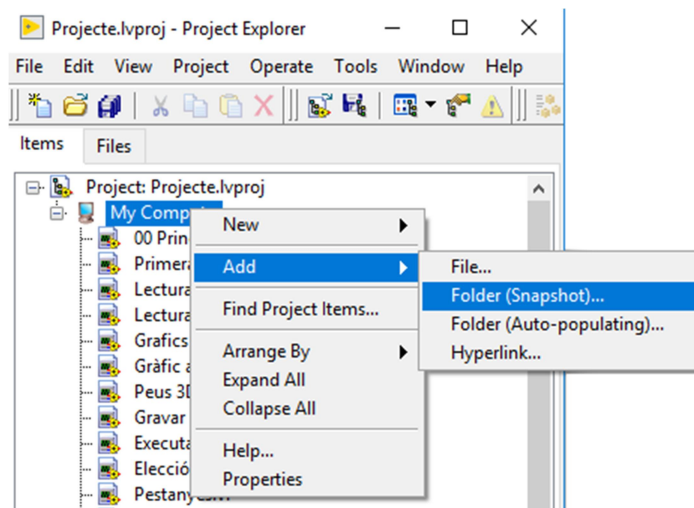
Amb açò finalitza tota la programació existent per al funcionament de la interfície proposada. A l'annex 2, com ja s'ha indicat abans, es troben els codis complets dels VI paral·lels de la "pàgina principal de control".

5.4. WEZARP

Com a complement a la interfície presentada, es proposa aquesta aplicació que permet la utilització d'interfícies desenvolupades a LabVIEW en dispositius mòbils. L'únic requisit és tindre l'ordinador i el mòbil connectats a la mateixa xarxa.

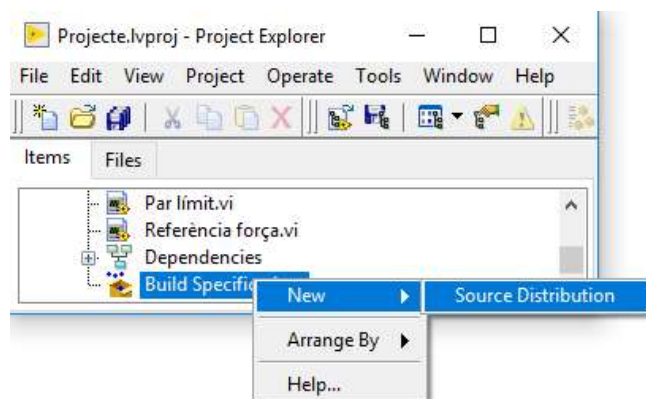
Per tal de fer funcionar aquest programa s'ha d'instal·lar, a l'ordinador, la llibreria de Wezarp per a LabVIEW present a la seua pàgina oficial <http://www.wezarp.com/> i, al mòbil o tableta, l'aplicació de Wezarp per a dispositius mòbils present a la "app store" o a "itunes".

Una vegada instal·lada a l'ordinador s'ha d'accedir al projecte on es troba l'instrument virtual en el què va a ser utilitzat. Es fa "clic dret sobre my computer/Add/Folder(Snapshot)..." (il·lustració 102) i es busca l'arxiu anomenat "WEZARP_SRV".



Il·lustració 102. Procés per carregar el WEZARP_SRV

Després es procedeix, sense eixir del projecte, amb “clic dret sobre build specifications/New/Source distribution” (il·lustració 103). Amb açò s’obrirà un diàleg nou on s’ha d’accedir a “Source files”, polsar el “WEZARP_SRV” abans afegit i passar-lo amb les fletxes a “always included”.



Il·lustració 103. Procés per afegir el WEZARP_SRV

Acabat aquest procés, s’accedeix a l’instrument virtual on es vol utilitzar la duplicació de pantalla i s’arrossega, al “block diagram”, l’instrument virtual de Wezarp, que usualment es troba a “block diagram/programming/Wezarp”. S’executa el programa de manera normal i per tal de configurar els clients, s’accedeix a la fletxa de Windows “mostrar icones ocultes” o “Windows systray” (il·lustració 104), es clica amb botó dret sobre la icona de Wezarp i s’accedeix a “clients/configuration”.



Il·lustració 104. Configurar clients

Amb el diàleg nou es clica el botó de baix a l’esquerra per afegir usuaris nous i es configura el tipus d’accés, s’indica la IP que fica a l’aplicació de mòbil com a “client IP” (il·lustració 105) i el port que es va a utilitzar, i s’accepta.



Il·lustració 105. Aplicació mòbil de Wezarp

Al mòbil s'ha de crear un nou servidor amb la fletxa blava de dalt a l'esquerra de la pantalla i apareixerà el requadre superior que es pot veure a la il·lustració anterior. Ací es pot, polsant el nombre que es vulga, modificar-lo per tal de ficar la IP de l'ordinador i el port que s'haja ficat al configurar el client. Una vegada configurat tot açò, el dispositiu duplica la pantalla de LabVIEW, i si està configurat per poder modificar també, permet la utilització de l'instrument virtual des del dispositiu mòbil.

6. CONCLUSIONS

Amb aquest projecte s'ha aconseguit l'objectiu final de desenvolupar una interfície funcional per al robot paral·lel de rehabilitació disponible. A més a més s'han obert algunes possibilitats per tal de poder millorar el funcionament entre el robot i la interfície.

També s'ha aconseguit un mètode de generació dels exercicis el més proper a la rehabilitació manual, ja que al haver de moure el robot manualment s'eviten possibles errors al generar els exercicis, com poden ser que el metge no entenga correctament els controls o que la inclinació a la que es va a sotmetre al peu siga excessiva.

Per altra banda, s'han pogut avaluar les prestacions de la comunicació via topic entre el robot i l'ordinador, observant la presència d'una limitació en la velocitat de publicació, la qual no permet enviar o rebre dades a velocitats majors de 50 ms. I s'ha pogut constatar també, a part d'un retard al principi de la lectura dels valors quan s'executa o grava un exercici, que si es demora molt la publicació a un topic posteriorment ja no es reben dades a través d'ell.

Un dels problemes més importants que s'afrontà a l'hora de fer funcionar el robot va ser el temps de transmissió de dades entre el robot i l'ordinador, ja que el primer calculava l'acció per als motors cada 10 ms, però la interfície no permetia enviar dades amb eixa freqüència. Per solucionar-ho, primerament es va decidir enviar i rebre les dades amb menor freqüència, el que repercutia en variacions de posició superiors. I en segon lloc, modificar els paràmetres del controlador dels motors, perquè la resposta a una referència nova fora més suau, evitant així moviments bruscos indesitjats.

Amb açò en ment i la interfície plantejada, s'obrin diverses possibilitats per poder continuar el projecte amb l'objectiu de fer-lo més intuïtiu i millorar l'experiència de la rehabilitació. Una d'aquestes possibilitats seria intentar incloure un "joystick" real per tal que el metge tinga dues possibilitats en la gravació dels exercicis, i no estiga limitat, si és el cas, per la seua condició física. I també s'obri la possibilitat de preparar una interfície per al pacient amb l'objectiu que aquest poguera veure, des del seu mòbil amb el Wezarp, el desenvolupament de la sessió, la seua progressió i que poguera deixar "feedback" de la mateixa per veure si s'estan produint millores.

BIBLIOGRAFIA

1. Vallés, M., Casalilla, J., Valera, Á, Mata, V., Page, Á, & Díaz-Rodríguez, M. (2017). "A 3-PRS parallel manipulator for ankle rehabilitation: Towards a low-cost robotic rehabilitation". *Robotica*, 35(10), 1939-1957. doi:10.1017/S0263574715000120.
2. José Ignacio Casalilla Morenas, "Implementación basada en el middleware orocos de controladores dinámicos para un robot paralelo" (Tesina final de màster).
3. "Wezarp for NI LabVIEW user guide".
http://www.download.wezarp.com/Docs/WEZARP_LABV_UserGuideEn.pdf

En quant a les il·lustracions utilitzades en aquest document són totes pròpies exceptuant (tots els hipervincles de la present memòria han sigut revisats a 16/5/2018):

Il·lustració 1. Unimate

https://twitter.com/ria_robotics/status/730801035720151040

Il·lustració 2. Shakey

<http://www.ai.sri.com/shakey/images.php>

Il·lustració 3. Stanford Arm

<https://ifr.org/robot-history>

Il·lustració 4. T3

<http://www.computerhistory.org/collections/catalog/102640478>

Il·lustració 5. Puma

<http://migibot.blogspot.com.es/p/blog-page.html>

Il·lustració 6. Segons la transmissió A)rodes, B)erugues, C)potes

https://www.researchgate.net/figure/Figura-54-Robots-moviles-con-diversos-sistemas-de-locomocion-a-ruedas-b-orugas-c_fig3_309398090

Il·lustració 7. Robot serial

<https://www.escueladigitalcad.com/proximos-cursos/robot-kuka/>

Il·lustració 8. Robot paral·lel

<http://www.directindustry.com/prod/mgs-machine-corporation/product-148389-1577056.html>

Il·lustració 9. Vista superior del robot paral·lel de pintura de Pollard

<http://shvachko.net/?p=1523&lang=en>

Il·lustració 10. Vista lateral del robot paral·lel de pintura de Pollard

<http://shvachko.net/?p=1523&lang=en>

Il·lustració 11. Baxter

<https://phys.org/news/2012-09-lines-baxter-pick-up-video.html>

Il·lustració 12. Husky A200

<https://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>

Il·lustració 13. Nao

<https://www.robot-rentals.com/robots-for-rent/humanoid-robot/>

Il·lustració 14. Esquema del funcionament de ROS

<http://robohub.org/ros-101-intro-to-the-robot-operating-system/>

Il·lustració 21. Bota per al peu i sensor de força del robot objecte d'estudi

Present al document bibliogràfic [1]

Il·lustració 22. El robot objecte d'estudi

Present al document bibliogràfic [1]

Il·lustració 23. Possibles moviments del peu

https://www.researchgate.net/figure/Figura-1-Movimientos-del-tobillo-con-respecto-a-los-tres-ejes_fig1_315777987

I en relació a les taules també són pròpies exceptuant:

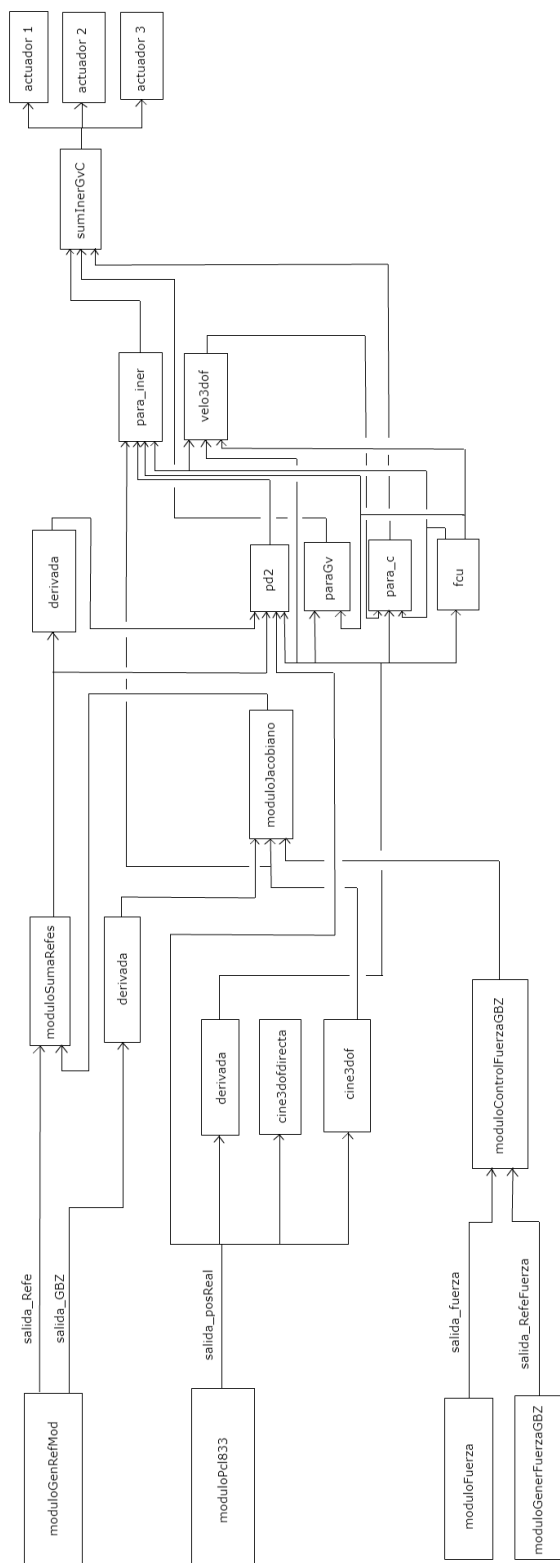
Taula 2. Moviments del turmell

Present al document bibliogràfic [1]

Taula 3. Forces màximes aplicables al turmell

Present al document bibliogràfic [1]

ANNEX 1

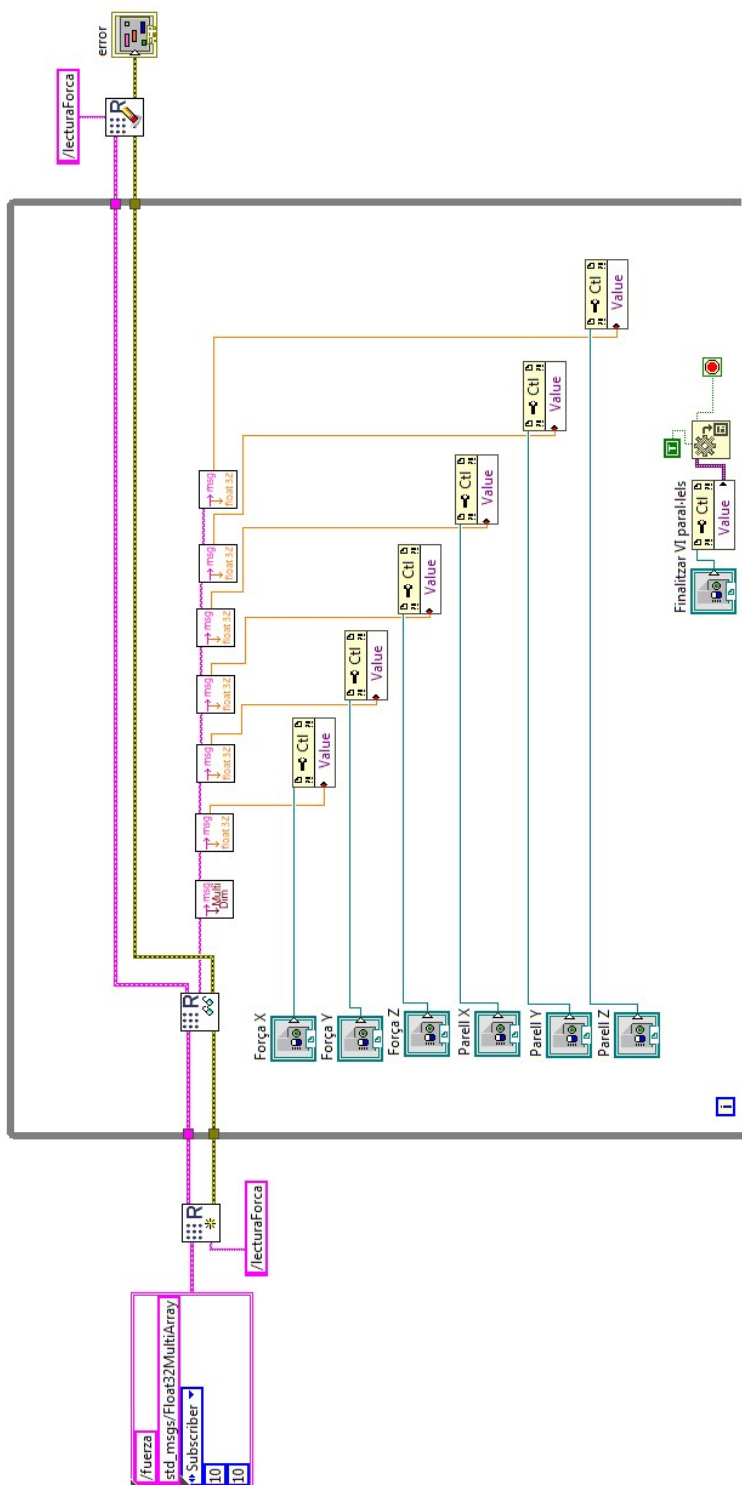


Il·lustració 106. Esquema dels mòduls del controlador

ANNEX 2

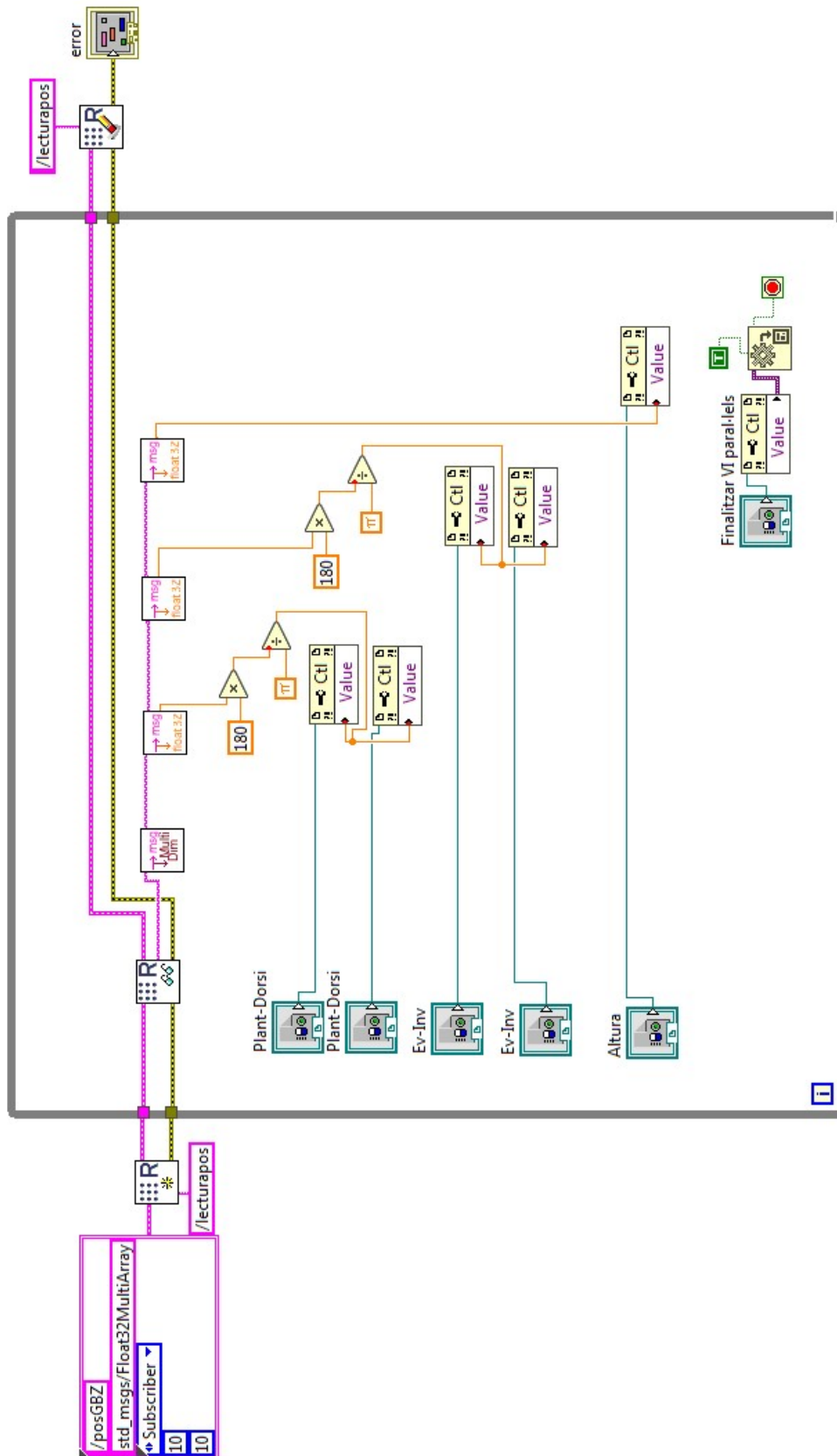
Codis complets dels instruments virtuals paral·lels emprats a la pestanya “pàgina principal de control” no mostrats a l'apartat corresponent.

Lectura de força



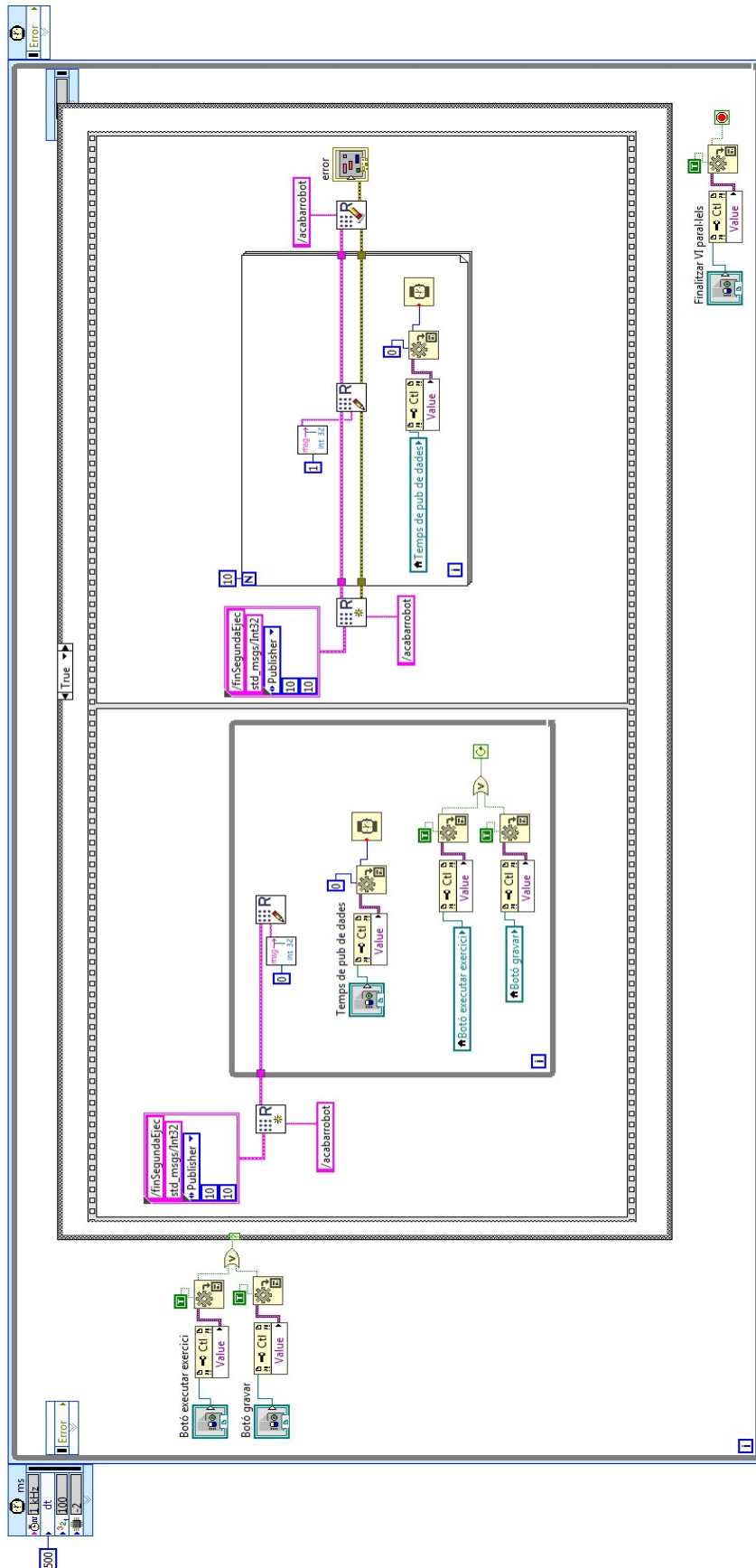
II-il·lustració 107. Codi complet lectura de força

Lectura de posició



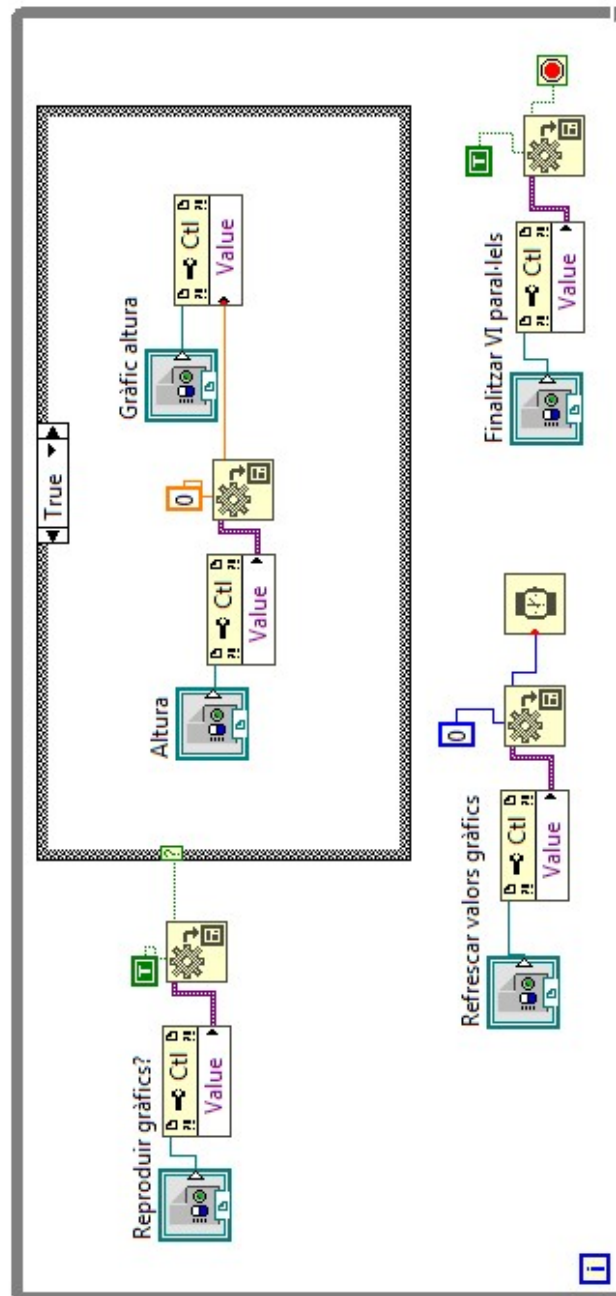
Il·lustració 108. Codi complet lectura de posició

Indicador del final d'un exercici de gravació o d'execució al robot



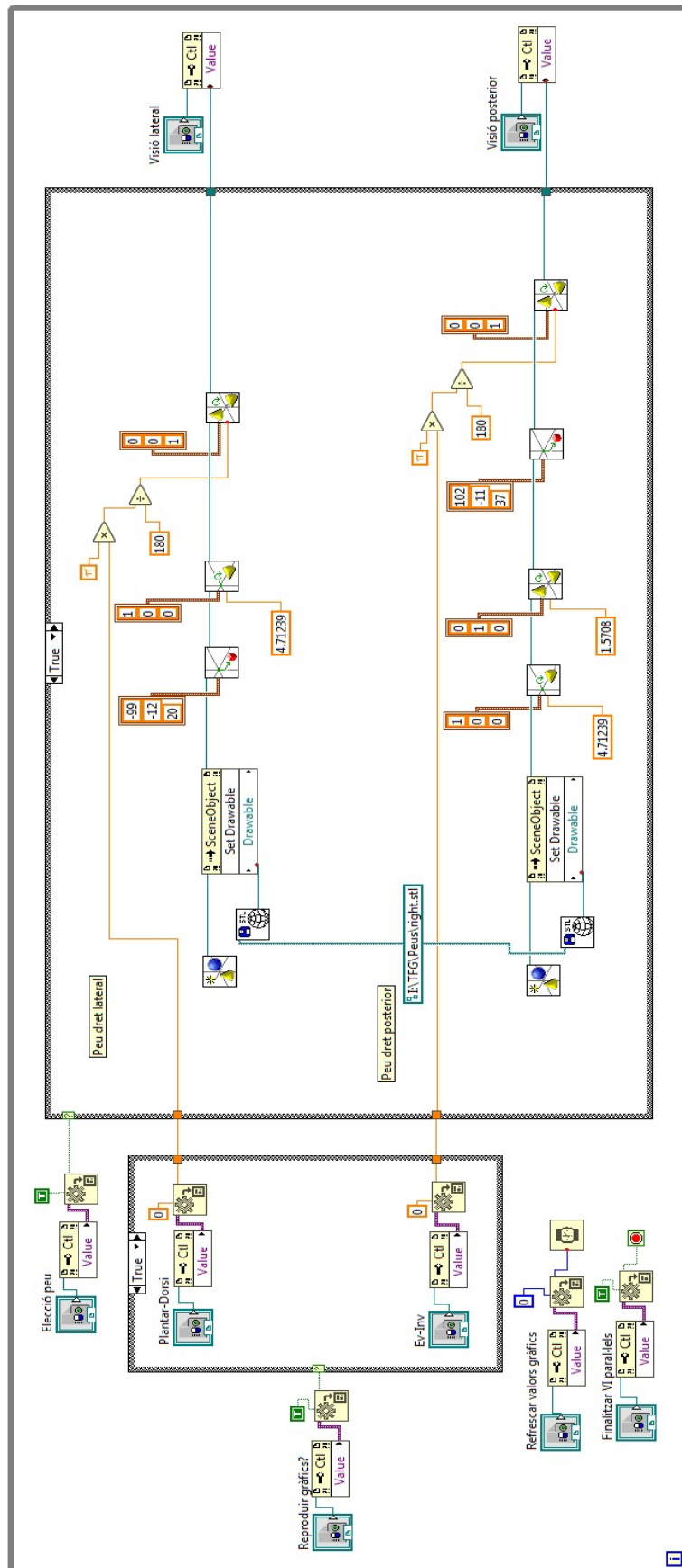
Il·lustració 109. Codi complet per indicar el final d'un procés

Gràfic altura

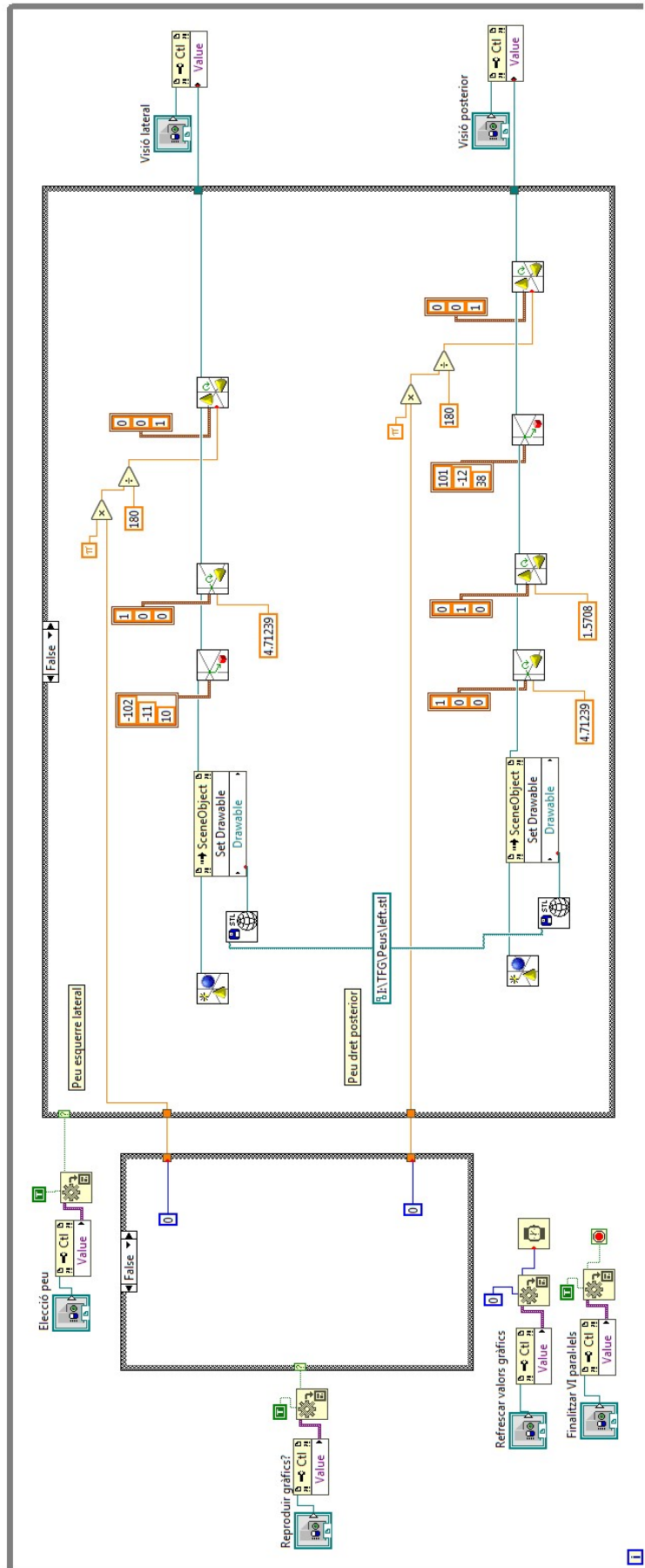


Il·lustració 110. Codi complet per refrescar el gràfic d'altura

Models 3D dels peus

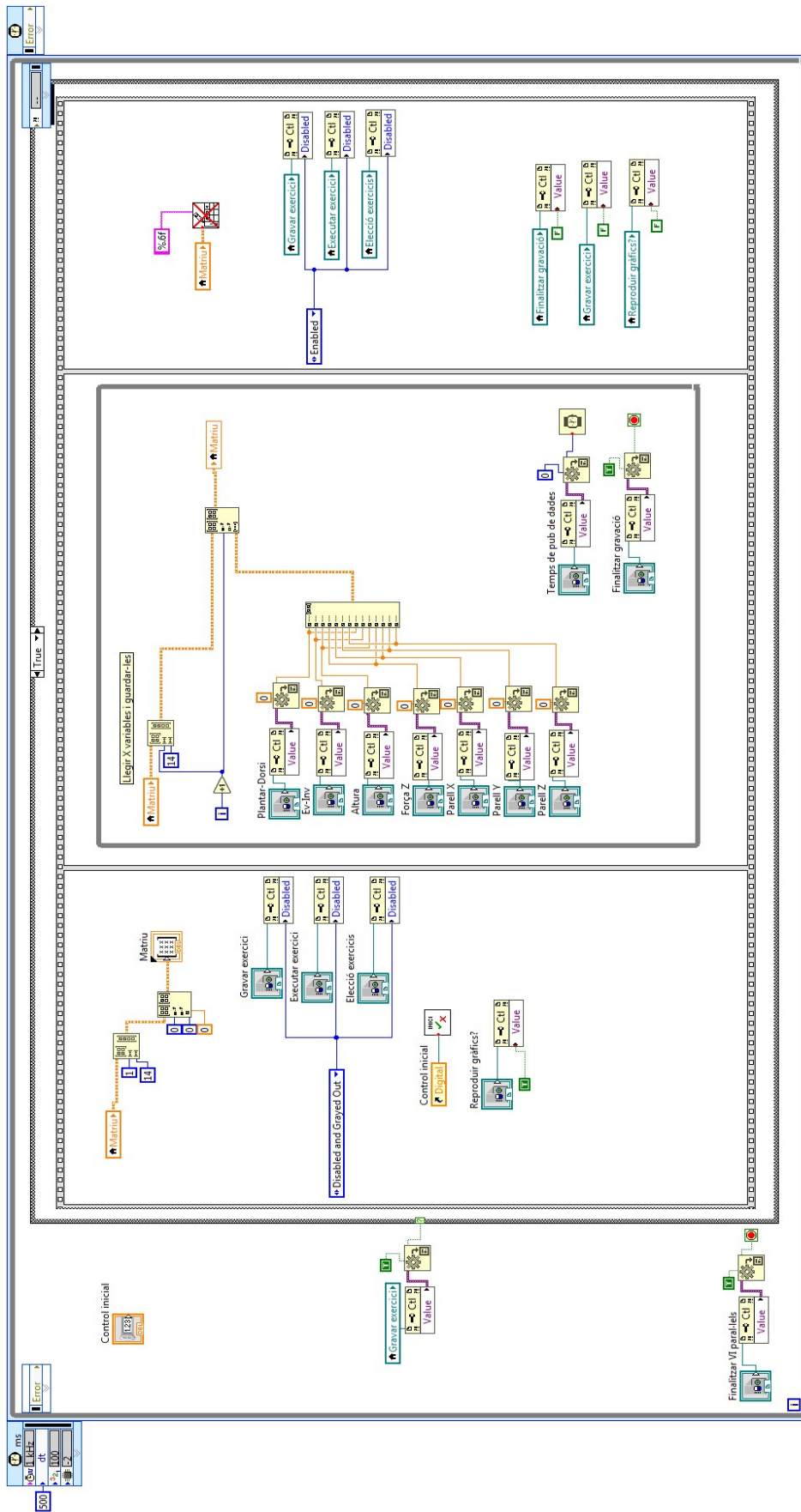


Il·lustració 111. Codi complet models 3D (1)



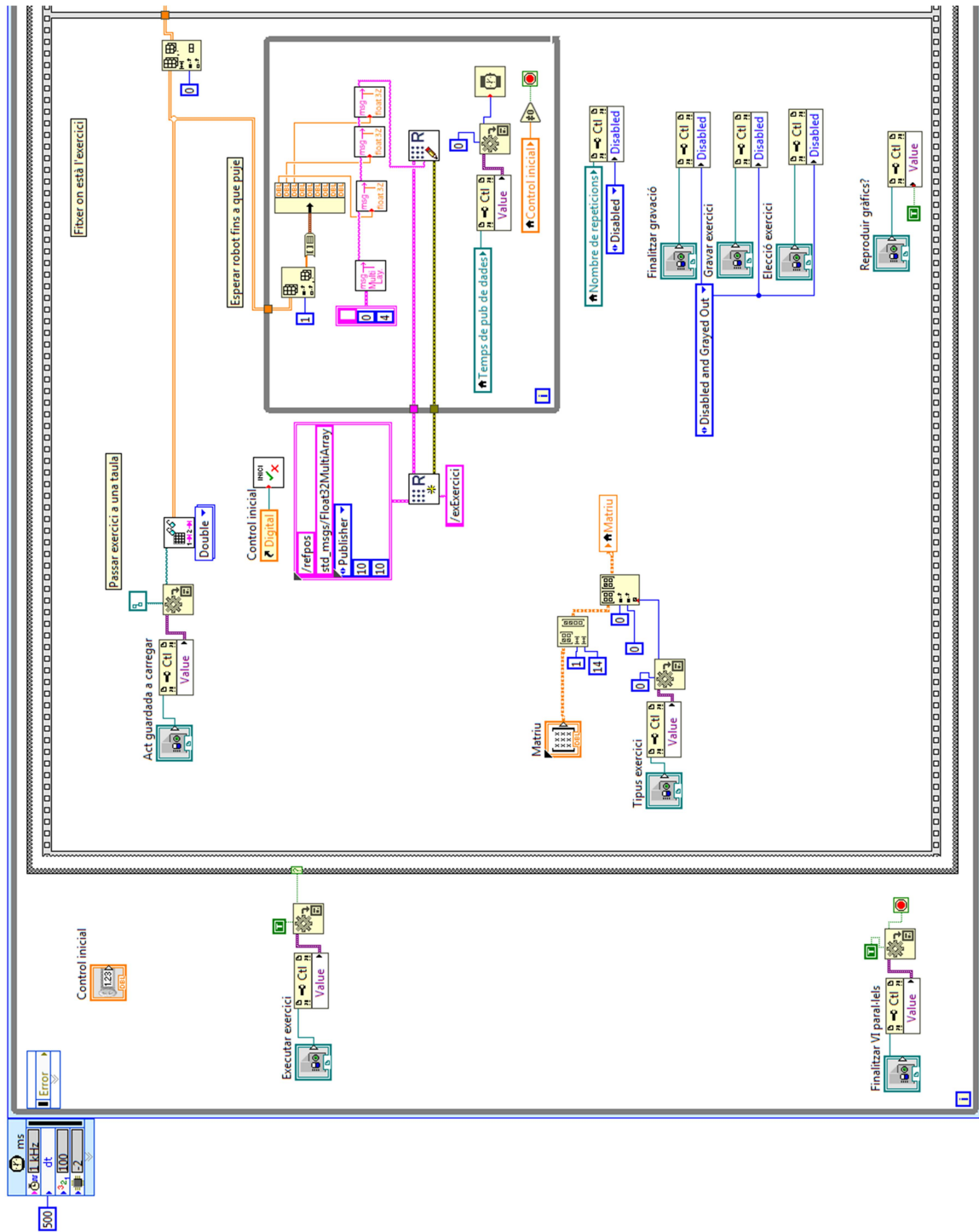
II·lustració 112. Codi complet models 3D (2)

Gravar exercici

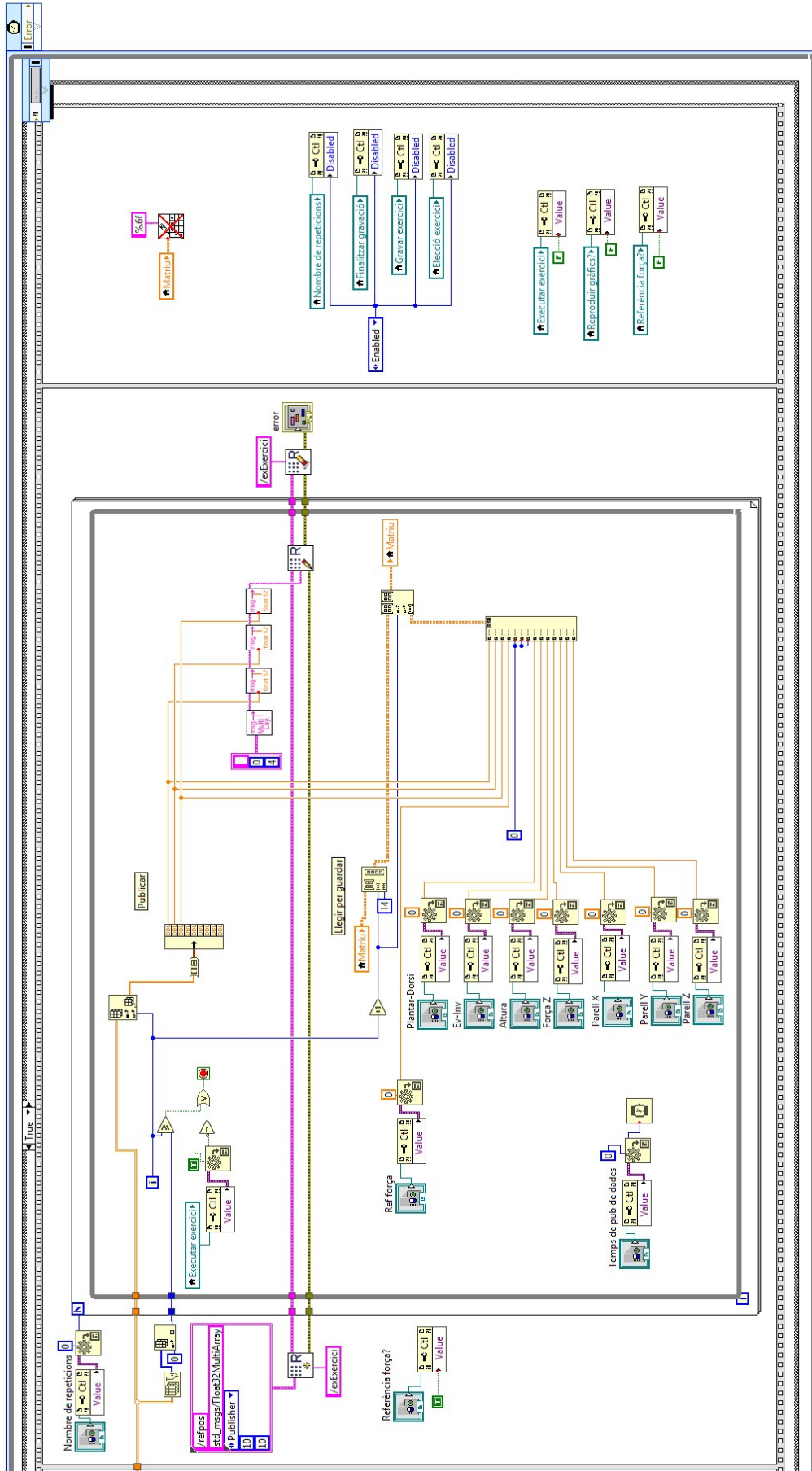


Il·lustració 113. Codi complet per gravar exercicis

Executar exercici

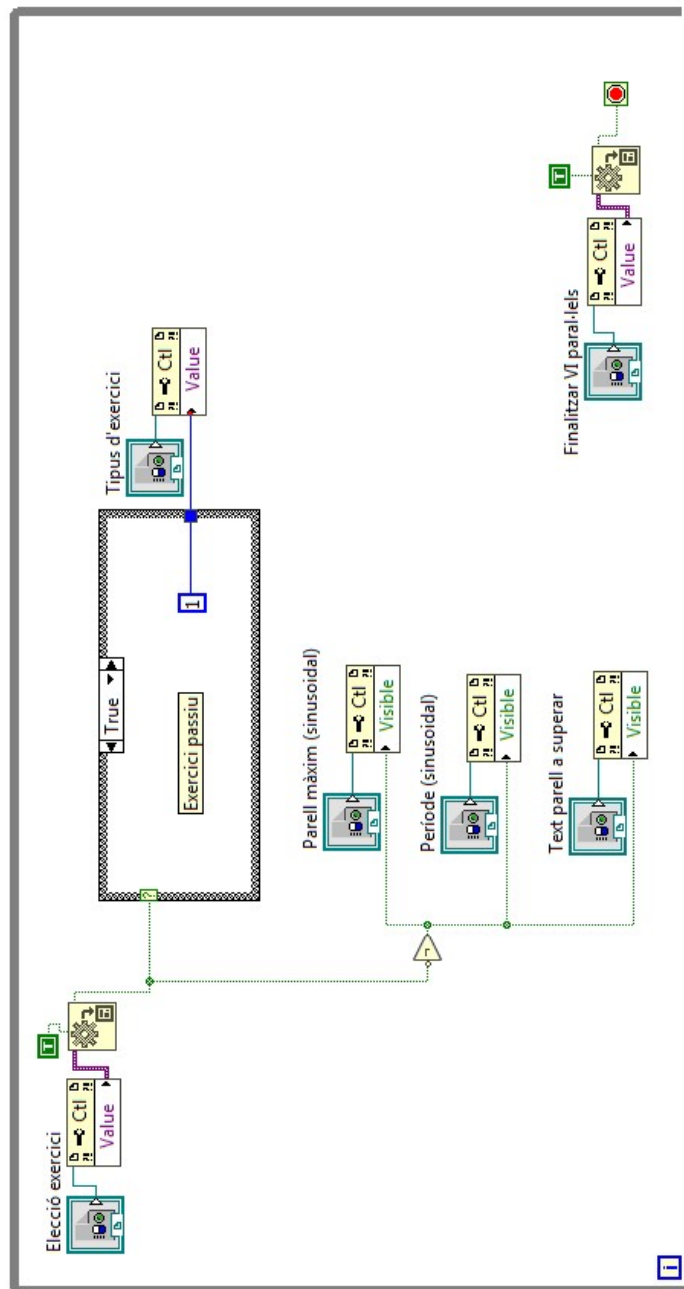


Il·lustració 114. Codi complet per executar exercicis (1)



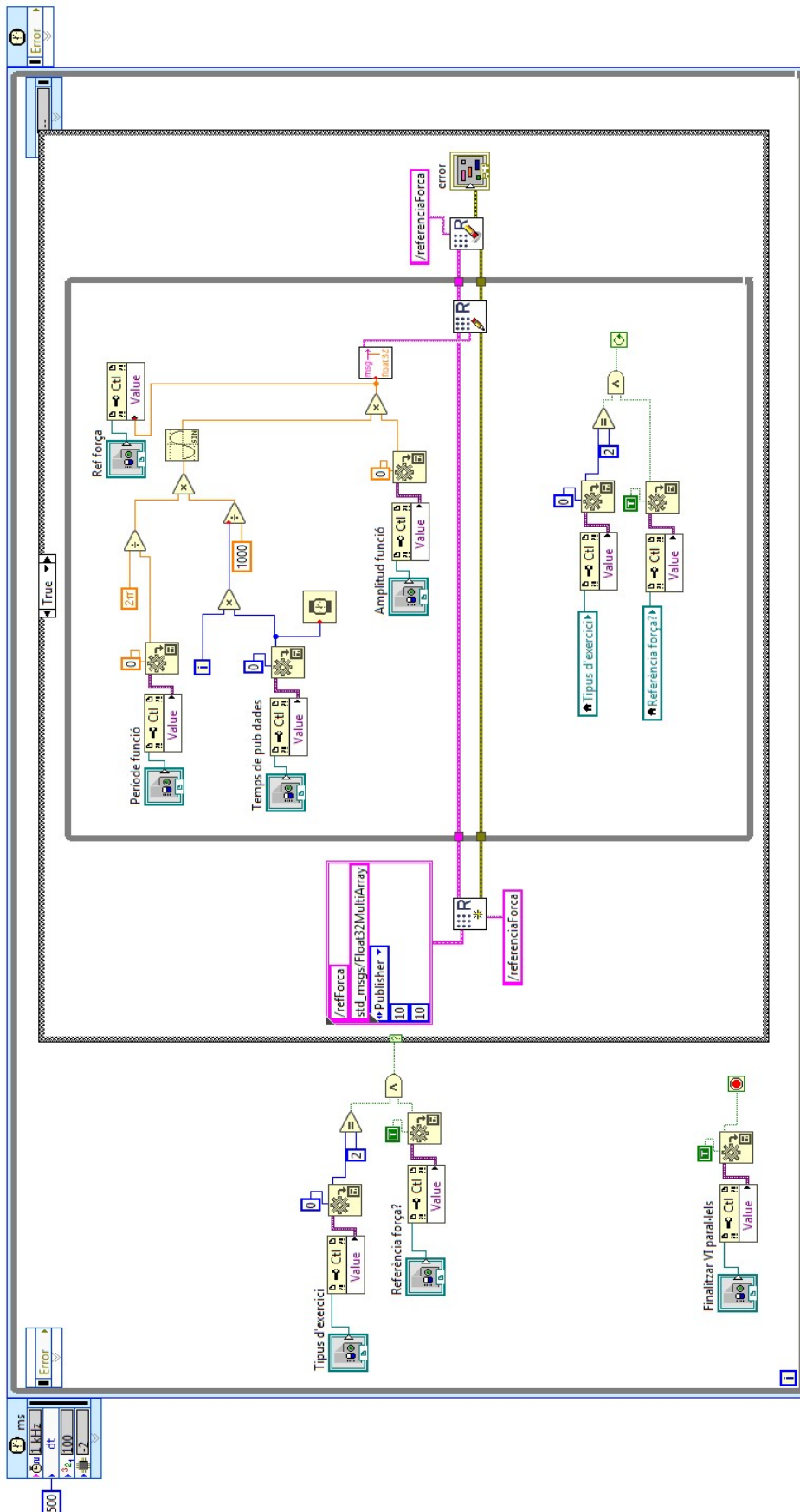
II-l·lustració 115. Codi complet per executar exercicis (2)

Canvis per l'elecció del tipus d'exercici



Il·lustració 116. Codi complet per executar els canvis pel tipus d'exercici

Referència de força



II-lustració 117. Codi complet per generar la referència de força

PRESSUPOST

En aquest projecte s'ha considerat que el robot ja es troba en mans del client interessat, per tant el preu d'aquest i de la seua programació no entren dins del pressupost.

1. DEFINICIÓ DE LES UNITATS QUE INTERVENEN

1.1. Mà d'obra

- Enginyer industrial: suposa el treballador principal i responsable del projecte. S'encarrega de la programació de la pròpia interfície, de la comprovació que tot funciona correctament i de la redacció dels documents posteriors.
- Enginyer tècnic: col·labora en la feina de comprovació del correcte funcionament de la interfície, per tant deu tindre coneixements de programació i robòtica.

1.2. Material

Taula 1. Quadre descriptiu del material

Nom	Definició
Hardware	
Torre d'ordinador	Asus M32CD-SP022T. Consta de Intel core i5 de 4 nuclis amb memòria RAM de 4 GB, disc dur intern de 1 TB i targeta gràfica integrada. Es necessita un ordinador per generar la xarxa de ROS (sistema operatiu Linux) i un altre per portar la interfície gràfica (sistema operatiu Windows).
Pantalla d'ordinador	HP V197 18.5. Pantalla de 18.5 polsades i resolució de 1366x768.
Software	
LabVIEW	El programa principal per al desenvolupament d'aquest projecte.
DSC module	Llibreria que permet l'accés d'usuaris a LabVIEW.
Wezarp	Permet duplicar la pantalla de LabVIEW a dispositius mòbils.

2. QUADRES DE PREUS UNITARIS

2.1. Mà d'obra

Taula 2. Quadre de preus unitaris de la mà d'obra

Categoria	Cost unitari (€/h)
Enginyer industrial	25,00
Enginyer tècnic	19,00

2.2. Material

En tot el pressupost s'ha considerat que el treball es realitza en el lloc on es va a establir el robot amb el material disponible, és a dir, amb el software i el hardware pressupostat.

Taula 3. Quadre de preus unitaris del material

Categoria	Unitat	Cost unitari (€/unitat)
Hardware		
Torre ordinador	Unitats	449,00
Pantalla ordinador	Unitats	78,65
Software		
LabVIEW	Llicències (per any)	399,00
DSC module	Llicències (per any)	3.214,00
Wezarp	Llicències (per any)	99,00

3. PRESSUPOSTS PARCIAIS

Es van a dividir en: l'apartat de desenvolupament del projecte, les proves per confirmar el correcte funcionament i la redacció posterior.

3.1. Material que s'empra en tot el projecte

Taula 4. Quadre de costos del material

Material					
Tipus	Descripció	Cost (€/unitat)	Quantitat	Unitat	Total (€)
Torre ordinador	Asus M32CD-SP022T	449,00	2	unitats	898,00
Pantalla ordinador	HP V197 18.5	78,65	2	unitats	157,30
LabVIEW	Programa per crear la interfície	399,00	1	llicències (per any)	399,00
DSC module	Accés d'usuaris	3.214,00	1	llicències (per any)	3.214,00
Wezarp	Duplicació de pantalla	99,00	1	llicències (per any)	99,00
				Subtotal	4.767,30

3.2. Desenvolupament del projecte

Taula 5. Quadre de mà d'obra del desenvolupament

Mà d'obra						
Acció	Descripció	Participen	Cost (€/unitat)	Quantitat	Unitat	Total (€)
Reunió tècnica	Establiment del procediment, plans, calendari...	Enginyer industrial	25,00	3	hores	75,00
		Enginyer tècnic	19,00	3		57,00
Programació del codi	Programació principal de la interfície d'usuari	Enginyer industrial	25,00	130	hores	3.250,00
					Subtotal	3.382,00

3.3. Proves

Taula 6. Quadre de mà d'obra de les proves

Acció	Descripció	Participen	Cost unitari (€/unitat)	Quantitat	Unitat	Total (€)
Reunió tècnica	Planificació de les proves a realitzar	Enginyer industrial	25,00	2	hores	50,00
		Enginyer tècnic	19,00	2		38,00
Test	Comprovar el correcte funcionament de la interfície juntament amb el robot	Enginyer industrial	25,00	20	hores	500,00
		Enginyer tècnic	19,00	20		380,00
					Subtotal	968,00

3.4. Redacció

Taula 7. Quadre de mà d'obra de la redacció

Acció	Descripció	Participen	Cost unitari (€/unitat)	Quantitat	Unitat	Total (€)
Redactar	Redacció del manual d'usuari i de les possibilitats de la interfície	Enginyer industrial	25,00	15	hores	375,00
					Subtotal	375,00

4. PRESSUPOST TOTAL D'EXECUCIÓ

Taula 8. Quadre del pressupost total d'execució

Pressupost parcial	Cost (€)
Material que s'empra en tot el projecte	4.767,30
Desenvolupament del projecte	3.382,00
Proves	968,00
Redacció	375,00
Pressupost total d'execució	9.492,30

5. PRESSUPOST TOTAL D'EXECUCIÓ PER CONTRACTA

Taula 9. Quadre del pressupost total d'execució per contracta

	Cost (€)
Pressupost total d'execució	9.492,30
Despeses generals (13%)	1.234,00
Beneficis industrials (6%)	569,54
Pressupost total d'execució per contracta	11.295,84

6. PRESSUPOST BASE DE LICITACIÓ

Taula 10. Quadre del pressupost base de licitació

	Cost (€)
Pressupost total d'execució per contracta	11.295,84
IVA (21%)	2.372,13
Pressupost base de licitació	13.667,97

En conclusió, el pressupost base de licitació ascendeix a **tretze mil sis-cents seixanta-set euros amb noranta-set cèntims**.